Detecting Social Spam Campaigns on Twitter

Zi Chu[1], Indra Widjaja[2] and Haining Wang[1]

[1] Department of Computer Science, The College of William and Mary, Williamsburg, VA 23187, USA
{zichu, hnw}@cs.wm.edu
[2] Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ 07974, USA
iwidjaja@research.bell-labs.com

Abstract. The popularity of Twitter greatly depends on the quality and integrity of contents contributed by users. Unfortunately, Twitter has attracted spammers to post spam content which pollutes the community. Social spamming is more successful than traditional methods such as email spamming by using social relationship between users. Detecting spam is the first and very critical step in the battle of fighting spam. Conventional detection methods check individual messages or accounts for the existence of spam. Our work takes the collective perspective, and focuses on detecting spam campaigns that manipulate multiple accounts to spread spam on Twitter. Complementary to conventional detection methods, our work brings efficiency and robustness. More specifically, we design an automatic classification system based on machine learning, and apply multiple features for classifying spam campaigns. Our experimental evaluation demonstrates the efficacy of the proposed classification system.

Keywords: Spam Detection, Anomaly Detection, Machine Learning, Twitter

1 Introduction

With the tremendous popularity of online social networks (OSNs), spammers have exploited them for spreading spam messages. Social spamming is more successful than traditional methods such as email spamming by taking advantage of social relationship between users. One important reason is that OSNs help build intrinsic trust relationship between cyber friends even though they may not know each other in reality. This leads to users to feel more confident to read messages or even click links from their cyber friends. Facilitated by this fact, spammers have greatly abused OSNs and posted malicious or spam content, trying to reach more victims.

Detecting spam is the first and very critical step in the battle of fighting spam. Our work chooses Twitter as the battlefield. Currently, Twitter is the most popular micro-blogging site with 200 million users. Twitter has witnessed a variety of spam attacks. Conventional spam detection methods on Twitter

mainly check individual tweets or accounts for the existence of spam [30, 16]. The tweet-level detection screens individual tweets to check whether they contain spam text content or URLs. As of August 2011, around 8.3 million tweets are generated per hour [9], and they demand near real-time delivery. Thus, the tweet-level detection would consume too much computing resources and can hardly meet time-stringent requirements. The account-level detection checks individual accounts for the evidence of posting spam tweets or aggressive automation behavior. Accounts violating the Twitter rules of spam and abuse [11] will get suspended. Suspending spam accounts is an endless cat and mouse game as it is easy for spammers to create new accounts to replace suspended ones.

Our work shifts the perspective from individual detection to collective detection and focuses on detecting spam campaigns. A spam campaign is defined as a collection of multiple accounts controlled and manipulated by a spammer to spread spam on Twitter for a specific purpose (e.g., advertising a spam site or selling counterfeit goods). Detecting spam campaigns is an important complement to conventional spam detection methods. Moreover, our work brings two additional benefits. (1) Efficiency. Our approach clusters related spam accounts into a campaign and generates a signature for the spammer behind the campaign. Thus, not only our work can detect multiple existing spam accounts at a given time, it can also capture future ones if the spammer maintains the same spamming strategies. (2) Robustness. There are some spamming methods which cannot be detected at individual level. For example, Twitter defines the behavior of "posting duplicate content over multiple accounts" as spamming. By grouping related accounts, our work can detect such a collective spamming behavior.

We have performed data collection for three months in 2011, and obtained a dataset with 50 million tweets posted by 22 million users. Using the dataset, we cluster tweets with the same final URL into a campaign, partitioning the dataset into numerous campaigns based on URLs. We perform a detailed analysis over the campaign data and generate a set of useful features to classify a campaign into two classes: spam or legitimate. Based on the measurement results, we present an automatic classification system using machine learning. We validate the efficacy of the classification system. The experimental results show high accuracy with low false positive rate.

The remainder of the paper is organized as follows. Section 2 presents a brief background of Twitter and covers related work of social spam detection. Section 3 details the data collection and measurements on Twitter. Section 4 describes our automatic classification system. Section 5 evaluates the system efficacy for detecting spam campaigns. Finally, Section 6 concludes the paper.

2 Related Work

As spammers often use Twitter-specific features to allure victims, we first briefly describe the background of Twitter and its working mechanism. Then, we survey related work in social spam detection and discuss the scope of our work.

2.1 Twitter and Related Social Spam Detection

Users post textual messages on Twitter, known as tweets. The tweet length is up to 140 characters, which limits the spam content the spammer can include in a tweet. Thus, embedding an external URL in a tweet becomes a routine for spammers to allure users to spam websites. A tweet may contain some textual features for better user interaction experience, which are also abused by spammers. A *hashtag*, namely a word or a phrase prefixed with the # symbol, is used to group tweets by their topic. For example, #Japan_Tsunami and #Egyptian_Revolution are two of the worldwide trending hashtags on Twitter in March 2011. Spammers may attach popular hashtags to unrelated spam tweets to increase the chance of being searched. This spamming trick is called hashtag hijacking. The *mention* feature, namely the @ symbol followed by a username in a tweet, enables the direct delivery of the tweet to the user. This feature facilitates spammers to directly send spam to targeted users.

Traditional spam methods include sending spam emails [31] and creating spam web content [27]. The past few years have witnessed the rapid rise of online social networks. One key feature of such systems is the reliance on content contributed by users. Unfortunately, the system openness coupled with the large user population has made OSNs an ideal target of social spammers. By exploiting the social trust among users, social spam may achieve a much higher success rate than traditional spam methods. For example, Grier *et al.* analyzed the click-through rate of spam on Twitter [21], and found out that around 0.13% of spam tweets generate a visit, orders of magnitude higher than click-through rate of 0.003% - 0.006% reported for spam email [24].

As a countermeasure, Twitter has released its rules against spam and abuse [11]. Accounts violating the rules will result in permanent suspension. The set of rules mainly define spam on Twitter in the following categories of content, behavior and social relationship. In the content category, it is forbidden to post content or URLs of any kinds of spam. Large numbers of unrelated @replies, mentions and #hashtags, or duplicate content are also disallowed. The behavior category covers both individual and collective behavioral codes. At the individual level, aggressive automation such as constantly running programs to post tweets without human participation is prohibited. At the collective level, using multiple accounts to post duplicate content is also considered as spamming. In terms of social relationship, one cannot follow a large number of users in a short amount of time, or have a small number of followers compared to the number of friends it is following, or create or purchase accounts in order to gain followers.

To avoid being detected by Twitter rules, social spammers have adopted a similar idea of email spam campaigns by coordinating multiple accounts to achieve a specific purpose. The spammer distributes the workload among spam accounts, thus individual accounts now may exhibit stealthy spam behavior and fly under the radar. Besides, multiple accounts also can spread spam to a wider audience. Some related studies have demonstrated the wide existence of spam campaigns on OSNs, such as Twitter and Facebook, respectively [21, 20]. The existing work mainly relies on the URL feature. More specifically, related mes-

sages with the shared final landing URL are clustered into a campaign. Then, the URL is looked up in URL blacklists. If the URL is blacklisted, the campaign is classified as a spam campaign; otherwise it is legitimate. Currently, the existing detection methods have some disadvantages listed as follows. First, URL blacklists have the lag effect, allowing more than 90% of visitors to click on a spam URL before it becomes blacklisted [21]. Furthermore, URL blacklists can only cover part of spam URLs, and thus some spam campaigns may escape detection. Second, some URL blacklists generate false positive errors as they only check the hostname component of a URL, instead of the whole URL. For example, the URL shortening service http://ow.ly is listed on the URIBL blacklist [13] because it is greatly abused by spammers. Although http://ow.ly/6eAci is a benign URL that redirects to a CNN's report of Hurricane Irene, it is blacklisted by URIBL based on the hostname. Third, the URL feature generates false negative errors. For instance, consider a campaign that advertises a benign website in an aggressive spamming way. The spammer manipulates multiple accounts to post duplicate tweets about the website. The URL feature cannot classify the tweets as a spam campaign since the website URL is benign and not blacklisted. The first two disadvantages may be overcome by improving blacklisting process, but the third cannot be fixed by merely using the URL feature. Thus, the other features, such as collective posting content and behavior, should also be included. This paper improves the existing work by introducing new features. The details of classification features are covered in Section 4.1.

2.2 Scope of This Paper

A variety of spam attacks exist on Twitter. This paper solely focuses on characterizing and detecting large-scale spam campaigns conducted on Twitter. The definition of spam in this paper is spreading malicious, phishing or scam[3] content in tweets. Spammers may carry different purposes, but spam campaigns exhibit a shared feature that, they either create or compromise a large number of Twitter accounts to spread spam to a wide range of audience. Our work does not screen individual tweets to detect spam, and may miss small spam campaigns[4]. As a complement to existing spam detection methods, the main contribution of this paper is detecting multiple related spam tweets and accounts in a robust and efficient way.

Note that after detecting a spam campaign, a site administrator may further classify the involved accounts into Sybil and compromised accounts, and process them accordingly. Here Sybil accounts refer to those created by spammers and exclusively used to post spam tweets. Compromised accounts refer to those used by legitimate users but hijacked by spammers to post spam without the permission of owners. Sybil accounts will be permanently suspended, while the owners

[3] We define a scam as any webpage that advertises a spectrum of solicitations, including but not limited to pornography, online gambling, fake pharmaceuticals.

[4] According to our clustering algorithm presented in Section 3.2, a single tweet may be clustered as a campaign if no other related tweets exist in the dataset.

of compromised accounts can be notified for spamming activities via their registration emails. The differentiation between these two types of accounts is out of the scope of this paper.

3 Characterization

3.1 Data Collection

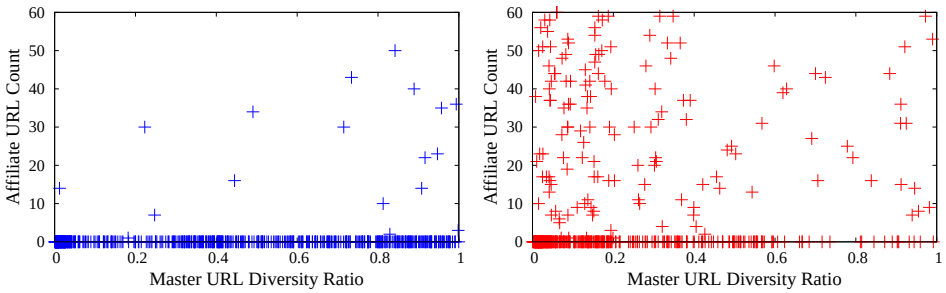
To measure the pervasiveness of spam, we conduct the data collection on Twitter from February to April in 2011. Thanks to Twitter's courtesy of including our test accounts to its whitelist, our dataset accumulates more than 50 million tweets posted by around 22 million accounts. We develop a crawler in PHP which taps into Twitter's Streaming API [12] and Search API [14], respectively. The Streaming API outputs a small proportion of real-time global tweets via random sampling, and constitutes the majority of our dataset. The Search API enables the crawler running specific searches against the real-time index of recent tweets. Since this work studies spam campaigns, we exclude tweets without URLs, and focus on the remaining 8 million tweets with URLs in the dataset. Due to the limited length of tweets, most spam tweets contain URLs to allure users to visit external spam websites. Thus, we assume that tweets without URLs are not spam. As shown in Section 3.2, our clustering algorithm is based on shared URLs.

URL redirection is widely used on Twitter. Normal users apply URL shortening services, such as t.co and bit.ly, to convert arbitrarily long URLs to short ones to better fit in tweets. Spammers also use shortening and other redirection techniques to hide original spam URLs and to avoid blacklist detection. We develop a Firefox extension in JavaScript to automatically visit every URL in the dataset and convert to its final landing URL if redirection is used. Some spammers tend to use long redirection chains that involve multiple hops (such as original URL -> intermediate URL -> ... -> final URL) to hide their traces. The extension records the whole chain, and provides a classification feature.

3.2 Clustering

We develop a clustering algorithm that clusters tweets into campaigns based on shared final URLs[5]. The idea behind the algorithm is that those tweets that share the same final URL are considered related. A tweet is modeled as the <textual content, URL> pair. A given campaign, c_i, is denoted by a vector $c_i =< u_i, T_i, A_i >$, where u_i is the shared final URL i for the campaign, T_i is the set of tweets containing u_i, and A_i is the set of accounts that have posted tweets in T_i. Let C denote the current set of campaigns. The clustering procedure iteratively chooses without replacement an arbitrary tweet t in the dataset. If the tweet's URL is $u_{i'}$ and $c_{i'} \in C$, then the tweet is added in the campaign

[5] The subsequent campaign classification applies a variety of features, including both content and URL of tweets. More feature details are presented in Section 4.1.



(a) Legitimate Campaigns (b) Spam Campaigns

Fig. 1. URL Statistics of Campaigns

by updating $T_{i'} = T_{i'} \cup \{t\}$. If t's account, a, is also new, then an update $A_{i'} = A_{i'} \cup \{a\}$ is also performed. If $c_{i'} \notin C$, then a new campaign $c_{i'}$ is created and $C = C \cup \{c_{i'}\}$ is updated.

In our implementation, we store the dataset in MySQL database, and create a table for the clustering result. Every URL string is hashed, and the hash value is set as the table index. Two URL strings are compared by their indexed hash values to improve the clustering performance. Once complete, the dataset includes 5,183,656 campaigns. The largest contains 7,350 accounts with 9,761 tweets posted.

3.3 Ground Truth Creation

After campaigns have been clustered, we create a ground truth set containing samples labeled as spam and legitimate campaigns. We select some campaigns from our dataset, manually perform several heuristics tests, and use human expertise to label unknown campaigns. Due to the limited raw data returned by the Twitter API with low privilege, we favor the campaigns associated with a large number of accounts and tweets during the selection process as large campaigns carry abundant collective behavior characteristics. Small campaigns are excluded from our selection.

More specifically, we follow Twitter's spam rules during the manual inspection, and check both collective and individual features of an unknown campaign. First, we inspect the campaign's final URL. A batch script is performed to check the URL in five blacklists: Google Safe Browsing, PhishingTank, URIBL, SURBL and Spamhaus [1, 3, 13, 7, 6]. More details of the blacklist detection will be presented in Section 4.1. If the URL is captured by the first two blacklists, the related campaign is directly labeled as spam without further manual inspection required.

Second, we check the tweet content of the campaign. The human inspects the content to see if (1) it contains spam information, (2) it is unrelated with the URL's web content (namely, the URL is misleading), (3) duplicate or similar

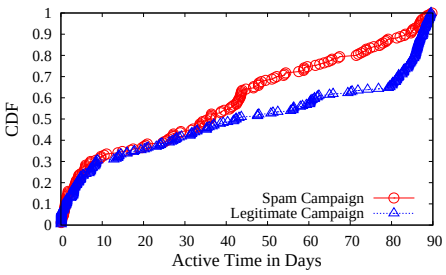


Fig. 2. CDF of Campaign Active Time

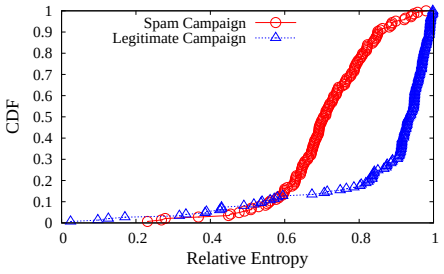


Fig. 3. CDF of Entropy of Posting Inter-arrivals

content is posted via single or multiple accounts. In addition, we also check content-related Twitter properties.

Third we check the automation degree exhibited in the campaign, as automation is a good indicator of spam. The script presents the human inspector with the posting device makeup, the median, and the entropy value of the posting inter-arrival timing sequence. The formal description of these features will be detailed in Section 4.1. Aggressive automation may raise the red flag, and influence the human's classification decision for the campaign.

By taking all of the above into consideration, the human inspector reaches the decision to label the campaign as spam or legitimate. In practice, we find out that most spam campaigns carry obvious characteristics of URL and content, making it easy to differentiate them from legitimate campaigns. We acknowledge that we may make mistakes in labeling campaigns, but believe that the error rate is very low. Finally, the ground truth set contains 744 spam campaigns and 580 legitimate ones.

3.4 Campaign Analysis

We now examine the characteristics of spam campaigns and compare with legitimate ones. The data analysis leads to the formal definition of classification features in Section 4.1.

We first discuss using URL statistics to reveal account connection in the campaign. We have observed that accounts in a legitimate campaign are usually run by independent users, while those involved in a spam campaign are often controlled by the same spammer. The URL statistics can provide hints of account connection. For clarity, we first define two terms: master URL and affiliate URL. For a normal URL such as http://biy.ly/5As4k3, affiliate URLs with it can be created by appending random strings as the query component to the URL, such as http://biy.ly/5As4k3?=xd56 and http://biy.ly/5As4k3?=7yfd. The original URL is denoted as master URL. Affiliate URLs help track the origin of click traffic. By assigning every account with a specific affiliate URL, the spammer can evaluate the spamming effect of individual accounts. This trick widely exists

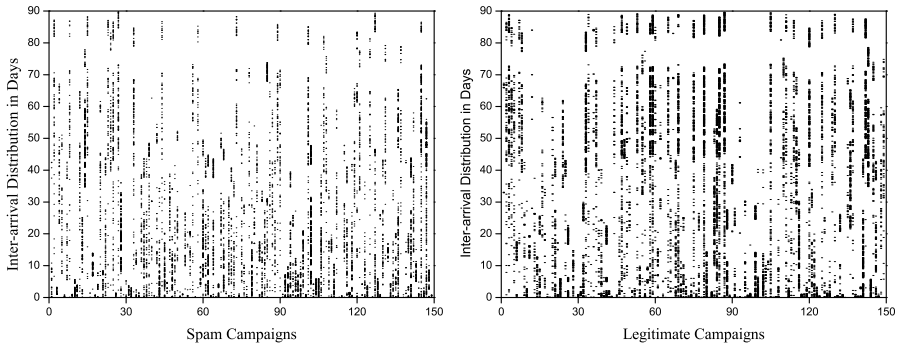


Fig. 4. Inter-arrival Timing Distribution of Campaigns

in online pyramid scams. Frequent appearance of affiliate URLs indicates strong connection among accounts. In contrast, different forms of master URLs indicate account independence. Although the tweets in a campaign share the same final URL, they may have different master URLs, such as http://bit.ly/1wgYxU and http://ow.ly/6jRqX[6]. We define the master URL diversity ratio as the number of unique master URLs over the number of tweets in a campaign. A low ratio indicates the wide usage of affiliate URLs and account dependence, whereas a high ratio indicates the account independence. Figure 1 shows that more than 50% of spam campaigns use affiliate URLs, while only 3.6% of legitimate campaigns contain affiliate URLs. The average master URL diversity ratio of spam campaigns is 0.225, much lower than that of legitimate campaigns, at 0.423.

Now we analyze the temporal properties of campaigns. We define the active time of a campaign as the time span between its first and last tweet in our dataset. We point out a limitation of our dataset as our collection runs for three months while a campaign may exist before and/or after the measured period. While the largest possible active time in our dataset is 90 days, the actual time may be greater. Figure 2 shows the cumulative distribution function (CDF) of active time (in days) of spam and legitimate campaigns. Around 40% of campaigns in both categories have active time less than 30 days. For those longer than 30 days, the average active time of legitimate campaigns is 72.0 days, greater than that of spam campaigns at 59.5 days. Thanks to the workload distribution among accounts, the spamming behavior of an account may be stealthy during its initial stage, and avoid Twitter's detection. It explains the equal proportions of both categories within the 30-day time window. The accumulation of spamming behavior and the increase of campaign size expose spam accounts, and many of them get suspended by Twitter. Beyond the 30-day window, the average active time of spam campaigns is clearly shorter than that of legitimate ones. However, more efforts need to be made to detect and eliminate spam campaigns in the initial stage for damage control.

[6] All the URLs in this paragraph lead to http://twitter.com.

The burstiness characterizes the overall workload distribution of spam campaigns. Figure 4 plots the inter-arrival timing pattern of two categories of campaigns. Due to space limit, each category contains 150 individual campaigns. Each campaign is represented by a vertical strip. Each tweet corresponds to a tiny horizontal segment in the strip, and a block of intensive strips represent a burst of tweets in the campaign. A large number of spam campaigns show burstiness in the early stage. Some spammers aim to achieve the spamming goal in a quick way, and direct spam accounts to massively post tweets. Although the workload is distributed to multiple accounts, the collective inter-arrival pattern can reflect the overall campaign workload. The gradual suspension of spam accounts causes the stagnation in the late stage [7]. Many legitimate campaigns tend to take a while to grow up, and demonstrate burstiness in the late stage. A popular legitimate campaign generates the epidemic effect by making more users tweet about it, spreading to even the larger audience.

Entropy is another temporal property that detects periodic or regular timing of posting patterns in a campaign. In Information Theory, the entropy rate is a measure of the complexity of a random process [19]. A high entropy rate indicates a random process, whereas a low entropy rate indicates a regular one. More theoretical proofs can be found in our previous work [18]. To get relative entropy for every campaign, we normalize entropy values via dividing them by the maximum value of the campaign in the ground truth set. Figure 3 plots the CDF of relative entropy of posting inter-arrivals of both categories. The behavior of auto programs (namely Twitter bots) is often less complicated than that of humans, which can be measured by low entropy rate. In the range between [0.6, 1], the relative entropy of the legitimate category is clearly higher than that of the spam category. The majority of spam campaigns (and a large proportion of their accounts) run auto devices to post, driven by regular or pseudo-random timers. In contrast, tweets in legitimate campaigns are mostly posted by humans. The intrinsic irregularity and complexity of human behavior generates a higher entropy rate. We also find an interesting fact that, a small part of spam campaigns post their tweets manually, generating high entropy. We speculate it is either a form of click farm on Twitter, or some spammers are not professional, and do not run auto programs to tweet.

Finally we discuss a dilemma spammers often face, namely reusing spam accounts. If multiple tweets in the campaign are posted by an account, considering the tweets share the same final URL, the account exhibits the evidence of duplicated posting, which is an indicator of spam. We introduce the account diversity ratio feature. For normalization, this feature is defined as the number of accounts in the campaign over that of tweets. Figure 5 plots the CDF of this feature of both categories. Spammers want to operate accounts in a stealthy way, which requires individual accounts to post few tweets. In reality, it costs effort to get followers to a spam account, and the number of "influential" accounts owned by a spammer is limited. Thus, the spammer tends to repeatedly use accounts to

[7] We re-visit the accounts involved in a spam campaign, and observe that a high proportion of these accounts have been suspended by Twitter.

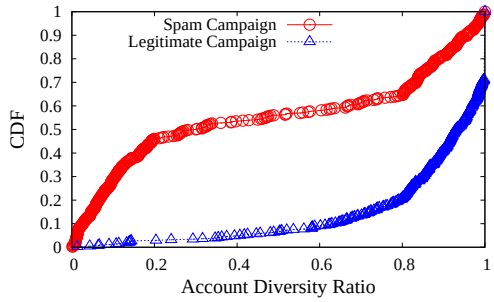


Fig. 5. CDF of Account Diversity Ratio of Campaigns

post duplicate spam, causing the low ratio. The figure clearly demonstrates that, the account diversity ratio of legitimate campaigns is much higher than that of spam campaigns. In particular, about 28.8% legitimate campaigns have the ratio as 1, meaning every tweet in the campaign is posted by a unique account. The average ratio of legitimate campaigns is 86.4%, while that of spam campaigns is 45.0%. It further suggests that, legitimate campaigns have stronger account independence than spam campaigns.

4 Classification

In this section, we first present the design philosophy of the classification system. In particular, we formally describe classification features and introduce semantic similarity to detect duplicate content in a campaign. Then, we implement the classifier based on the Random Forest algorithm.

4.1 Classification Features

The classification involves a variety of features, ranging from individual tweet/account levels to a collective campaign level. No single feature is capable of discriminating effectively between spam and legitimate campaigns. Here we introduce these features used in our classification, and later the machine learning algorithm will decide the importance (namely weight) of the features during the training, which is shown in Section 5.1.

Tweet-level Features. We start with tweet-level features, as tweets are the atomic unit of Twitter. A tweet is modeled as the <textual content, original URL> pair.

Spam Content Proportion. Some spam tweets carry explicit spam information, such as "buy Viagra online without a prescription" and "get car loan with bad credit". We create a list of spam words with high frequency on Twitter to capture spam content based on our observation and some existing lists of spam trigger words [5, 2]. The tweet text is tokenized into words which are

further checked in the spam word list. This feature is defined as the number of spam words over the total word number in a tweet .

URL Redirection. We develop a Firefox extension to check the original URL in the tweet. If URL redirection is used, it records the final landing URL. By recording the status change in the browser's address bar, the extension logs the whole redirection chain (such as original URL -> intermediate URL -> ... -> final URL). Besides the binary redirection flag, hop number also serves as a useful feature. Spammers tend to use multi-hop redirection to hide spam origins and avoid URL blacklists.

URL Blacklisting. We check the final URL in five blacklists including Google Safe Browsing, PhishingTank, URIBL, SURBL, and Spamhaus. Google Safe Browsing checks URLs against Google's constantly updated lists of suspected phishing and malware pages. PhishingTank focuses on phishing websites. The mechanisms of URIBL, SURBL and Spamhaus are similar. They contain suspicious websites that have appeared in spam emails. If the URL appears in any of the blacklists, the feature is set as true. As the tweets in a campaign share the same final URL, this operation only needs to be performed once.

Account-level Features. We also collect data of Twitter accounts involved in a campaign by calling Twitter's REST API [10], and present account-level features to characterize accounts.

Account Profile. An account has a self-introduction profile consisting of a short description text and homepage URL. We check whether the description contains spam or the URL is blacklisted.

Social Relationship. Tweets of an account can only be delivered to its followers. To achieve a wide influence, the spammer needs to accumulate a large number of followers. However, normal users are unlikely to follow spam accounts. A common trick shared by spammers is following a great number of users (either targeted or randomly selected), and expecting some of them to follow back. Many spam victims blindly follow back "spammer friends" without carefully checking those suspicious accounts. For an account, we calculate its friend count, follower count, and the ratio between them.

Account Reputation. Extended from the previous feature, we have observed that users are likely to follow "famous" accounts. This feature is calculated and normalized as $follower\ count/\ (follower\ count + friend\ count)$. A celebrity usually has many followers and few friends[8], and its reputation is close to 1. However, for a spammer with few followers and many friends, its reputation is close to 0.

Account Taste. Intuitively, the account chooses whom to follow (namely, friends), and this reflects its "taste". If it follows spammers, its "taste" is bad. By doing this, it helps spread spam to more users, making itself a "supporter" of spammers. This feature is defined as average *Account Reputation* of all the friends of the account.

[8] For example, @Yankees, the official Twitter account of New York Yankees, has 400,000 followers and only 29 friends.

Lifetime Tweet Number. Spam accounts may get suspended for aggressively posting spam. Due to the short lifetime, averagely spam accounts may post fewer tweets. This feature shows the number of tweets an account has posted in lifetime when it is visited by our crawler.

Account Registration Date. Spammers may frequently create new accounts to replace suspended ones. Many spam accounts in our measurement have been created recently.

Account Verification. Twitter verifies accounts for celebrities and organizations. It is difficult for spammers to acquire verified accounts. This binary feature shows whether the account is verified or not.

Account Protection. For user privacy, an account that opts in the protection option makes its tweets invisible to general public, and only visible to approved followers. The option conflicts with the purpose of spreading spam to the wide audience, and may not be adopted by spam accounts.

Campaign-level Features. Collective features may reveal the characteristics of spam campaigns that cannot be observed through individual features. At last we present the campaign-level features as follows. The features of the account diversity ratio, the original URL diversity ratio, the affiliate link number and the entropy of inter-arrival timing have been explained in Section 3.4.

Hashtag Ratio. Spammers often hijack trending hashtags and append them to unrelated spam tweets to increase the chance of being searched and displayed. The feature is defined as the number of hashtags in the tweets over the number of tweets of the campaign.

Mention Ratio. Another trick spammers often play is using @mention to deliver spam to targeted users even without the existing social relationship. The feature is defined as the number of mentions in the tweets over the number of tweets of the campaign.

Content Self-similarity Score. A spam campaign may contain similar tweets created by spam content templates. Users in a legitimate campaign usually contribute content individually, and may not show a strong self-similarity. This feature measures the content self-similarity of the campaign. The details are presented in Section 4.2.

Posting Device Makeup. Twitter supports a variety of channels to post tweets, such as web, mobile devices, and 3rd-party tools. The 8 million tweets in our campaign dataset are posted by 44,545 distinct devices. In the perspective of behavior automation, they can be divided into two categories: manual and auto devices. Manual devices require direct human participation, such as tweeting via web browser or smart-phone. Auto devices are piloted programs that automatically perform tasks on Twitter, and require minimum human participation (such as importing Twitter account information). We manually label the top 100 devices as manual or auto, and use the tdash's API [8] to process the rest. In the campaign dataset, around 62.7% of tweets are posted by manual devices, and the rest 37.3% by auto devices. For every campaign, the script checks its posting devices against the labeled device list, and calculates the proportions of manual and auto devices as the value of posting device makeup.

4.2 Content Semantic Similarity

Spammers may use content templates to create similar spam tweets. Calculating semantic similarity can detect duplicate or similar content in multiple tweets in the campaign. The calculation is challenging as short messages like tweets do not carry as many semantic features as long texts (i.e. email bodies). Our work applies the Vector Space Model [29] that converts tweet texts into vectors, and then calculates the cosine similarity between them. Equation 1 denotes the cosine similarity between two n-dimensional vectors, A and B.

$$cos\_sim = \frac{A \bullet B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{1}$$

For implementation, we use SenseClusters, an open-source program [4], that clusters text messages based on contextual similarity. Given the set of tweets in the campaign, we treat it as a text corpus, and generate a vocabulary by extracting distinct words from the corpus. Then we generate an occurrence matrix with tweets as rows, and words in the vocabulary as columns. The value of $cell_{ij}$ is the TF-IDF (Term Frequency - Inverse Document Frequency) weight [15], which represents the occurrence frequency of $word_j$ in $tweet_i$. As the most intuitive approach, 1st-order similarity detects the number of exact words shared (or overlapped) between tweets. Because spam templates often adopt synonym interchanging for the purpose of obfuscation, our work applies 2nd-order similarity to measure similar tweets. Its general idea is to replace the context with something else that will still represent it, and yet likely provide more information from which similarity judgments can be made [28]. Given the tweet corpus, SenseClusters divides N tweets into K clusters based on the semantic sense on the fly.

We design Equation 2 to measure the self-similarity of the campaign's tweet content.

$$self\_sim\_score = \sum_{i=1}^{K} \frac{cluster_i\_size^{w1} * cluster_i\_sim^{w2}}{K^{w3}}, \tag{2}$$

where K is the number of semantic clusters in the campaign, and $w1$ to $w3$ are weight factors with their tuning procedure presented in Section 5.1.

4.3 Machine Learning Classifier

Our classification problem can be defined as follows. Given a campaign, $c =< u, T, A >$, the classifier determines c as a either spam or legitimate campaign. We choose Random Forest [17] as the machine learning algorithm[9], and train the classifier to make the binary decision. Random Forest serves as an ensemble

[9] The reason is explained in Section 5.2

classifier that includes multiple decision trees. The algorithm combines the bagging idea in [17] and random feature selection in [23] to construct a "forest" of decision trees with controlled variation. Suppose the training set contains M features, and each decision tree only uses $m(<< M)$ features to reach the decision. For classification, an unknown sample is pushed down the tree, and assigned with the class of the leaf node where the sample ends up. More details about decision tree can be found in [25]. Given a specific sample, every decision tree makes a classification decision (either spam or legitimate campaign in our case), and Random Forest applies the majority voting of all the trees to reach the final decision.

5 Evaluation

In this section, we first train the classifier. Then, we evaluate the accuracy of our classification system based on the ground truth set.

5.1 Training

As described in Section 3.3, our ground truth set consists of manually labeled campaigns. More specifically, 744 spam campaigns contain around 70,000 accounts and 131,000 tweets, whereas 580 legitimate campaigns contain around 150,000 accounts and 180,000 tweets.

Before training the classifier, we need to determine the content self-similarity feature by tuning the weight factors in Equation 2 with the following method. We choose Decision Tree as the tuner, and the feature represented by the self-similarity score as the only classification feature. We try different combinations of numeric values of $w1$ to $w3$. In every test round, a combination generates a different self-similarity score for a campaign in the ground truth set. The decision tree associates the self-similarity feature with the root as it is the only feature in the classification, and calculates the best split between spam and legitimate campaigns. The combination of $(w1 = 0.8, w2 = 0.5, w3 = 1)$ generates the highest overall accuracy on the ground truth set, and is chosen for Equation 2. According to the algorithm of the SenseClusters, the cluster similarity score assigned is also no greater than 1. Note that $w1$ and $w2$ are decimal fractions, and they add more weights to the cluster size and cluster similarity. Furthermore, $w2$ makes cluster similarity more important than cluster size, as $w2$ is less than $w1$.

5.2 Cross Validation

By calculating the values of the features described in Section 4.1, a feature vector is generated for each campaign. Weka supports a collection of machine learning algorithms for classification, including mainstream categories of Bayes, trees and so on [22]. We try multiple algorithms in each category, list and compare performance results for the top classifiers with accuracy greater than 80% in Table

Table 1. Algorithm Performance Comparison

Feature	Accuracy (%)	FPR (%)	FNR (%)
Random Forest	94.5	4.1	6.6
Decision Table	92.1	6.7	8.8
Random Tree	91.4	9.1	8.2
KStar	90.2	7.9	11.3
Bayes Net	88.8	9.6	12.4
SMO	85.2	11.2	17.6
Simple Logistic	84.0	10.4	20.4
Decision Tree	82.8	15.2	18.8

1. For each classifier, we use Cross Validation with ten folds to train and test it over the ground truth set [26]. The dataset is randomly partitioned into ten complementary subsets with equal size. In each round, one out of ten subsets is retained as the test set to validate the classifier, while the remaining nine subsets are used as the training set to train the classifier. The individual results from ten rounds are averaged to generate the final estimation.

Table 1 lists three metrics for evaluating the classification performance sorted on accuracy. Considering the confusion matrix with spam campaigns as positive cases, *Accuracy* is the proportion of samples that are correctly identified, *False Positive Rate* (FPR) is the proportion of negatives cases that are incorrectly classified as positive, and *False Negative Rate* (FNR) is the proportion of positives cases that are incorrectly classified as negative. During evaluation, we expect to constrain the FPR low at the cost of accepting the medium FNR. Classifying benign campaigns as spam upsets legitimate users, while missing a small part of spam campaigns is tolerable. Random Forest achieves the highest accuracy, lowest FPR and FNR, and hence is selected as the final classifier for our dataset.

Some features play a more important role than others during the classification. Subsequently, we attempt to evaluate the discrimination weight each feature has. Similar to the tuning method for Equation 2, in each test, we use only one feature to independently cross validate the ground truth set with Decision Tree[10]. The one with the highest accuracy may be considered as the most important feature. Table 2 presents the performance results of the top 10 features, which are also sorted on accuracy. The Account Diversity Ratio feature has the highest accuracy at 85.6%. Technically this one is not difficult to bypass, because spammers could use a large amount of accounts to distribute the workload and lower the ratio. However, spam accounts with limited normal followers cannot generate the satisfying propaganda. We speculate that, in reality, spammers tend to repeatedly use "influential" accounts to deliver spam to a wide audience. The Timing Entropy feature captures the intrinsic complexity of human behavior, that is difficult for bot accounts to bypass. However, many

[10] Random Forest transforms to Decision Tree in the case of single-feature classification. There is only one decision tree to build, and the single feature is associated with its root.

Table 2. Feature Performance Comparison

Feature	Accuracy (%)	FPR (%)	FNR (%)
Account Diversity Ratio	85.6	16.2	13.0
Timing Entropy	83.0	9.5	22.8
URL Blacklists	82.3	3.2	29.0
Avg Account Reputation	78.5	25.6	18.3
Active Time	77.0	16.2	28.3
Affiliate URL No	76.7	9.6	34.0
Manual Device %	74.8	10.3	36.8
Tweet No	75.4	28.6	21.5
Content Self Similarity	72.3	33.7	23.0
Spam Word Ratio	70.5	25.8	32.4

spam campaigns involve manual accounts (probably in the form of click farm), that generate the high FNR at 22.8% for the feature.

We are particularly interested in the performance of the URL Blacklist feature, as it is used as the only feature for spam campaign detection in some existing work [21]. We present the performance comparison between our work based on Random-Forest-based classifier that applies multiple features and the previous work based on the single blacklist feature. Blacklists are haunted by the inevitable lag effect, and cannot include all spam sites "in-the-wild". Besides, blacklists cannot detect duplicate spamming over multiple accounts. These factors generate a high FNR at 29.0%. By using multi-dimensional features, our classifier manages to capture more spam campaigns that would have been missed by the blacklist feature, and lowers the FNR to 6.6%. The low FPR of the blacklist feature is caused by the fact that, some blacklists only check the hostname of URL, and mis-classify some benign web pages hosted by the blacklisted websites. The FPR of our approach (4.1%) is slightly higher than that of the blacklist feature (3.2%). Most importantly, our approach improves the accuracy from 82.3% to 94.5%.

6 Conclusion

Spam haunts social networks, as social relationship facilitates spam spreading. Conventional spam detection methods check individual accounts or messages for the existence of spam. In this paper, we exploit the collective detection approach to capturing spam campaigns with multiple accounts. Our work uses the features combining both content and behavior to distinguish spam campaigns from legitimate ones, and build an automatic classification framework. Our work can be applied to other social networks by integrating application-specific features. Spam detection is an endless cat-and-mouse game. As spamming methods may evolve in the future, some features may be added or replaced with new ones, and the classifier should also be re-trained with the up-to-date ground truth dataset.

Bibliography

[1] Google safe browsing api. http://code.google.com/apis/safebrowsing/ [Accessed: Aug. 27, 2011].

[2] The list of email spam trigger words. http://blog.hubspot.com/blog/tabid/6307/bid/30684/The-Ultimate-List-of-Email-SPAM-Trigger-Words.aspx [Accessed: Apr. 15, 2012].

[3] Phishtank, join the fight against phishing. http://www.phishtank.com/ [Accessed: Aug. 27, 2011].

[4] Senseclusters. http://senseclusters.sourceforge.net/ [Accessed: Sept. 2, 2011].

[5] Spam words by wordpress. http://codex.wordpress.org/Spam_Words [Accessed: Apr. 15, 2012].

[6] The spamhaus project. http://www.spamhaus.org/ [Accessed: Aug. 27, 2011].

[7] Surbl. http://www.surbl.org/lists [Accessed: Aug. 27, 2011].

[8] tdash's api of twitter applications statistics. http://tdash.org/stats/clients [Accessed: Sept. 6, 2011].

[9] Twitter blog: Your world, more connected. http://blog.twitter.com/2011/08/your-world-more-connected.html [Accessed: Aug. 17, 2011].

[10] Twitter rest api resources. https://dev.twitter.com/docs/api [Accessed: Aug. 30, 2011].

[11] The twitter rules. http://support.twitter.com/entries/18311-the-twitter-rules [Accessed: Aug. 17, 2011].

[12] Twitter's streaming api documentation. https://dev.twitter.com/docs/streaming-api [Accessed: Aug. 30, 2011].

[13] Uribl, realtime uri blacklist. http://http://www.uribl.com/about.shtml [Accessed: Aug. 27, 2011].

[14] Using the twitter search api. https://dev.twitter.com/docs/using-search [Accessed: Aug. 30, 2011].

[15] A. Aizawa. The feature quantity: an information theoretic perspective of tfidf-like measures. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 104–111, 2000.

[16] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *Proceedings of the CEAS 2010*.

[17] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[18] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is tweeting on twitter: human, bot or cyborg? In *Proceedings of the 2010 Annual Computer Security Applications Conference*, Austin, TX, USA, 2010.

[19] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 2006.

[20] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th annual conference on Internet measurement*, pages 35–47, 2010.

[21] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 27–37, 2010.

[22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18, 2009.

[23] T. K. Ho. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20:832 –844, aug 1998.

[24] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage. Spamalytics: an empirical analysis of spam marketing conversion. *Commun. ACM*, 52:99–107, September 2009.

[25] R. Kohavi and R. Quinlan. Decision tree discovery. In *In Handbook of Data Mining and Knowledge Discovery*, pages 267–276. University Press, 1999.

[26] G. McLachlan, K. Do, and C. Ambroise. *Analyzing microarray gene expression data*. Wiley, 2004.

[27] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th international conference on World Wide Web*, pages 83–92, 2006.

[28] T. Pedersen. Computational approaches to measuring the similarity of short contexts : A review of applications and methods. *CoRR*, abs/0806.3787, 2008.

[29] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, November 1975.

[30] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, 2010.

[31] M. Xie, H. Yin, and H. Wang. An effective defense against email spam laundering. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 179–190, 2006.