A Social-Spam Detection Framework

De Wang, Danesh Irani, and Calton Pu
Georgia Institute of Technology
Atlanta, Georgia 30332-0765
{wang6, danesh, calton}@cc.gatech.edu

ABSTRACT

Social networks such as Facebook, MySpace, and Twitter have become increasingly important for reaching millions of users. Consequently, spammers are increasing using such networks for propagating spam. Existing filtering techniques such as collaborative filters and behavioral analysis filters are able to significantly reduce spam, each social network needs to build its own independent spam filter and support a spam team to keep spam prevention techniques current. We propose a framework for spam detection which can be used across all social network sites. There are numerous benefits of the framework including: 1) new spam detected on one social network, can quickly be identified across social networks; 2) accuracy of spam detection will improve with a large amount of data from across social networks; 3) other techniques (such as blacklists and message shingling) can be integrated and centralized; 4) new social networks can plug into the system easily, preventing spam at an early stage. We provide an experimental study of real datasets from social networks to demonstrate the flexibility and feasibility of our framework.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval - Information filtering; I.5.2 [**Pattern Recognition**]: Design Methodology - Classifier design and evaluation

General Terms

Algorithm, Experimentation

Keywords

Social-spam, Framework, Detection, Classification

1. INTRODUCTION

Social networks are growing at an alarming rate, with networks like Facebook and Twitter attracting audiences of over 100 million users a month, they are becoming an important medium of communication. This in turn attracts spammers to the social networks as well, causing an increase in the incidence of social spam.

Social spam is low-quality information on social networks that is similar to email spam in that it is unsolicited bulk messages that users do not ask for or specifically subscribe to. Such spam, is a nuisance to people and hinders them from consuming information that is pertinent to them or that they are looking for. Individual social networks are capable of filtering a significant amount of the spam they receive, although they usually require large amounts of resources (e.g, personnel) and incur a delay before detecting new types of spam.

We propose a social spam detection framework which can be used by any social network to detect spam. The framework will avoid the need for each social network to build their own spam detection mechanism and hire anti-spam personnel. Using this framework, once a new type of spam is detected on one network, it can automatically be identified on the other networks as well. In addition, new social networks can quickly protect their users from social spam by using the framework.

The social-spam detection framework can be split into three main components. Figure 1 shows an overview of the system and we provide a brief explanation for each part here: 1) Mapping and Assembly: Mapping techniques are used to convert a social network specific object into a framework-defined standard model for the object (e.g., profile model, message model, or webpage model). If associated objects can be fetched based on this object, it is assembled here; 2) Pre-filtering: Fast-path techniques (e.g., blacklists, hashing, and similarity matching) are used to check incoming objects against known spam objects; 3) Classification: Supervised machine learning techniques are used to classify the incoming object and associated objects. We use a Bayesian technique to combine the classification results into spam or non-spam.

More concretely, we make the following contributions:

- Build a social-spam detection framework to filter spam on multiple social networks. We build the three main components of the system and demonstrate the use of the system on data from Twitter, MySpace, and the WebbSpamCorpus.
- Demonstrate cross social-corpora classification and measure the feasibility of doing so. Namely, we show that we can build a classifier for a particular model on one social network and apply it to another social

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CEAS 2011 - Eighth annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference Sept. 1-2, 2011, Perth, Australia
Copyright 2011 ACM 978-1-4503-0788-8/11/09 ...$10.00.

network. After doing so, we use existing datasets to approximate the accuracy of this technique.

- Demonstrate associative classification or classification in which the results depend not only on the object being classified, but also on objects associated with it. e.g., classification of a message object takes into account classification outcomes of the associated webpage objects that may be linked inside the message. We also measure the feasibility of this technique.

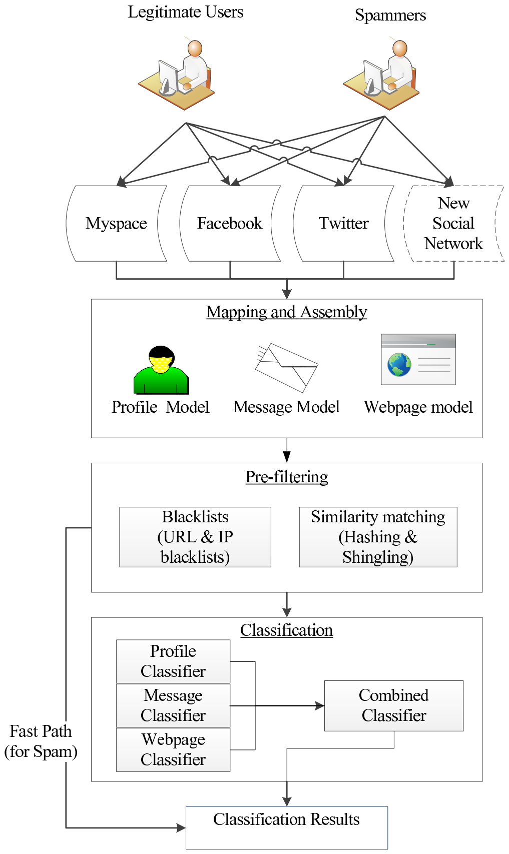


Figure 1: Overview of the spam detection framework

The remainder of the paper is organized as follows. We motivate the problem further in Section 2. Section 3 provides the structure and implementation of the framework. Section 4 describes the experimental setup used to evaluate our framework and Section 5 presents our experimental results. We introduce related work in Section 6 and conclude the paper in Section 7.

2. MOTIVATION

With the rise of social networks as an important medium of communication, spammers have increasingly targeted social networks with spam. In most social networks, spammers can send spam to other users in a number of ways, such as messages, friend requests, wall posts, tweets, and profiles. In most cases spammers can also include links to a website where the user will take another action.

Facebook, Twitter, MySpace, and other major social networks employ dozens of people to fight spam on their network [19]. Most of these social networks use collaborative filtering (where users report objects that are spammy) and behavioral analysis (where logs of interactions are used to detect spamming patterns) to detect spam on their network. Such dynamic methods may be eventually able to detect social spam, but require a non-trivial amount of lag time to accumulate sufficient evidence.

Apart from this, social networks will also employ classification based techniques which use labeled training data to find similar occurrences of spam on the social network. Due to the evolving nature of spam [21, 15], these classification based techniques need to be retrained and adapted to newer spam [22].

Although techniques to propagate spam may vary from one social network to another, due to specificities of each social network, anecdotal evidence suggests that spam generally fall into the category of pharmaceutical, pornographic, phishing, stocks, and business promotion campaigns. Botnets have already been shown to use templates to send varying spam campaigns messages (similar to the campaigns previously mentioned) [18] to different targets. With techniques to distribute a legitimate message across multiple social networks already implemented [1, 2], it is only a matter of time before botnets and spammers also employ such techniques.

We propose a social spam detection framework that uses general models of a profile, message, and webpage model to perform classification across social networks. Misclassifications or feedback via other methods can be used to update the classification models which apply across social networks. This will allow new types of spam detected on one social network to be detected across social networks, and also reduce the burden of the social network spam teams responsible for keeping the classifiers updated. Further, new social networks which do not have any spam detection solutions can use the social spam detection framework to protect their users from spammers.

3. SOCIAL-SPAM DETECTION FRAMEWORK

In this section we present the social spam detection framework. An overview of the framework is shown in Figure 1 and we present the three main parts in the following subsections.

3.1 Mapping and Assembly

To build a framework that is social network agnostic, we have to create a standard model for the objects within the social network. We define a model of an object as a schema containing the most common attributes of the object across social networks. Once a model is defined, we need to map incoming objects from the social networking into objects of the model. We discuss both these steps in more detail below.

3.1.1 Models

Our framework defines three models representing the most important objects in social networks, namely: profile model, message model, and web page model. We omit other models, as they are not required to demonstrate the feasibility of the framework.

The profile model we defined has 74 attributes and is derived from the Google Open Social Person API [3]. The attributes we selected cover attributes most commonly used in user profiles across websites like Facebook, MySpace, Twitter, and Flickr.

The message model we defined has 15 attributes based on common attributes used in messages – such as “To”, “From”, “Timestamp”, “Subject”, and “Content”. We also include in the message model a few attributes which would be common for a social-network to have and also found in e-mail messages, e.g., “Sender-IP”, and other header attributes.

The web page model we defined has attributes based on common HTTP session header information (based on work done by Steve et al. [29]) and content. For example, “Connection”, “Content-length”, “Server” and “Status” et al. are common features in HTTP session header. For the content of web pages, we extracted visible text (non-HTML tags) from them and focused on text classification.

A model is akin to a class, and an object is an instance of the model (or class). All models are stored in XML, so they are extensible and can be modified easily.

Data Types: An attribute can be one of four types. These types are standard attribute types found in many machine learning implementations, namely: Numerical, String, Categorical (Nominal), and Date. An snippet of the person model definition is shown in XML 1.

```
<model type="person">
  <attribute>
    <attributeName>AboutMe</attributeName>
    <attributeType>String</attributeType>
  </attribute>
  <attribute>
    <attributeName>Age</attributeName>
    <attributeType>Numerical</attributeType>
  </attribute>
  ...
</model>
```

XML 1: Definition snippet of the “person” model.

3.1.2 *Mapping*

Mapping transforms incoming social network objects into the respective object model in the framework. This mapping is mostly done automatically by providing to the framework a list of incoming attributes and their attributes in the respective model. These are specified in an XML file due to easy updatability and simplicity.

Name mappings are the simplest to handle and type mappings are also straight-forward except in illegal cases, which we disallow (e.g., “Date” to “Categorical”). For some data types such as categorical type, we need to specify the mapping for each value in the domain of the data type. The handling of semantic mapping is done by manual code written within a special XML tag to perform the necessary conversion. An example of name mappings is shown in Table 1 (shown in table format for simplicity).

3.1.3 *Assembly*

Assembly is the process of probing each model object for associated objects and then subsequently fetching those model objects. For example, if we are dealing with a message object and the content contains URLs, we fetch the web pages associated with those URLs and create web page objects which are then assembled together with the message object. This additional information is often critical for spam detection as it can provide a rich source of information for the further stages.

Table 1: The name mapping between Twitter profiles and our profile model. Empty mappings are omitted.

Twitter Profile	Profile Model
Id	Id
Name	NickName
Location	CurrentLocation
Description	AboutMe
Profile_Image	ProfileImage
Url	ProfileUrl

3.2 Pre-filtering

In order to reduce classification cost, we adopt fast-path techniques to quickly filter out previous classified or similar spam in incoming social network objects. Some of these techniques involve:

- Blacklists: lists of entries, such as URL, DNS, and IP address, which are to be immediately rejected. Entries are added to these lists due to prior spamming or bad behavior, and thus it is expected that objects which contain such entities should be rejected.
- Similarity matching: Hashing and shingling can be used to quickly calculate similarity against previous spammy entries. The number of previous spammy entries an object is checked against can be limited in order to avoid high lookup costs.

These techniques may have shortcomings due to their lagtime in detecting new spam, although their significantly improve time taken to classify an object as spam or non-spam.

3.3 Classification

We build one classifier for each model and use over 40 different types of supervised machine learning classifiers, including standard algorithms such as naïve Bayes [23], Support Vector Machine (SVM) [9] and LogitBoost [6, 10].

An incoming objects to be classified can have associated objects which will be retrieved in the Assembly stage (see Section 3.1.3). For instance, a profile is passed to the profile classifier for classification followed by associated messages being passed to message classifier to do the classification. If the message object contains a URL (such as a Tweet), then the associated object will be a web page object which will be passed to the web page classifier. This process is illustrated in Figure 2. We apply combination strategies to the results of the classifiers after obtaining all results.

After the classifier for each model involved return a decision, it is passed on to the combiner. There are four different combination strategies available for us to adapt in our framework: AND strategy, OR strategy, Majority voting strategy, and Bayesian strategy. AND strategy classifies an object as spam if all classifier, for each model, classifies it as spam. OR strategy classifies an object as spam if any classifier, for each model, classifies it as spam. Majority voting strategy classifies the object as spam only when majority of classifier, for each model, classifies it as spam. Bayesian strategy is a slightly modified version of a strategy from previous research

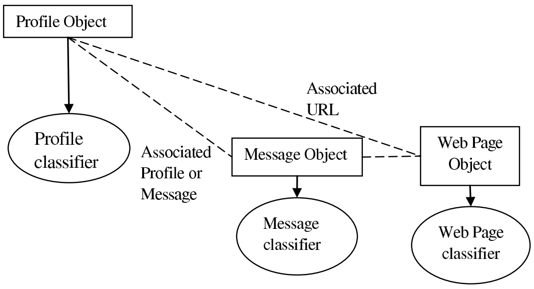


Figure 2: Using associated objects to assist in classification.

on creating an anti-spam filter combination framework for text-and-image emails [5]. We use a subscript i to distinguish different models and t to denote incremental learning cycles at time t. Suppose we receive a object x and ω is the class associated with x, either being spam or legitimate. Then, assuming a hidden variable Z for an event to select one model, a probability for a class ω given $x, P(\omega|x)$, can be expressed as a marginal probability of a joint probability of Z and ω.

$$P(\omega|x) = \sum_i P(\omega, Z_i|x) = \sum_i P(\omega|Z_i, x) P(Z_i|x).$$

To express each classifier's confidence given x, we use external knowledge $P(Z_i|x)$. For instance, if a certain classifier model becomes unavailable, we will set the corresponding $P(Z_i|x)$ to be zero. Also if one classifier dominates over other classifiers, one could assign a large probability for the corresponding $P(Z_i|x)$.

Most data types are supported directly by the classifiers we use, except for the String data type. We use bag of words to change the string into a list boolean attribute (where each boolean attribute represents the presence or absence of a word) after using stemming and removing stop words.

Before classification, we represent each object (or model instance) as a attribute vector f of n attributes: $\langle f_1, f_2, ..., f_n \rangle$. All of attributes are boolean; hence, if $f_i = 1$, the attribute is present in a given object; otherwise, the attribute is absent in a given object.

4. EXPERIMENTAL PLAN

In this section we first discus the datasets we use to demonstrate our framework, followed by the implementation of our framework, and finally detail our evaluation plan.

4.1 Datasets

The datasets contain raw profiles or messages from the social networks which are then parsed into their respective XML objects. The XML objects are submitted to the social spam detection framework, where they go through the three main processing components. We use these datasets as sample input to perform experiments and measure the performance of our framework.

MySpace Profile Dataset: We use a previously collected sample of over 1.8 million MySpace profiles [8] from June to September 2006. In addition, approximately 1,500 spam MySpace profiles were also gathered from a previous study [31] on honeypots in MySpace collected in late 2007. We summarize the strategies used to collect the MySpace profiles below:

- Legitimate Top 8 Crawl — starting with a seed list of random (legitimate) profiles, the top 8 most popular friends were crawled in a breath first search manner. This resulted in a collection of over 890,000 connected profiles.
- Random Crawl — profiles were crawled by generating random UserId's and retrieving the profile represented by that user. This resulted in a collection of over 960,000 profiles.
- Honeypot Spam — Social honeypot accounts were configured across the U.S. and were used to crawl profiles that initiated contact with them and were identified as spam accounts.

Twitter Profile, Message, and Web Page Datasets: A previous study [17] on Twitter collected over 900,000 Twitter users, over 2.4 million Tweets and fetched any links in the Tweets. Twitter users are represented in the profile model, Tweets can be represented in the message model, and webpages associated with the links in the web page model. These tweets were gathered by querying the top trending topics every minute and represent a over 600 topics over a span of November 2009 to February 2010. Twitter users and Tweets (messages) marked as suspended or removed due to Twitter terms of service violations were marked as spam and there were over 26,000 Twitter such users and 138,000 such Tweets.

TREC Email Datasets: We include the use of emails in our social-spam detection study to have additional data for the message model. We use the TREC 2007 [26] corpus which contained over 50,000 spam emails and over 25,000 legitimate emails. Spam messages in this dataset was collected from various email honeypots, with legitimate messages being donated by various users.

Webb Spam corpus and WebBase dataset: The Webb Spam corpus [28] is a collection of nearly 350,000 spam web pages, crawled from spam links found in email messages between November 2002 to January 2006. As there were no legitimate pages in the Webb Spam Corpus, we augmented it with a dataset from the WebBase Web Page Repository [14]. We downloaded and used December 2005 crawl of the WebBase corpus and used a stratified random sample of over 392,000 legitimate web pages.

4.2 Experiment Implementation

4.2.1 *Mapping and Assembly*

For each of the datasets earlier presented, we create a mapping file from attributes in the dataset to attributes in the model. The assembly process takes care of retrieving associated object models if required.

To be specific, the mapping and assembly in relation to the profile model is demonstrated by mapping/assembling MySpace and Twitter (users) to the profile model. The mapping and assembly in relation to the message model is demonstrated by mapping/assembling TREC and Twitter (tweets) to the message model, whereas the mapping and assembly in relation to the webpage model is demonstrated by mapping/assembling Twitter (webpages linked in tweets) and WebbSpamCorpus/WebBase to the web page model.

4.2.2 Pre-filtering

Due to the lack of historic blacklists to apply to our data and the likelihood that using new blacklists on older data would skew the results, we do not use blacklists in our experiments.

4.2.3 Classification

The two core parts of our detection framework are: cross social-corpora classification and associative classification. To evaluate the effectiveness of cross social-corpora classification, we build a webpage model classifier using the WebbSpamCorpus and WebBase dataset, followed by classifying the webpage part of Twitter dataset. Next, for incoming objects, we retrieved their associated objects and use cross social-corpora classification for each object model. For the profile classifier, we use the MySpace profile dataset for training and evaluate the cross classifier on the Twitter profile dataset. For the message classifier, the TREC 2007 dataset is used for training and evaluate the cross classifier on the Twitter message dataset. We then combine the results from web page model and message model classifications, to obtain the result of message classification. Finally, the results of incoming profiles classification are obtained by the combination of the results of message model and profile model classification.

We use over 40 different classifiers implemented in the Weka software package [13]. Weka is an open source collection of machine learning algorithms that has become a standard tool in the machine learning community.

Most data types used in our models can be directly used by classifiers in Weka except for the String data type. As previously mentioned, we use bag of words after using stemming and removing stop words. The StringToWordVector filter performs this transformation for us and includes applying the Snowball Stemming algorithm (variation of the Porter stemmer) and removing a built-in list of stop words.

Once the object is in an amenable form, steps to build a model classifier include resampling and classifier selection. When resampling, we use stratified random sampling to balance the spam and legitimate classes in order to avoid any biases that may arise from imbalanced classes.We evaluate and compare the performance of a number of classifiers based on the F1-Measure and accuracy. More details on the evaluation can be found next.

4.3 Evaluation

We use several criteria evaluate the classifiers' performance, namely F1-Measure and Accuracy. In our framework, F-measure (also F-score) is calculated based on precision and recall. Before introducing the details of precision and recall, we review the relationship between true positive, true negative, false positive, and false positive — shown in Table 2.

Table 2: The relationship between true-positive, true-negative, false-positive and false-positive.

Actual Label	Predicted Label	
	Positive	*Negative*
Positive	True-Positive (TP)	False-Negative (FN)
Negative	False-Positive (FP)	True-Negative (TN)

The definitions of precision(P), recall(R), F-measure(FM), and accuracy(A) for classification are based on above terms, and are given by the following formulas.

$$P = \frac{TP}{(TP+FP)}; \quad R = \frac{TP}{(TP+FN)}$$

$$FM = 2 \cdot \frac{P \cdot R}{P+R}\ ;\ A = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

F1-measure is the Harmonic mean of precision and recall (traditionally F-measure is represented in terms of precision and recall). Accuracy represents the number of instances correctly classified and is equals to the sum of true positives and true negatives divided by the total number of instances. Our goal is to obtain high F-measure and accuracy.

5. EXPERIMENTAL RESULTS

In this section, we present the results of the two core parts of our detection framework, namely the cross social-copora classification and associative classification.

5.1 Cross Social-Corpora Classification

During practical usage of our framework, we expect our classifier models to be built using a number of social networks. Incoming objects from the same social networks or other social networks can then be classified using these models. We evaluate an extreme case of this classification, where we build a classifier using one social-network dataset and test the results using another dataset.

We first show cross social-corpora classification based on web page model as the web page model can be used in conjunction (via associated objects) with both profile and message models in our framework. We will use the web page model classifier to improve the accuracy of the other models.

To build (train) the web page model classifier, we use the WebbSpamCorpus (contains spammy pages) and WebBase (contains legitimate pages). We apply the classifier to labeled web pages associated with Tweets from the Twitter dataset. These datasets previously described in Section 4.1 consist of HTTP session headers and content web pages.

Previous work [30] used HTTP session headers to detect spam for web pages, but based on our datasets we found that HTTP session headers are not robust to temporal differences in the cross-corpora classification. This is likely due to HTTP session headers containing transitory features that become exiting due to the arms-race between spammers and spam-researchers [21, 15]. We therefore perform classification on content of web pages for cross-corpora and cross-temporal datasets.

Using a bag of words approach on the content of web pages results in over 50,000 words (after stripping HTML tags, removing stop words, and removing words which occur less than 10 times). As this attribute set is too large to use practically, we explore the impact of feature set size and corpus sample size on the effectiveness of our classifiers. We vary the feature set size between 1,000 and 10,000, based on the features with most information gain, and varied the corpus sample size similarly, with a unified random sample of spam and legitimate instances to generate an equal class distribution (thereby minimizing the class-specific learning biases). The size of total features in datasets influences the

the size of feature set we choose. After performing this evaluation, we found the majority of our classifiers consistently exhibited their best performance with 7,000 retained features and a corpus sample size of 10,000 instances. Their information gain values are shown in Figure 3. We use these settings for the rest of our experiments involving the web page model.

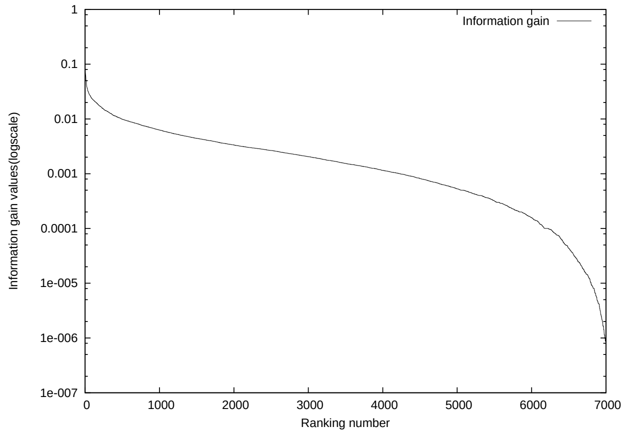


Figure 3: The information gain values of 7,000 features for WebbSpamCorpus and WebBase web pages

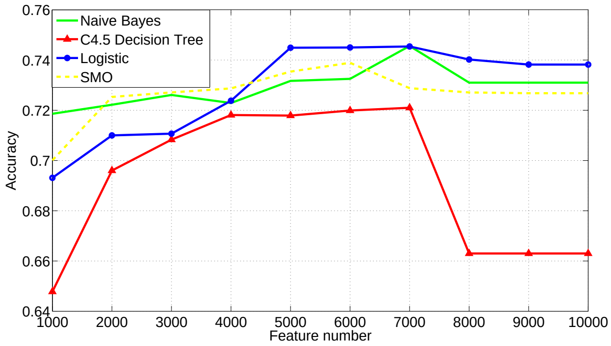
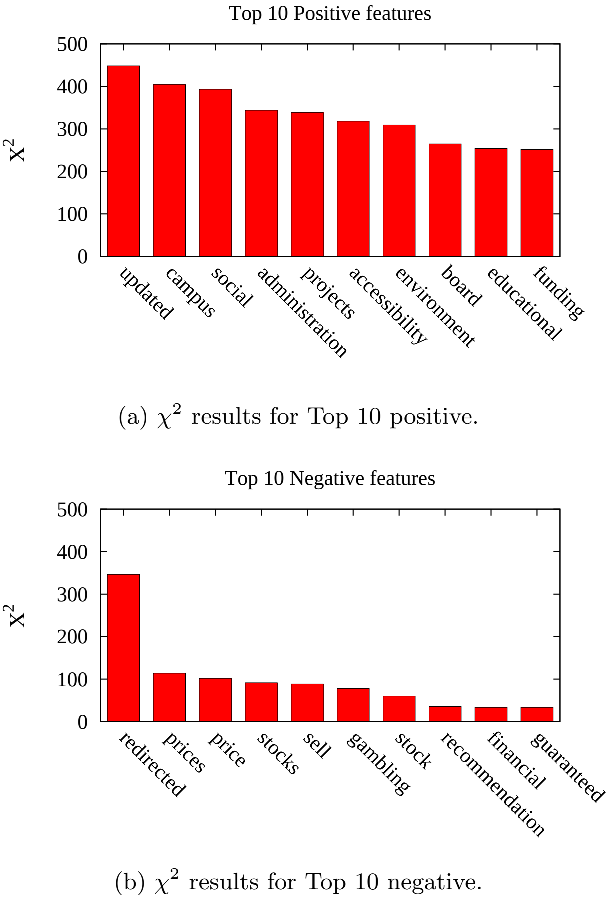


Figure 4: Classifier performance results for Cross-corpus learning on web page model

Using the sample size and feature size above we evaluated the performance of 40 classifiers in Weka. The performance metrics for a sample of the four most effective classifiers from our evaluation are shown in Figure 4. We can see that all of our classifiers preformed adequately, with Naïve Bayes performing the best overall in terms of average accuracy (Table 3 shows the confusion matrix that resulted from the Naïve Bayes classification choosing 7000 features). We also ranked the Top 10 positive and negative attributes by χ^2 Test shown in Figure 5 — positive attributes are those words which appear more in legitimate web pages than web spam. Negative attributes are those words which appear more in web spam than legitimate web pages.

Table 3: The results of Naïve Bayes classifier

	Predicted Legitimate	Predicted Spam
True Legitimate	3286	1714
True Spam	830	4170



(a) χ^2 results for Top 10 positive.

(b) χ^2 results for Top 10 negative.

Figure 5: Top 10 positive and negative attributes for WebbSpamCorpus and WebBase.

To investigate the misclassifications, we manually reviewed the 1,714 misclassified legitimate web pages from Twitter and put them into several categories based on their content (shown in Table 4). Non-text category may contain pictures, flash, video, audio, radio, TV, and other types, which usually contain very little visible text content. Foreign language category is misclassified due to unrecognized words, such as web pages in Japanese, Germany, and Chinese. Short text category mostly is from the messages or comments archive in social networks (e.g. Twitter messages the length of which is under 140). Comments list category is represented by web page which contains a list of comments following an object in social networks like article or blog. Downloading links category includes web pages which link to file downloads. Search engine category contain index pages for search engines like Google or Bing. Shopping/advertisement sites category contains pages which are filled with many descriptions and pictures of products.

The classification of these types of web pages can be fixed by whitelisting the respective sites. The whitelist is from WebBase project and contains about 30,000 legitimate URLs or domain names. After whitelisting these sites, we obtained better results using naïve Bayes classifier. The F1-Measure

Table 4: The Categories of Misclassified Legitimate Web Pages

Categories	Amount	Ratio
Non-text	543	32.7%
Foreign language	366	21.4%
Short text	188	11%
Comments list	51	3%
Downloading links	137	8%
Search engine	57	3.3%
Shopping/advertisement sites	17	1%
Unidentified	355	20.7%

and accuracy achieved 0.894 and 95.9% respectively.

Table 5: The results of naïve Bayes Classifier after whitelisting legitimate sites

	Predicted Legitimate	Predicted Spam
True Legitimate	2890	121
True Spam	561	4242

We performed the same cross social-corpora experiments above on message model. We used the TREC email corpus to build the message model classifier and the Twitter message dataset for testing. The results achieved were not as good as those achieved with the web page model and the reasons are as follows. Firstly, Twitter messages have a limit of 140 characters and are very short. In addition, a large number of messages contain short-hand, abbreviated words, or contain URLs, which reduce the amount of content available for classification. For TREC data, the length of most email messages is longer than 140 and the lexicon is vastly different from Twitter's. The top 10 attributes based on χ^2 test for both datasets are shown in Table 6. Therefore, if we use TREC data to test Twitter message data based on text content, a lot of Twitter messages are misclassified.

Table 6: Top 10 attributes for Twitter message data and TREC data

Rank	TREC	Twitter	Rank	TREC	Twitter
1	org	de	11	stat	write
2	list	en	12	math	click
3	mail	free	13	minim	person
4	wrote	n	14	reproduc	u
5	listinfo	support	15	project	thi
6	code	cc	16	ethz	show
7	mailman	make	17	unsubscrib	la
8	comment	time	18	ch	real
9	post	g	19	guid	h
10	read	watch	20	provid	part

Another cross-corpora experiment we performed was using the profile model. We use the MySpace corpus for building the profile model classifier and the Twitter user dataset for testing. We find that the results are not good as previous achieved, as each Twitter profile only has five attributes: "Picture", "Name", "Location", "Web", and "Bio", whereas a MySpace profile has over 11 attributes. We compared the spam and legitimate profiles on Twitter, expecting that spam profiles would use "Web" URLs and "Bio"s to propagate spam URLs (based on an observation made in a previous study [8]), but we find the percentage of empty values in "Web" URL and "Bio" attributes for spam profile is higher than the percentage of empty values in legitimate profiles. Meanwhile, the length of "Bio" attribute on Twitter also contains a small amount content due to a limit of 160 characters (enforced by Twitter). These reasons all result in performance not being as good as the web page model when training using MySpace profiles to test Twitter profiles.

Table 7: Statistics on "Web" and "Bio" fields in Twitter profile.

	Web and Bio (not empty)	Total	Percentage
Total Profiles	536291	938190	57.16%
Spam Profiles	4675	26818	17.43%

Some of the challenges faced with cross-temporal and cross-dataset issues can be helped by using the associative classification component of our framework to deal with the issues. In the following section, we introduce the associative classification of our framework.

5.2 Associative classification

Associative classification takes advantage of models that may be associated with a model being classified. Previous studies have shown that the integration of diverse spam filtering techniques can greatly improve the accuracy of classification [5]. Spammers are typically interested in advertising, which is usually done via links to websites hosting the spam. Thus, spammers usually post URLs whereas non-spammers post messages or status updates without URLs. To take advantage of the web page model, we use the associated URL information from our message model and profile model. Therefore, taking the message model, as an example, the classification process works as follows.

If a new message arrives with a URL in it, we extract the URL and fetch the associated web page content. Using the web page model classifier, we can then classify the web page and use the result to assist in making a determination of the classification of the message — combination strategies for different classifier results have been described in Section 3.3.

For messages which do not contain URLs, we classify them using the message model classifier. Previous research has achieved accurate classification results using content attributes and user behavior attributes [12, 7]. Under our data constraints, we only perform classification based on the content of messages.

We demonstrate associative classification using the profile model as an example. We randomly choose 3,000 spam and legitimate Twitter profiles, also retrieving the associated messages. For messages, if they have an associated URL, we retrieve it using our crawler. Finally, we obtain all objects needed to help us in the profile classification. They amount to 6,000 profiles, 28,841 messages, and 43,905 URLs (of which 17,211 are non-redirection). Using the WebbSpam corpus and WebBase corpus, the accuracy of our associative classifier is 91%.

Table 8: The results of web page model using naïve Bayes Classifier.

	Predicted legitimate	Predicted spam
True legitimate	10553	622
True spam	931	5105

The confusion matrix of the result is shown in Table 8. We once again manually checked the misclassifications and find they fall into similar categories as previously described. Some web pages contain sparse text or mainly non-text content and some are in foreign languages. The results of twitter messages classification also have two parts: one is for messages which contain URLs, in which we use the content of the messages as well as web page model classifier to help us predict the final classification label for the message. The other is for messages which do not contain any URLs, we use the content only to do the classification. We use the TREC 2007 corpus to train the message model (the training of the web page model has been previously described). Combining results from both classifiers, we obtain the final results of Twitter message classification for which the accuracy achieved is 89.31% (Table 9 and 10).

Table 9: The results of text classification for messages having no URL.

	Predicted legitimate	Predicted spam
True legitimate	1418	609
True spam	921	8682

Table 10: The combination results of messages.

	Predicted legitimate	Predicted spam
True legitimate	11971	1231
True spam	1852	13787

Using the above to obtain the results of messages associated with profiles, we perform threshold-based classification of Twitter profiles (akin to reputation systems). We apply threshold-based classification and vary the threshold between 1 and 10, depending on how many messages have been sent by users. We choose the threshold 1 to obtain the largest spam profiles for our dataset, although this also leads to the largest number of false-positives. The results of which are shown in Figure 6.

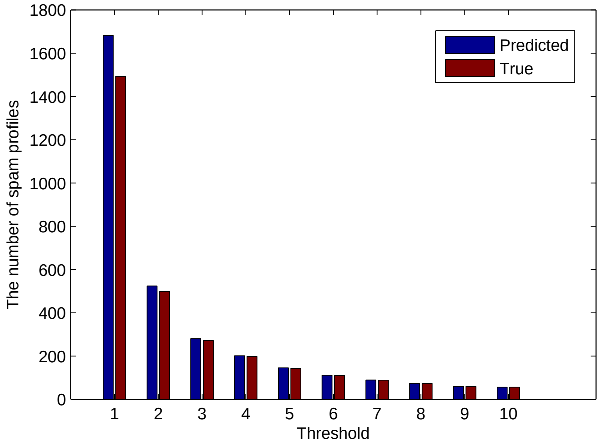


Figure 6: Using messages to identify spam profiles

One of the possible reasons for errors is that some spam profiles may have not been suspended at the time we collected data (thus leading to the labels on the profiles being legitimate instead of spam). Another reason is that for the Twitter dataset, there are some profiles which do not have any messages associated with them. For the profiles which do not have associated messages, we use MySpace profile data as training data to do the classification.

Table 11: The results of profile classification.

(a) Results of profile classification using messages

	Predicted legitimate	Predicted spam
True legitimate	1245	189
True spam	12	1493

(b) Results of text classification for profiles

	Predicted legitimate	Predicted spam
True legitimate	1543	23
True spam	591	904

(c) The combination results of profiles

	Predicted legitimate	Predicted spam
True legitimate	2788	212
True spam	603	2397

Finally, we combined the results from text classification for profiles and message classification. From Table 11, we see the accuracy achieved by the profile classifier is 86.42%.

6. RELATED WORK

Most previous work on social spam has focused on spam prevention on a single social network (e.g., Facebook [11, 24], MySpace [16], Twitter [4]). A number of techniques are employed in these papers including classification, collaborative filtering, behavioral analysis, and in some cases friend-graph analysis.

Although aspects of previous research have been incorporated into our framework to improve results. For example the study by Webb et al. [28] on automatically detect web spam using email spam on detecting Twitter spam using web pages inspired us to classify incoming social spam by taking into account classification of associated content. And we also incorporated Web spam classification methods used by Webb et al. [30] and social profile spam detection (as demonstrated on MySpace) methods used by Irani et al. [16] in our framework.

A large number of classifiers have been used in spam detection but choosing the right classifier and the most efficient combination of them is still a problem. Previous work by Byungki et al. [5] proposes a Bayesian framework, which is theoretical efficient and practically reasonable method of combination, when investigating the integration of text and image classifiers.

Several novel classification approaches are proposed and implemented in cross-domain text classification. Pu Wang et al. [27] presented semantics-based algorithm for cross-domain text classification using Wikipedia based on co-clustering classification algorithm. Elisabeth Lex et al. [20] described a novel and efficient centroid-based algorithm Class-Feature-Centroid Classifier(CFC) for cross-domain classification of weblogs, also they have discussed the trade-off between complexity and accuracy.

Also, some URL spam filtering technique has been proposed by Kurt Thomas et al. [25] to better address different web services such as social networks. They presented a real-time URL spam filtering system named Monarch and demonstrated a modest deployment of this system on cloud infrastructure and its scalability.

7. CONCLUSIONS

We have implemented a spam detection framework to detect spam on multiple social networks. Through the experiments, we show that our framework can be applied to multiple social networks and is resilient to evolution due to the spam arms-race. In the future, we plan on testing and evaluate the framework on live feeds from social-networks. In addition, integrating the detection of spammers' behavior to our framework is considered as future work.

8. REFERENCES

[1] http://hootsuite.com/.

[2] http://tweetdeck.com/.

[3] http://code.google.com/apis/opensocial/.

[4] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *Proceedings of the Seventh annual Collaboration, Electronic messaging, AntiAbuse and Spam Conference(CEAS 2010)*, 2010.

[5] B. Byun, C. Lee, S. Webb, D. Irani, and C. Pu. An anti-spam filter combination framework for text-and-image emails through incremental learning. In *Proceedings of the the Sixth Conference on Email and Anti–Spam (CEAS 2009)*, 2009.

[6] X. Carreras and L. Marquez. Boosting trees for anti-spam email filtering. *Arxiv preprint*, 2001.

[7] J. Caverlee, L. Liu, and S. Webb. Socialtrust: tamper-resilient trust establishment in online communities. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digitial libraries*, 2008.

[8] J. Caverlee and S. Webb. A large-scale study of MySpace: Observations and implications for online social networks. *Proceedings of the International Conference on Weblogs and Social Media*, 8, 2008.

[9] H. Drucker, D. Wu, and V. Vapnik. Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on*, 10(5):1048–1054, 1999.

[10] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The annals of statistics*, 28(2):337–407, 2000.

[11] Gosier and Guadeloupe. Social networks as an attack platform: Facebook case study. In *Proceedings of the Eighth International Conference on Networks*, 2009.

[12] Z. Gyongyi, H. Garcia-Monlina, and J. Pedersen. Combating web spam with trustrank. In *Proceeding of the Thirtieth international conference on Very large data bases*, volume 30, 2004.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[14] J. Hirai, S. Raghavan, H. Garcia-Molina, and A. Paepcke. WebBase: A repository of web pages. *Computer Networks*, 33(1-6):277–293, 2000.

[15] D. Irani, S. Webb, J. Giffin, and C. Pu. Evolutionary study of phishing. *eCrime Researchers Summit, 2008*, pages 1–10, 2008.

[16] D. Irani, S. Webb, and C. Pu. Study of static classification of social spam profiles in myspace. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, May 2010.

[17] D. Irani, S. Webb, C. Pu, and K. Li. Study of trend-stuffing on twitter through text classification. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS 2010)*, 2010.

[18] C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. Voelker, V. Paxson, and S. Savage. On the spam campaign trail. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pages 1–9. USENIX Association, 2008.

[19] M. Learmonth. Twitter getting serious about spam issue. http://adage.com/article/digital/digital-marketing-twitter-spam-issue/142800/, 2010.

[20] E. Lex, C. Seifert, M. Granitzer, and A. Juffinger. Efficient cross-domain classification of weblogs. *International Journal of Intelligent Computing Research*, 1(1), 2010.

[21] C. Pu and S. Webb. Observed trends in spam construction techniques: a case study of spam evolution. In *Proceedings of the Third Conference on Email and Anti-Spam (CEAS 2006)*, 2006.

[22] C. Pu, S. Webb, O. Kolesnikov, W. Lee, and R. Lipton. Towards the integration of diverse spam filtering techniques. In *Proceedings of the IEEE International Conference on Granular Computing (GrC06)*, pages 17–20.

[23] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 workshop*, volume 62, pages 98–05. Madison, Wisconsin: AAAI Technical Report WS-98-05, 1998.

[24] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In *Proceedings of the forth ACM EuroSys Workshop on Social Network Systems(SNS2011)*, 2011.

[25] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time url spam filtering service. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2011.

[26] E. Voorhees, D. Harman, N. I. of Standards, and T. (US). *TREC: Experiment and evaluation in information retrieval*. MIT press USA, 2005.

[27] P. Wang, C. Domeniconi, and J. Hu. Cross-domain text classification using wikipedia. *IEEE Intelligent Informatics Bulletin*, 9(1), 2008.

[28] S. Webb, J. Caverlee, and C. Pu. Introducing the webb spam corpus: Using email spam to identify web spam automatically. In *Proceedings of the Third Conference on Email and Anti-Spam (CEAS 2006)*, 2006.

[29] S. Webb, J. Caverlee, and C. Pu. Characterizing web spam using content and http session analysis. In *Proceedings of the Fourth Conference on Email and Anti-Spam (CEAS 2007)*, pages 84–89, August 2007.

[30] S. Webb, J. Caverlee, and C. Pu. Predicting web spam with http session information. In *Proceedings of the Seventeenth Conference on Information and Knowledge Management (CIKM 2008)*, October 2008.

[31] S. Webb, J. Caverlee, and C. Pu. Social honeypots: Making friends with a spammer near you. In *Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS 2008)*, 2008.