DEIM Forum 2010 B4-3

Topic Classification of Spam Host based on URLs

Young-joo CHUNG Masashi TOYODA and Masaru KITSUREGAWA

Institute of Industrial Science, University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505 Japan

E-mail: {chung, toyoda, kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract In this paper, we determined the main topic of spam hosts based on their uniform resource locator(URL)s. Topic classification of web spam can help personalize spam filters for the web browser, collect topic-specific spam samples with a focused crawler, and understand web spamming activity as a social phenomenon in the cyber space. To classify a URL into different spam topics, we first defined spam topic categories by investigating URLs of spam hosts in our Japanese web archives. Next, we constructed a training set using URLs that were manually categorized into spam topics, and built classifiers using a machine learning approach. In addition, based on the assumption that a small spam link structure consists of pages about a single topic, we used URLs from those structures as additional training data. We categorized URLs of spam hosts from our large scaled Japanese web archive into several topics using two classifiers built by different training sets, and compared the classification results. We improved classification performance from the baseline approach by about 10%.

Keyword Web spam, Topic classification, URL, Machine learning

1. Introduction

The Web has become a major source of information and place for commercial activities for the last two decades. Many people now access the Web via search engines such as Google, Yahoo! And MSN to get knowledge, buy daily product, and join social communities.

However, the half of users look at no more than the top five results in search lists while most people rely on the search engines to find necessary information [1]. Therefore, obtaining a high ranking in the search result list is essential to attract visitors and yield profits. In this situation, some pages started to use unfair ways in order to boost the ranking of their pages. These pages are called *spam pages*, and the behavior that creates spam pages is *web spamming*.

In this paper, we try to examine the topic distribution of spam hosts. The topical study on web spam can contribute to a personalized browser, a topic-specified crawler, and social studies on the cyber space. The web browser can be personalized by displaying a spam link with its topic tag, which enables users to see the contents of the hyperlink without clicking it. The focused crawler can collect spam samples on a particular topic, which is useful for computing a topic specific search ranking of a page, and for updating spam filters. In addition, topic distribution in web spam pages will reflect the characteristic of spammers' behavior which is an interesting object for sociologists.

We categorize spam hosts into different topics based on lexical features of their uniform resource locator(URL)s. We build a topic classifier by a machine learning approach that trains the classifier using URLs of spam hosts that are already classified into the topic. However, it is difficult to obtain sufficient labeled spam URLs for training because the cost of labeling a hostname is expensive. In particular, the hostname of web spams usually contains words from various languages, a name of a specific product or a person, and misspelled words. These characteristics of spam URLs make the labeling harder.

To obtain sufficient labeled spam URLs for classifier training, we use spam hosts in a *spam farm*. A spam farm is a densely connected link structure created by spammers with the purpose of boosting the ranking of spam pages [3]. It can be assumed that a relatively small spam farm consists of pages about similar topics. Based on this assumption, we use unlabeled hostnames in categorized spam farms for training samples. We determine the topic of a small spam farm with keywords that appear most frequently in it. Hostnames will have the same label as that of the spam farm where they belong.

Spam farms can be obtained by applying the strongly connected component (SCC) decomposition to the web graph. Strongly connected component of a graph is a directed subgraph where every pair of nodes has a direct path between them. Since the spam farm is a densely connected link structure, it can be supposed that the spam farm is a SCC. In our previous work [4] [5], we confirmed that SCCs that consist of over 100 hosts are very likely to be spam farms.

We define spam topics and implement a binary classifier for each topic. Seven spam topics are defined based on the study on spam topics in e-mail [10] and our investigation into the topics of spam hostnames in large SCCs. We implement seven binary classifiers to determine the topic of the hostname. Given a hostname, each binary classifier checks whether a hostname is related with one topic or not.

In order to confirm whether hostnames labeled by spam farms can improve the classification performance, we build classifiers using two types of training sets: the set contains hand-labeled URLs only and the set of hand-labeled and SCC-labeled URLs. We compare the classification results and find that labeling hostname by SCCs can help classify spam hostnames.

The rest of this paper is organized as follows. In Section 2, we introduce the related work of our study. Section 3 provides the explanation for the learning algorithm, features and SCCs in detail. In Section 4, the experimental results are described. Finally we summarize and conclude our work in Section 5.

2. Related Work

Several studies on the classification web pages by their URLs have been conducted. Kan and Thi suggested the approach to web page classification using URLs [6]. They showed the classification with URLs is useful when page contents are not available. Baykan et al. tried to determine the type of language in which a page is written only based on URLs [7]. They used various lexical features from URLs and classified pages into 5 different languages with high accuracy. In [8], Baykan et al. categorized pages into 15 topics based on their URLs. Topics and pages are obtained from Open Directory Project[1]. This work is similar to ours in that URLs are used for topic classification, but we focus on spam hosts. Ma et al. identified spam sites by lexical and host-based features of their URLs [9]. They employed online-learning algorithms to handle the large scale data set from a web mail provider. The result showed their approach can classify malicious URLs with high accuracy. This is different from our work in that they classify spam URLs from non-spam ones, and did not classify spam URLS into topics.

On the other hand, some research has been done on the spam topics. Hulten et al. categorized spam e-mail messages by the type of a product that spammers try to advertise [10]. They manually examined 1,200 spam messages from 2003 and 2004 and divided them into 10 categorizes. This study is similar to ours in that they try to categorize spam. However, we classify web spam pages based on their hostnames and using automatic classifiers.

3. Approach

In this section, we briefly review a machine learning algorithm that is used in this paper. After that, we show features for the topic classifier. We also explain the strongly connected component that will be a source of labeled samples for classification.

3.1. Learning Algorithm

We used an online learning algorithm because it guarantees faster convergence than other learning algorithms and can handle large scale data such as the web graph [9].

In online learning, a classifier try to assign a correct label on each sample that comes into in sequential manner. We can denote a pair of sample and its label in round i by $(\mathbf{x}_i,\ \mathbf{y}_i)$ where $\mathbf{x}_i$ is a feature vector of a sample and $y_i \in \{+1,-1\}$ is its label. At each round, the algorithm predict a label of a sample based on its weight vector $\mathbf{w}_i$ and produces $y_i\ (\mathbf{w}_i \cdot \mathbf{x}_i)$ as a margin. Such a margin is can be regarded as the distance between the sample and the hyper-plane that divide classes. If the margin is positive, the prediction was correct. Otherwise, algorithm modify weight vector $\mathbf{w}_i$ to produce more accurate prediction on next coming sample $\mathbf{x}_{i+1}$.

Among several online algorithms, we use the confidence-weighted (CW) learning algorithm proposed by [11] [12]. The CW learning algorithm maintains a Gaussian distribution over weight vectors with a covariance matrix that implies confidence about feature weights and a correlation. With this information, the CW algorithm updates the feature weigh vector with less confidence more aggressively. The Gaussian distribution for confidence has the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$. $\boldsymbol{\mu}_p$ and $\boldsymbol{\Sigma}_{p,p}$ represents knowledge of and confidence in the weight for feature p. Hence, the smaller $\boldsymbol{\Sigma}_{p,p}$ means we have the more confidence in $\boldsymbol{\mu}_p$, the mean weight value of feature p.

In the CW algorithm, the Kullback-Leibler(KL) divergence between previous and updated distributions is used to measure the magnitude of the update. he algorithm updates the model to minimize KL divergence while satisfy the condition that $\mathbf{x}_t$ can be correctly classified with probability over η. Therefore, the optimization constraint will be as following:

$$(\boldsymbol{\mu}_{\mathrm{i}+1}, \boldsymbol{\Sigma}_{i+1}) = \arg\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \mathrm{D}_{\mathrm{KL}}\left(N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \| \, N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right)$$

$$\mathrm{s.t.}\,\Pr[y_i(\mathbf{w}\ \cdot \mathbf{x}_i) \geq 0] \geq \eta.$$

[1] http://www.dmoz.org/

When CW-Stdev algorithm [12] receives a sample x_i labeled as y_i, the mean $\boldsymbol{\mu}_{\mathrm{i}}$ and the covariance matrix $\boldsymbol{\Sigma}_i$ will be updated as follows:

$$\boldsymbol{\mu}_{\mathrm{i}+1} = \boldsymbol{\mu}_{\mathrm{i}} + \alpha_i y_{\boldsymbol{i}} \boldsymbol{\Sigma}_i \mathbf{x}_i \ ,$$
$$\boldsymbol{\Sigma}_{i+1} = \boldsymbol{\Sigma}_i - \beta_i \boldsymbol{\Sigma}_i \mathbf{x}_i \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\Sigma}_i \ .$$

$\alpha,\ \beta$ are given by:

$$\alpha_i = \max\left\{0, \frac{1}{\mathbf{v}_i\zeta}\left(-\mathrm{y}_i(\boldsymbol{\mu}_{\mathrm{i}} \cdot \mathbf{x}_i)\psi + \sqrt{\mathrm{y}_{\mathrm{i}}(\boldsymbol{\mu}_i \cdot \mathbf{x}_i)^2 \frac{\phi^4}{4} + \mathbf{v}_i \phi_i^2 \zeta}\right)\right\} \ ,$$

$$\beta_i = \frac{\alpha_i \phi}{\sqrt{\mathbf{u}_i} + \mathbf{v}_i \alpha_i \phi} \ ,$$

where $\phi, \zeta,\ \psi, \mathbf{v}_i, \mathbf{u}_i$ are:

$$\phi = \boldsymbol{\Phi}^{-1}(\eta),$$
$$\zeta = 1 + \phi^2,$$
$$\psi = 1 + \phi^2/2,$$
$$\mathbf{v}_i = \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\Sigma}_i \mathbf{x}_i,$$
$$\mathbf{u}_i = \frac{1}{4}\left(-\alpha \mathbf{v}_i \phi + \sqrt{\alpha^2 \mathbf{v}_i^2 \phi^2 + 4\mathbf{v}_i}\right)^2.$$

The detailed explanation for each parameter is shown in [12].

3.2. Features

We use two types of lexical features of hostnames: bag-of-words and n-grams.

Bag of words Each URL is lower-cased and split into tokens by using punctuation marks, numbers or other non-alphabetic characters as delimiters. Among obtained tokens, we removed tokens of which the length is less than 2, and tokens that start with two same characters. For instance, hostname **"www.free-download-ringtones.com"** will produce tokens **"free", "download", "ringtones",** and **"com"**.

***n*-gram** From tokens created by the above method, *n*-grams are extracted. *N*-gram is the sequences of n-characters. If a token contains characters fewer than *n*, that token is not changed. Therefore, if we use 5-gram, we can divide **"cheaphotel"** into six 5-grams including **"cheap", "heaph", "aphot", "phote"** and "**hotel"**. In this paper, we use 3, 4, 5, 6, 7 and 8 grams.

3.3. Strongly Connected Components

To obtain sufficient spam hostnames related with a single topic, we consider strongly connected component (SCC). SCC of a graph is the directed subgraph where every pair of nodes has a direct path between them. Since spam hosts construct a densely connected link structure [3], and links between spam and normal site seldom exist, it can be expected that spam hosts form a SCC. Our previous work [4] [5] confirmed that SCCs of the size over 100 are very likely to be a spam structure.

We consider a small SCC as a topic unit based on the assumption that spammers create a link structure to promote related products so that nodes in the same SCC are related to each other. Therefore, if we determine the category of one SCC, we can assume that hostnames in that SCC belong to the same category as the SCC.

SCCs are obtained by the recursive SCC decomposition algorithm with node filtering [5]. This algorithm decomposes the whole web graph into SCCs. After that, it prunes nodes with small degrees from the largest SCC, so called the core, and decomposed the pruned core into SCCs recursively with increasing a threshold of node degree. That is, after we decompose the whole host graph into SCCs, we remove hosts in the core whose in-degree and out-degree are smaller than 2, and decompose the remaining hosts into SCCs again. As a result, we can extract denser SCCs in the core. Next, we consider the largest SCC among newly obtained SCCs, and filter out hosts from it of which in and out degrees are smaller than 3, and apply the decomposition algorithm to the remaining hosts. This process is performed recursively with incrementing the degree threshold, and continued while we have large SCCs in the results.

To determine the topic of a small SCC, we count the frequency of tokens (See Section 3.2) from URLs in each SCC and manually check tokens in the top of the frequency list. For example, if tokens like **"casino**", **"poker"** and **"game"** appear in the top of the token frequency list of a SCC, that SCC can be categorized into the "gamble" topic. All hostnames in that SCC then will be assumed to be related with gambling. This approach will help to obtain sufficient training samples for classifiers.

4. Experiments

4.1. Data Set

Large scale snapshots of Japanese Web archive are used for experiments. These snapshots are built by crawling that conducted from 2004 to 2006. Our crawler is based on the breadth first crawling [13], but it focuses on pages written in Japanese. If a page is written in Japanese, our crawler collected that page even if it is located outside the .jp domain. The crawler stopped collecting pages from a site if it could find no Japanese pages on the site within the first few pages. Therefore, our snapshot contains pages written in various languages including English. Our crawler does not have an explicit spam filter while it detects mirror servers and tries to crawl only representative servers. As a result, our archive includes spam pages without mirroring.

In this paper, we will use a host graph, where each node is a host and each edge between nodes is a hyperlink between pages in different hosts. Host graphs for 2004, 2005 and 2006 were built. In each graph, we included only hosts that existed in the 2006 archive, and did not consider hosts disappeared from 2004 to 2005. This is because we cannot distinguish whether those hosts really disappeared or they were just not reached by our crawler. The properties of our Web snapshot are shown in Table 1.

Table 1 The Properties of the host graph

Year	2004	2005	2006
# of nodes(hosts)	2.98M	3.70M	4.02M
Number of edges	67.96M	83.07M	82.08M

Spam hosts are obtained by applying recursive SCC decomposition algorithm with node filtering to host graphs (See Section 3.3). From SCCs that are obtained during five iterations, we select ones that contains over 100 hosts. The number of such SCCs and hosts in them are described in Table 2.

Table 2 The number of SCCs of size over 100 and host in them. SCCs are obtained during 5 iterations.

	#of SCCs(100<)	# of Hosts in SCCs
2004	268	215,515
2005	235	160,049
2006	239	193,817
Total	742	569,381

From 569,318 hosts, we remove duplicate hostnames and finally obtain 245,822 hosts. Based on our previous work [4] [5], we regard these hosts as spam.

4.2. Topic Categorization of URLs of Spam Host

To determine the type of web spam topic, we refer to the topic categorization of e-mail Spam [8]. However, since techniques of web spamming are different from those of e-mail spamming, we remove and add some categories after the investigation on spam hosts in our data set.

- **Adult contents** This type of URLs contains porno-related words and/or the names of movie stars and singers from various countries.
- **Dubious product** This type of URLs is related with illegal products such as a crack, a key generator and pirate DVDs. The crack is used to remove protection methods like copy protection and serial keys of digital products, and a key generator generates illegal serial key for such product.
- **Financial produc**t This type of URLs contains the word like banking, credit card, loan, mortgage and real estate.
- **Gamble** This type of URLs includes words like gamble, casino, and many different type of poker game.
- **Mobile phone** This type of URLs are related with mobile contents such as wall-paper, ringtone, text-message formats and mobile games.
- **Jobs** This type of URLs includes words about employment, job, and affiliation.
- **Travel** This type of URLs consists of words about hotels, accommodations, flight tickets, and car rental.

We find a number of hostnames that advertise a specific product or provide regional information as weather or news, but we do not define topic categories for them. A category for mobile phone contents is defined because there are a number of hostnames in various languages related with downloading contents like wallpapers, ringtones and games for the mobile phone.

4.3. Topics of Small Strongly Connected Components

To expand the size of the training set consists of only hand-labeled hostnames, we add hostnames from small SCCs to the training samples. For this, we extract small SCCs of size under 180. The total number of obtained SCCs was 299.

Table 3 The number of SCCs and hosts about each topic.

	# of SCC	# of Hosts
Adult	78	6,082
Dubious	3	330
Financial	10	658
Gamble	14	938
Jobs	18	1,048
Mobile	11	642
Travel	31	2,250
Total	165	11,948

We check the token frequency list of each SCC in order to categorize SCCs into topics described in Section 4.2,. Table 3 shows the detail. Note that we discard the SCC that contains meaningless hostnames like **“qhht.sz.focus.cn”**, **“10.sai.jp”**. We also exclude the SCC if keywords related with different categories are shown in the top of its token frequency list.

4.4. Topic Classification

4.4.1. Experimental setup

We build multiple binary classifiers for each topic, rather than single multi-way classifiers. Since we have seven spam topics, total seven binary classifiers are prepared for our experiment. For each classifier, hostnames related with a specific topic are labeled as positive, while the rest of them is labeled as negative. One classifier determines whether a sample belongs to a topic or not. For the implement of CW algorithm, we use the online learning library, oll [14].

Hand-labeled 200 samples and SCC-labeled 400 samples are prepared for each topic. We randomly choose 200 hostnames from the whole spam hostnames and categorize them by hand. Since our end is classifying spam URLs, we do not investigate host contents intensively during classification. That is, if we have to classify a hostname like **"planwagenfahrt.de"** that contains no spam keywords**,** we discard this hostname and select another one to classify. Only after we have categorized hostnames, we check their contents to confirm that hostnames are labeled properly. 400 samples are selected from the small SCCs that are described in Section 4.3[2]. In total, we have 1400 hand-labeled and 2800 SCC-labeled samples.

Seven test sets are then created from the whole samples. For each topic, we randomly selected 50 hostnames from a hand-labeled set and 50 from a SCC-labeled set and check their contents. As a result, a test set for each classifier contains 700 samples that consist of 100 positive and 600 negative samples.

To verify our assumption that a small SCC contains hostnames associated with a single topic, we create two different training sets. For each topic, the first training set includes only 1,050 hand-labeled hostnames that consist of 150 positive and 900 negative samples. The second training set includes 3,500 hostnames that contain 500 positive samples (150 hand-labeled and 350 SCC-labeled hostnames) and 3,000 negative samples. Note that we do not balance the number of positive and negative samples in both training sets. In each training set, the number of negative samples is six times as many as that of positive samples.

We divide every training set into 2 subsets, then train classifiers with them and evaluate their performance. In other words, we train every classifier with two different training sets and test it with the single test set. The final performance will be the average of two different results.

.

4.4.2. Evaluation Metric

To evaluate the performance of our classifiers, precision, recall and F-measure are used. They are given by:

$$\mathrm{Precision} = \frac{|\text{positive samples that are classified as positive}|}{|\text{samples classified as positive}|},$$

$$\mathrm{Recall} = \frac{|\text{positive samples that are classified as positive}|}{|\text{positive samples}|},$$

$$\mathrm{F-measure} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}.$$

4.4.3. Experimental result

Experimental results that are obtained by using different training sets and features are described in Table 3 and 4.

Hand-labeled only vs. Hand-labeled and SCC labeled Both Table 4 and Table 5 show that classifiers trained with hostnames labeled by both hand and SCCs performed better than those trained with only manually labeled hostnames. The bag-of-words based classifier for adult category outperformed by about 21% when we add hostnames from SCCs. The average of the improvement in F-measure was 0.13 when the features were bag-of-words, and 0.06 when the features were n-gram. This implies that a small SCC consists of hosts on the same topic.

Bag of word features vs. n-gram features By comparing the result of Table 4 and Table 5, we can notice that *n*-gram features performed better than bag of word features. We classified samples related with the dubious topic perfectly using n-gram features, and hand and SCC labeled training samples.

5. Discussion

We built classifiers using different features and labeling strategy and compared their results. The better performance by adding training samples from SCCs can be explained with the fact that hostnames in a SCC can be a context for other hostnames. For example, it is hard for non-German speakers to categorize the hostname

[2] Since the number of hostnames in the dubious product category was insufficient for our experiment, we added 70 hostnames that are randomly selected and categorized by hand. Thus, we have 170 hand-labeled and 330 SCC-labeled samples for the dubious product category.

“planwagenfahrt.de” of which contents is related with a tour in a covered wagon. However, if this hostname appears in a SCC containing hostnames like **“www.hotel-hunsrueck.de”**, **“last-minute.hotelliste.de”**, **“www.map-of-germany.com”**, we can assume that URL is related with the travel.

N-gram features were better than bag-of-words features to classify spam hosts. This might be because of that spammers deliberately use misspelled, broken or connected tokens in their URLs to avoid spam filters. Tokens like **“cheaaphotels”**, **“m0rtgage”**, **“reduceyourtaxes”** cannot be a useful feature for the spam topic classification unless we use n-gram features.

Table 4 The classification performance based on different training sets. Bag-of-words is used as features. P, R, and F represent precision, recall, and F-measure, respectively.

	Hand only			Hand + SCCs		
	P	R	F	P	R	F
Adult	0.799	0.725	0.760	0.948	0.995	0.971
Dubious	0.816	0.995	0.896	0.966	0.995	0.980
Financial	0.770	0.780	0.775	0.976	0.990	0.983
Gamble	0.835	0.955	0.891	0.970	0.980	0.975
Jobs	0.824	0.910	0.865	0.952	0.980	0.966
Mobile	0.870	0.920	0.894	0.976	0.980	0.978
Travel	0.828	0.840	0.834	0.975	0.980	0.977

Table 5 The classification performance of different training set. 3-8 grams are used as features.

	Hand only			Hand + SCCs		
	P	R	F	P	R	F
Adult	0.947	0.800	0.867	0.976	0.985	0.980
Dubious	1.000	0.995	0.997	1.000	1.000	1.000
Financial	0.978	0.815	0.889	1.000	0.995	0.997
Gamble	0.990	0.995	0.992	0.995	0.985	0.990
Jobs	0.988	0.835	0.905	1.000	0.985	0.992
Mobile	0.941	0.880	0.909	1.000	0.995	0.997
Travel	0.978	0.895	0.935	0.995	0.970	0.982

6. Conclusion

In this paper, we categorized spam hostnames into topics. To this end, we defined spam topics by the investigation on spam URLs in large scale Web archive. We used a spam link structure, or a SCC, to obtain sufficient training samples. SCCs are extracted by the recursive SCC decomposition algorithm with node filtering. We applied the SCC decomposition algorithm recursively to the largest SCC where nodes with small degrees are filtered out. We trained a binary classifier for each topic and evaluated the performance. Classification results based on lexical features of URLs showed high accuracy for all topics. Moreover, we have shown that using hostnames from SCC as prior information for training can improve the performance by average about 10%.

References

[1] S. Nakamura, S. Konishi, A.Jatowt, H. Ohshima, H. Kondo, T. Tezuka, S. Oyama and K. Tanaka. “Trustworthiness Analysis of Web Search Results,” Proc. the 11th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2007), 2007.

[2] Z. Gyöngyi and H. Garcia-Molina. “Web spam taxonomy”, Proc. the 1st international workshop on Adversarial information retrieval on the Web(AIRWEB '05), 2005.

[3] Z. Gyöngyi and H. Molina. “Link Spam Alliance”, Proc. the 31st international conference on very large data bases(VLDB '05), 2005.

[4] H. Saito, M. Toyoda, M. Kitsuregawa and K. Aihara. “A large-scale study of link spam detection by graph algorithms”, Proc. the 3rd international workshop on adversarial information retrieval on the Web (AIRWEB '07), 2007.

[5] Y. Chung, M. Toyoda, and M. Kitsuregawa, “A study of link farm distribution and evolution using a time series of web snapshots, ” Proc. the 5th International workshop on adversarial information retrieval on the Web(AIRWEB '09), pp. 9-16, 2009.

[6] M. Kan and H.O.N. Thi, “Fast webpage classification using URL features, ” Proc. The 14th ACM International Conference on Information and Knowledge Management (CIKM '05), New York, NY, pp. 325–326, 2005.

[7] E. Baykan, M. Henzinger, and I. Weber, “Web page language identification based on URLs, ” Proc. the VLDB Endowment, pp. 176-187, 2008.

[8] E. Baykan, M. Henzinger, L. Marian, and I. Weber, “Purely URL-based topic classification, ” Proc. the 18th international conference on World wide web(WWW '09), pp.1109-1110, 2009.

[9] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, “Identifying suspicious URLs: an application of large-scale online learning, ” Proc. the 26th annual international conference on machine learning, pp.681-688. 2009.

[10] G. Hulten, A. Penta, G. Seshadrinathan, and M. Mishra, “Trends in Spam Products and Methods, ” Proc. the First Conference on Email and Anti-Spam (CEAS '04), 2004.

[11] M. Dredze, K.Crammer, and F. Pereira. “Confidence-weighted linear classification. ” Proc. the 25th international Conference on Machine Learning (ICML '08) 2008.

[12] K. Crammer, M. Dredze, and F. Pereira, F. “Exact Convex Confidence-Weighted Learning. ” *Advances in Neural Information Processing Systems 21* (pp. 345--352), 2009.

[13] M. Najork and J. L. Wiener. “Breadth-first crawling yields high-quality pages”, Proc. the 10th international conference on World Wide Web (WWW '01), 2001.

[14] D.Okanohara and K. Ohta. Online Learning Library. http://code.google.com/p/oll/