Looking into the Past to Better Classify Web Spam

Na Dai Brian D. Davison Xiaoguang Qi
Department of Computer Science & Engineering
Lehigh University
Bethlehem, PA 18015 USA
{nad207,davison,xiq204}@cse.lehigh.edu

ABSTRACT

Web spamming techniques aim to achieve undeserved rankings in search results. Research has been widely conducted on identifying such spam and neutralizing its influence. However, existing spam detection work only considers current information. We argue that historical web page information may also be important in spam classification. In this paper, we use content features from historical versions of web pages to improve spam classification. We use supervised learning techniques to combine classifiers based on current page content with classifiers based on temporal features. Experiments on the WEBSPAM-UK2007 dataset show that our approach improves spam classification F-measure performance by 30% compared to a baseline classifier which only considers current page content.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web based services*; I.7.5 [**Document and Text Processing**]: Document Capture—*Document analysis*

General Terms

Algorithms, Experimentation, Measurements

Keywords

search engine spam, spam classification, temporal features, archival web

1. INTRODUCTION

Web spamming techniques aim to achieve undeserved rankings in search results. Frequently used techniques include keyword stuffing, cloaking, link farms, etc. The presence of spam sites in top results of queries not only degrades the retrieval quality of engines, but also harms other web sites that should be highly ranked. Therefore, web sites utilizing such techniques should be detected.

AIRWeb '09, April 21, 2009 Madrid, Spain.

However, it is not an easy task since the spamming techniques have also improved along with the anti-spam techniques. And there is no method which can entirely detect all kinds of spam pages. Existing spam detection approaches [5, 8, 10, 11, 15] utilize link and/or content information to help detect spam pages. However, most of them only consider current information (i.e., the most recent page content and link graph) without looking at previous versions of the page.

Both spam pages and normal pages change over time. Although each page has its own way to evolve, the evolution patterns of spam pages and non-spam pages may be differentiated in general, since the goals of these two kind of pages are likely to be different.

Therefore, we argue that historical information may also be useful for spam detection. For example, when an authoritative domain name is sold to a new owner (or registered by another party as soon as it expires), the incoming hyperlinks may still remain for some time. Therefore, even if the quality of the web site has decreased and its topic changed, it could still be ranked high due to the existing incoming authority flow. Although an ownership transition does not necessarily make a site spam, this effect is often harnessed by spammers in order to promote their sites. Although the current content of a site is often insufficient for a decision about whether this site is spam, by analyzing historical copies of the domain in question, we may identify such a transition of ownership, which indicates the necessity of reassessing the quality of this domain.

In this paper, we propose a new approach to spam classification that is able to utilize historical information. We extract a variety of features from archival copies of the web provided by the Internet Archive's Wayback Machine [12], and use them to train classifiers in conjunction with features extracted from current snapshots. We achieve a prominent improvement compared to approaches that only consider current information. In particular, experiments on the WEBSPAM-UK2007 dataset show that our approach improves spam classification F-measure performance by 30% compared with a baseline classifier which only considers current page content.

The contributions of this work include:

- we show that historical information is useful in addition to current information in spam classification;
- we determine the sensitivity of spam classification performance with respect to the time-span of extracted historical features.

The rest of this paper is organized as follows. In Section 2, we review related work in spam detection. We motivate our method in Section 3, and describe it in detail in Section 4. We report and analyze experimental results in Section 5 and discuss some issues in Section 6. The paper concludes with a summary in Section 7.

2. RELATED WORK

Web spam detection has been the focus of many research efforts (e.g., [8, 11, 15, 18, 21, 22]). Most of the work utilizes the characteristics of link structures or content to differentiate spam from normal pages or sites. Gyongyi et al. [10] propose a concept called spam mass and the method to estimate it, and successfully utilize it for link spamming detection. Wu et al. [23] use trust and distrust propagation through web links to help demote spam pages in search results. Andersen et al. [2] detect link spamming by examining the combination of parent pages which contribute most to the PageRank of a target node. Becchetti et al. [4] study the characteristics of multiple link-based metrics with respects to spam detection over a large collection of Web pages. Wu and Davison [22] detect semantic cloaking by comparing the consistency of two copies of a page retrieved from a browser's perspective and a crawler's perspective.

Research on spam detection by content usually utilizes statistics based on content characteristics. Ntoulas et al. [15] detect spam pages by building up a classification model which combines multiple heuristics based on page content analysis. Urvoy et al. [19] extract and compare the HTML similarity measures among pages to cluster a collection of documents and enhance the Web spam classification. Attenberg and Suel [3] clean the spam in search results by capturing pages' topical structure based on a term-based summary data structure. Biro et al. [6] use a modified Latent Dirichlet allocation method to train a collection of spam and non-spam topics respectively, and use them to help classify spam sites.

However, relatively little work about spam detection utilizes historical information. Google [9] filed a patent [1] on using historical information for scoring and spam detection. Lin et al. [14] show blog temporal characteristics with respect to whether the target blog is normal or not. They extract temporal structural properties from self-similarity matrices across different attributes, and successfully use them in the 2006 TREC Blog Track. Shen et al. [18] collect a web graph with two snapshots and use temporal features extracted from the difference between these two snapshots to demote link spam under the assumption that link spam tends to result in drastic changes of links in a short time period. However, they only extract the temporal features from the variation of web link structures. In this work, we extract temporal content-based features from multiple snapshots and train a spam classifier by using the proposed features within a real world data set.

3. MOTIVATION

Temporal content contains plenty of information, from which we can extract the trends of site's variation within a specified timespan. Such variation reflects how the site's quality and topic change over time, which may help classify spam. Consider such a scenario. When a classifier only uses the current snapshot to detect Web spam, it may judge the target page to be spamming when the fraction of popular words is found to be high (e.g., as in [15]). However, it will cause a false positive in some cases since some pages or sites do show popular content, but are not attempting to deceive search engines to get higher ranking positions. By combining the trend of a site's fraction of popular words within a previous time interval, we can introduce a new kind of feature for web spam detection. For example, if a page's fraction of popular words has a sudden increase within a short previous time interval, then we have higher confidence to classify it as a spam page, and vice versa. Hence, features extracted only from the current snapshot cannot detect the changes and provide such confidence estimation.

A change in the organization that is responsible for the target site may also be a signal for spam classification. Many spammers buy expired domains from previously normal sites in order to exploit the authority of the prior site before search engines realize such activities. We conjecture that there is a correlation between whether the organization changes within a short previous time interval and whether the target site should be labeled as spam. Our intuition is that non-spam sites are more likely to be under the administration of an unchanged organization whereas the ones which become spam pages are more likely to suffer from the change of owner. Suppose *A* is the event that a target site changes its organization information within the short previous time interval, and *B* is the event that it is a spam site now. We conjecture:

$$\mathrm{P}(A|B) > \mathrm{P}(A|\neg B).$$

Since we cast Web spam detection as a classification problem, we care about P(B|A) in particular. Bayes' theorem defines probability P(B|A) as:

$$\begin{aligned} \mathrm{P}(B|A) &= \frac{\mathrm{P}(A|B) \times \mathrm{P}(B)}{\mathrm{P}(A|B) \times \mathrm{P}(B) + \mathrm{P}(A|\neg B) \times \mathrm{P}(\neg B)} \\ &\propto \mathrm{P}(A|B) \times \mathrm{P}(B) \end{aligned}$$

For each example, we only care about whether $P(B|A) > P(\neg B|A)$ or not. Since the denominators in calculating $P(B|A)$ and $P(\neg B|A)$ are the same, we only care about the numerator part.

Thus, we can extract *temporal features* from historical snapshots of web pages to help classify Web spam since they are potentially complementary resources for improving spam classification.

4. USING TEMPORAL FEATURES TO DETECT WEB SPAM

Time is a continuous variable whereas the change of Web page content can be viewed as a stochastic process. Instead of analyzing the temporal series variation directly, we observe the change of pages by sampling at discrete time points; that is, we take snapshots uniformly within our specified time interval. By comparing the pair-wise differences between consecutive snapshots, we can ascertain the trends of page changes over time. Groups of features are extracted to reflect such changes, and used to train separate classifiers. We then combine the outputs of these classifiers to boost Web spam classification.

4.1 Temporal Features

Here, we focus on classification of web spam at the site level based on content. Our temporal features fall into two categories. The first category contains features aiming to capture the term-level importance variation over time for each target site. Features in the second category mainly focus on identifying whether the target site has changed its organization information during the time of observation.

We collect all the historical snapshots for each site homepage from the Wayback Machine. We uniformly select N snapshots ($H_1, \ldots, H_N$) for each site, which have an HTTP response code of 200 (denoting successful retrieval). Our hope is that the selected snapshots can well represent how the site changes. For each selected snapshot for the target site homepage, we follow its outgoing links and download them individually via the Wayback Machine. We use all of the downloaded snapshots' content within the same site for each selected time point to represent the content of the whole site at that time. We parse each site's content and decompose it into a term list by removing all HTML tags within it and replacing all non-alphanumeric characters with blanks. We treat such a term list at each time point as a *virtual document* (i.e., $V_1, \ldots, V_N$). Our corpus includes all the virtual documents at 10 time points for

Notation	Meaning
N	The number of virtual documents for one site or the number of time points we take
V_j	The jth virtual document
t_{ij}	The weight on the ith term of the jth virtual document.
$\overrightarrow{T_j}$	The term weight vector of the jth virtual document
s_{ij}	The weight difference on the ith term between the (j+1)th and the jth virtual documents, i.e., $t_{i(j+1)} - t_{ij}$
$\overrightarrow{S_j}$	The difference between $\overrightarrow{T_{j+1}}$ and $\overrightarrow{T_j}$, i.e., $(\overrightarrow{T_{j+1}} - \overrightarrow{T_j})$

Table 1: Notation used in temporal feature expression.

all the sites. We remove all stop words[1] and then order terms by the frequency of occurrence in this corpus. The 50,000 most frequent terms comprise our vocabulary.

The term weights in each virtual document are calculated by BM25 score [17], which are used to represent the importance for relevance for each term in each given virtual document. It is defined by:

$$\sum_{t \in Q} w \frac{(k_1+1)tf(k_3+1)qtf}{(K+tf)(k_3+qtf)} \quad (1)$$

Since we calculate BM25 score for each term in a specified virtual document, the query is every term in our vocabulary at one time. Equation (1) is converted into (for each term):

$$w \frac{(k_1+1)tf}{K+tf}$$

where K is given by

$$k_1((1-b)+b \times dl/avdl)$$

and dl and $avdl$ are document length and average document length respectively. w is the Robertson/Sparck Jones weight for each term, which is defined by:

$$\log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)}$$

where N is the total number of virtual documents, n is the number of virtual documents which contain the specified term, R is the number of virtual documents relevant to a specific topic, and r is the number of relevant virtual documents which contain this term. We set R and r to be 0 in calculating BM25 score related temporal features in this work.

The term weight vector, which is composed of the weights (i.e., BM25 scores) on all the terms defined in the vocabulary, represents the fine-grained topic distribution in every virtual document. By comparing the term weight vectors among the virtual documents at different time points, we can discover how the topics of a site change over time. The notation used in our feature definition is listed in Table 1. Our feature groups are described as follows:

- **Ave(T)**— For each site, Ave(T) calculates the average among term weight vectors for the virtual documents at all the time points. This group of features can enhance current content by providing a simple average historical content. Ave(T) is defined as:

$$Ave(T)_i = \frac{1}{N} \times \sum_{j=1}^{N} t_{ij}$$

- **Ave(S)**— For each site, Ave(S) calculates the mean difference between temporally successive virtual term weight vectors. This group of features captures the average variation of the importance of each term.

$$Ave(S)_i = \frac{1}{N-1} \times \sum_{j=1}^{N-1} s_{ij}$$

- **Dev(T)**— For each site, Dev(T) calculates the deviation among virtual term weight vectors at all time points.This group of features reflects the variation scope for the importance of each term in the virtual documents. Dev(T) is defined as:

$$Dev(T)_i = \frac{1}{N-1} \times \sum_{j=1}^{N} (t_{ij} - Ave(T)_i)^2$$

- **Dev(S)**— For each site, Dev(S) calculates the deviation of the term weight vector differences of successive virtual documents. Dev(S) is given by:

$$Dev(S)_i = \frac{1}{N-2} \times \sum_{j=1}^{N-1} (s_{ij} - Ave(S)_i)^2$$

- **Decay(T)**— For each site, Decay(T) accumulates the term weight vectors of each virtual document. The term importance influence reduces exponentially with the time interval between the time point of each virtual document and that of the latest one. The base of the exponential function (λ) is the *decay rate*. Decay(T) is defined as:

$$Decay(T)_i = \sum_{j=1}^{N} \lambda e^{\lambda(N-j)}(t_{ij})$$

4.2 Classification of organization change

Finally, in order to know whether a target site has changed its organization information within the past four years, we train a classifier, which automatically classifies the ones with changed organizations.

We randomly selected 20,078 external pages from the intersection of 2004 and 2005 ODP content, and track whether they are still in ODP directory [16] by 2007. Among these 20,078 examples, 53% are removed from ODP directory, whereas 47% are still retained by 2007. We randomly select examples from removed external pages, and manually label them based on the criteria whether they changed organization from 2004 to 2007. Sampling without replacement is used to generate the examples to be labeled. We repeat this procedure until we get 100 examples which have changed organizations. They comprise the positive examples in our ODP data set. We then randomly select 147 examples from the retained external pages as our negative examples. Since the pages of sites having changed organization usually have topic change, they can be identified by checking the consistency between ODP descriptions and page content in most cases. Thus, the selected 247 examples comprise our data set for training and validating our classifier of checking whether the sites have changed their organization information.

We download four years (from 2004 to 2007) historical snapshots for each selected external page from the Wayback Machine. We select nine successfully retrieved snapshots for each selected external page uniformly in order to represent how the page content changes within these four years. We extract several groups of

[1]http://www.lextek.com/manuals/onix/stopwords1.html

Content-based feature groups
• features based on title information
• features based on meta information
• features based on content
• features based on time measures
• features based on organization which is responsible for the external page
• features based on global bi-gram and tri-gram lists
Category-based feature groups
• features based on category
Link-based feature groups
• features based on outgoing links and anchor text
• features based on links in framesets

Table 2: The feature groups (Organization historical features) used in classifying whether site ownership has changed within the past four years.

historical features from the snapshots of each external page. Most of these features reflect the contrast between two snapshots. Table 2 shows the main feature groups (i.e., organization historical features) which are used. We use SVM^{light} [13] to train and validate our classifier, and then directly classify the site homepages selected in our spam detection data set (see Section 5.1 for a detailed description). Thus, the outputs (predictions) given by our trained classification model directly reflect the confidence that the target sites (test examples) have changed their organizations within four years. In total we generate 1270 features extracted from all the snapshots of each page.

4.3 Web Spam Classification

We use a two-level classification architecture to combine the results from separate classifiers, each trained on only one group of features. Figure 1 renders the overall architecture. A series of SVM (SVM^{light} implementation) classifiers using linear kernels comprise the low-level portion of the classification architecture. Each low-level SVM classifier is trained using one group of temporal features we introduced in Section 4.1. Here, since the output of SVM^{light} can reflect the distance of each test example to the decision boundary, representing the confidence of decisions, we directly use SVM for the low-level classifiers. The high-level classifier we used is a logistic regression classifier implemented in Weka 3.5.8 [20]. It uses the output values (predictions) given by low-level SVM classifiers as its input feature values, and combines them using logistic regression.

5. EXPERIMENTS

In this section, we present the experimental results. We first show the sensitivity of each group of temporal features with respect to the Web spam classification. We then report the performance by combining the outputs (predictions) generated by multiple low-level classifiers.

5.1 Data Set

We use the WEBSPAM-UK2007 data set to test our Web spam classification approach. 6479 sites are labeled by volunteers, in which about 6% are labeled as spam sites. We select the 3926 sites whose complete historical snapshots can be retrieved from Wayback Machine as our data set, in which 201 sites (5.12%) are labeled as spam, the rest as normal sites. For term-based temporal feature extraction, we download 10 snapshots covering from 2005 to 2007 for each site's homepage and corresponding up to 400 outgoing pages from Internet Archive. We use all the content of downloaded pages within the same site to represent the site content. For organization historical feature extraction, we only use the content of site homepages and extract features from it since the examples in our training set (ODP external pages) are also based on page level.

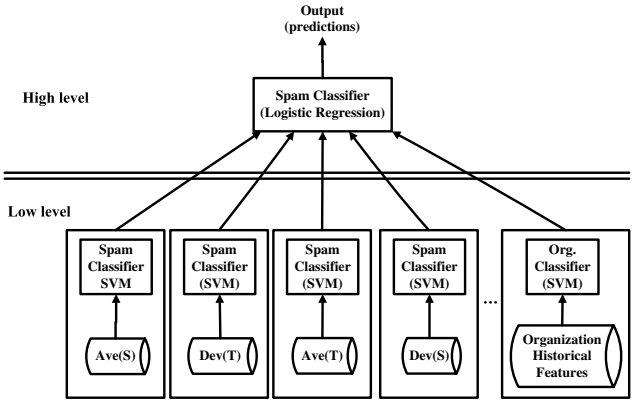

Figure 1: Two-level classification architecture.

Following a previous work [7], we set the parameters $k_1 = 4.2$ and $b = 0.8$ in BM25 calculation.

5.2 Metric

Precision, recall and F-measure are used to measure our spam classification performance.

- **Precision**: the percentage of truly positive examples (spam sites) in those labeled as spam by the classifier;
- **Recall**: the percentage of correctly labeled positive examples (spam sites) out of all positive examples;
- **F-measure**: a balance between **Precision** and **Recall**, which is defined as:

$$\text{F\_measure} = \frac{2 \cdot (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

We also use true positive (tp), false positive (fp), true negative (tn), and false negative (fn) (defined in Table 3) to represent the distribution of classifications in a confusion matrix.

5.3 Low-level Classification

Here, we first present the features' sensitivity with respect to time-span, and then show the spam classification performance of all low-level classifiers.

Recall that we extract 10 snapshots for each page covering 2004 to 2007. We vary the time-span we look backwards (i.e., the number of snapshots) when calculating feature vectors in order to show how each low-level classifier benefits from the historical information. All low-level classifiers, except the one for detecting organization variation, are examined based on five-fold cross-validation.

Notation	Meaning
tp	number of positive examples classified as spam
tn	number of negative examples classified as non-spam
fp	number of negative examples classified as spam
fn	number of positive examples classified as non-spam

Table 3: Base evaluation metrics.

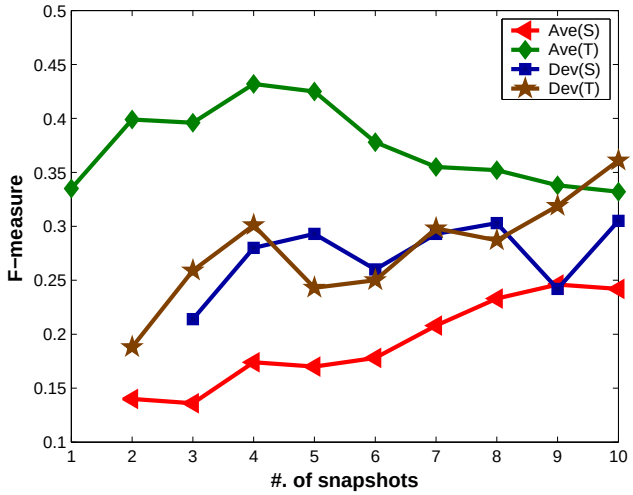

Figure 2: Features' sensitivity on F-measure performance with respected to time-span.

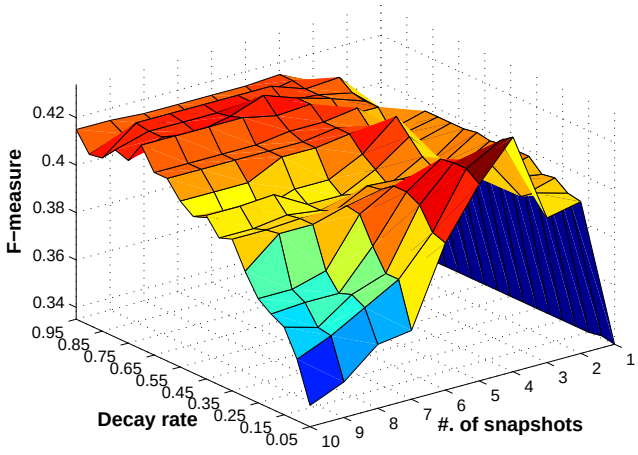

Figure 3: Feature Decay(T)'s sensitivity on F-measure performance with respect to time-span and decay rate.

Since the number of positive examples is small, we use oversampling to emphasize positive examples; i.e., positive examples are replicated 10 times so that the ratio of positive to negative examples in training set is about 1/2.

Figure 2 shows the trends about F-measure spam classification performance with respect to the number of snapshots. The classifier trained by Ave(T) has the best performance in most cases. Its F-measure increases first, but gradually falls when the number of used snapshots is greater than 4. Ave(S) keeps benefiting from the extended time-span until the number of snapshots reaches 9. Dev(S) and Dev(T) have similar trends, but both of them achieve unstable benefits with the extension of time-span. Figure 3 shows the sensitivity of Decay(T)'s F-measure performance with respect to decay rate and time-span. As we expected, the F-measure is more sensitive to the time-span when the decay rate is small, and becomes less sensitive with the increase in decay rate, as it de-emphasizes the influence of the earlier snapshots. The best performance is 0.434, which is achieved when decay rate is 0.05 and the number of snapshots is 4.

Recall that we use selected ODP external pages to train and validate as SVM classifier for detecting whether the target site has changed its organization from 2004 to 2007 (see Section 4.1 for a detailed description). We first perform five-fold cross-validation based on this data set. Precision, recall and F-measure are 0.825, 0.810 and 0.818, respectively. Since both precision and recall are above 80%, we directly apply this classifier on the spam data set and assume that it still can give relatively accurate labels. According to our classifier, 37.31% spam sites have changed their organizations while this figure is lower (14.8%) for non-spam sites.

Our baseline is the SVM classifier (SVM^{light} implementation) trained using the BM25 score feature vectors provided[2] for the current snapshot. We parsed all documents and represent the content of selected sites by concatenating the terms of all pages within the same site together. After removing stop words, the top 22,000 frequent words are chosen as feature words.

The performance comparison among low-level classifiers is presented in Table 4. Here, the performance is based on 10 snapshots by default rather than the best performance by tuning the number of snapshots. We set the decay rate of Decay(T) to be 0.95, hoping that its feature values can be much different from those of Ave(T). We find that Decay(T) has the highest F-measure performance, followed by BM25, Dev(T), and Ave(T). ORG has the lowest F-measure performance, but the highest recall. As we showed before, although spam sites have a higher probability of having changed their ownership, a portion of normal sites also changed.

[2]http://www.yr-bcn.es/webspam/datasets/uk2007/contents/

5.4 High-level Combination

We use logistic regression to combine the scores generated by each low-level SVM classifier. The low-level classifiers are trained using all ten snapshots; our goal is to ascertain whether Web spam classification can benefit from using temporal features, rather than finding the best performance by tuning this parameter for a specific data set. The decay rate used in Decay(T) is set to be 0.95 since we hope this group of features can have much difference from Ave(T). The combination performance is shown in Table 5. All the performance combined by one lower-level classifier and the baseline shows 7.2%–28.7% improvement on F-measure. The best F-measure is 0.527, which outperforms the baseline by more than 30% when combined with Dev(S), Dev(T), and ORG features. Higher recall is usually the reason for high F-measure, since precision performance is much higher in all cases. We also notice that F-measure performance has an overall slight decrease when the number of combined low-level classifiers is greater than 4.

6. DISCUSSION AND FUTURE WORK

In this section, we discuss a few issues, including the impact of using historical information to detect spam pages, some caveats in this work and potential future work.

	Prec.	Rec.	F-meas.	tp	fn	fp	tn
BM25	0.674	0.289	0.404	58	143	28	3697
Dev(S)	0.530	0.214	0.304	43	158	38	3687
Dev(T)	0.529	0.274	0.361	55	146	49	3676
Ave(S)	**0.744**	0.144	0.242	29	172	10	3715
Ave(T)	0.573	0.234	0.332	47	154	35	3690
Decay(T)	0.656	0.303	**0.415**	61	140	32	3693
ORG	0.120	**0.373**	0.181	75	126	552	3173

Table 4: Performance of low-level classifiers when using ten snapshots.

Combination	Precision	Recall	F-measure	tp	fn	fp	tn
BM(baseline)	0.674	0.289	0.404	58	143	28	3697
BM+Dev(S)	0.560	**0.418**	0.479	84	117	66	3659
BM+Dev(T)	0.675	**0.423**	**0.520**	85	116	41	3684
BM+Ave(S)	0.602	0.338	0.433	68	133	45	3680
BM+Ave(T)	0.636	0.348	0.450	70	131	40	3685
BM+Decay(T)	0.664	0.373	0.478	75	126	38	3687
BM+ORG	0.640	0.353	0.455	71	130	40	3685
BM+Dev(S)+Dev(T)	0.635	**0.433**	**0.515**	87	114	50	3675
BM+Dev(S)+Ave(S)	0.602	0.353	0.445	71	130	47	3678
BM+Dev(S)+Ave(T)	0.645	0.398	0.492	80	121	44	3681
BM+Dev(S)+Decay(T)	0.621	0.408	0.492	82	119	50	3675
BM+Dev(S)+ORG	0.579	**0.418**	0.486	84	117	61	3664
BM+Dev(T)+Ave(S)	0.600	0.343	0.437	69	132	46	3679
BM+Dev(T)+Ave(T)	**0.694**	0.373	0.485	75	126	33	3692
BM+Dev(T)+Decay(T)	0.664	0.403	0.502	81	120	41	3684
BM+Dev(T)+ORG	0.689	**0.418**	**0.520**	84	117	38	3687
BM+Ave(S)+Ave(T)	0.585	0.343	0.433	69	132	49	3676
BM+Ave(S)+Decay(T)	0.568	0.353	0.436	71	130	54	3671
BM+Ave(S)+ORG	0.619	0.348	0.446	70	131	43	3682
BM+Ave(T)+Decay(T)	0.667	0.358	0.466	72	129	36	3689
BM+Ave(T)+ORG	**0.726**	0.343	0.466	69	132	26	3699
BM+Decay(T)+ORG	**0.721**	0.373	0.492	75	126	29	3696
BM+Dev(S)+Dev(T)+Ave(S)	0.559	0.353	0.433	71	130	56	3669
BM+Dev(S)+Dev(T)+Ave(T)	0.609	0.403	0.485	81	120	52	3673
BM+Dev(S)+Dev(T)+Decay	0.618	0.403	0.488	81	120	50	3675
BM+Dev(S)+Dev(T)+ORG	0.650	**0.443**	**0.527**	89	112	48	3677
BM+Dev(S)+Ave(S)+Ave(T)	0.578	0.333	0.423	67	134	49	3676
BM+Dev(S)+Ave(S)+Decay(T)	0.580	0.378	0.458	76	125	55	3670
BM+Dev(S)+Ave(S)+ORG	0.608	0.363	0.455	73	128	47	3678
BM+Dev(S)+Ave(T)+Decay(T)	0.658	0.393	0.492	79	122	41	3684
BM+Dev(S)+Ave(T)+ORG	0.650	0.378	0.478	76	125	41	3684
BM+Dev(S)+Decay(T)+ORG	0.656	0.408	**0.503**	82	119	43	3682
BM+Dev(T)+Ave(S)+Ave(T)	0.568	0.313	0.404	63	138	48	3677
BM+Dev(T)+Ave(S)+Decay(T)	0.571	0.358	0.440	72	129	54	3671
BM+Dev(T)+Ave(S)+ORG	0.617	0.353	0.449	71	130	44	3681
BM+Dev(T)+Ave(T)+Decay(T)	0.685	0.378	0.487	76	125	35	3690
BM+Dev(T)+Ave(T)+ORG	**0.699**	0.358	0.474	72	129	31	3694
BM+Dev(T)+Decay(T)+ORG	**0.699**	0.393	**0.503**	79	122	34	3691
BM+Ave(S)+Ave(T)+Decay(T)	0.579	0.363	0.446	73	128	53	3672
BM+Ave(S)+Ave(T)+ORG	0.588	0.333	0.425	67	134	47	3678
BM+Ave(S)+Decay(T)+ORG	0.559	0.353	0.433	71	130	56	3669
BM+Ave(T)+Decay(T)+ORG	0.680	0.348	0.461	70	131	33	3692
BM+Dev(S)+Dev(T)+Ave(S)+Ave(T)	0.544	0.338	0.417	68	133	57	3668
BM+Dev(S)+Dev(T)+Ave(S)+Decay(T)	0.533	0.363	0.432	73	128	64	3661
BM+Dev(S)+Dev(T)+Ave(S)+ORG	0.566	0.363	0.442	73	128	56	3669
BM+Dev(S)+Dev(T)+Ave(T)+Decay(T)	0.642	0.393	0.488	79	122	44	3681
BM+Dev(S)+Dev(T)+Ave(T)+ORG	0.642	0.383	0.480	77	124	43	3682
BM+Dev(S)+Dev(T)+Decay(T)+ORG	0.667	0.398	0.498	80	121	40	3685
BM+Dev(S)+Ave(S)+Ave(T)+Decay(T)	0.567	0.358	0.439	72	129	55	3670
BM+Dev(S)+Ave(T)+Decay(T)+ORG	0.679	0.378	0.486	76	125	36	3689
BM+Dev(T)+Ave(S)+Ave(T)+Decay(T)	0.534	0.308	0.391	62	139	54	3671
BM+Dev(T)+Ave(T)+Decay(T)+ORG	**0.714**	0.373	0.490	75	126	30	3695
BM+Ave(S)+Ave(T)+Decay(T)+ORG	0.567	0.358	0.439	72	129	55	3670
BM+Dev(S)+Dev(T)+Ave(S)+Ave(T)+Decay(T)	0.507	0.338	0.406	68	133	66	3659
BM+Dev(S)+Dev(T)+Ave(S)+Ave(T)+ORG	0.541	0.328	0.409	66	135	56	3669
BM+Dev(S)+Dev(T)+Ave(S)+Decay(T)+ORG	0.522	0.353	0.421	71	130	65	3660
BM+Dev(S)+Dev(T)+Ave(T)+Decay(T)+ORG	0.648	0.393	0.489	79	122	43	3682
BM+Dev(S)+Ave(S)+Ave(T)+Decay(T)+ORG	0.565	0.348	0.431	70	131	54	3671
BM+Dev(T)+Ave(S)+Ave(T)+Decay(T)+ORG	0.549	0.308	0.395	62	139	51	3674
BM+Dev(S)+Dev(T)+Ave(S)+Ave(T)+Decay(T)+ORG	0.539	0.343	0.419	69	132	59	3666

Table 5: Combined performance.

6.1 Impact

Page historical information provides a valuable complementary resource to help classify Web spam since spam pages and normal pages show different patterns of evolution. The idea and the features that we use can be generalized to a variety of other applications. For example, the anchor text on the links which point to target pages are often not updated promptly since the content and topics of target pages change over time without notifying the links pointing to them. By tracking how target pages evolve, we can automatically identify whether the anchor text has become stale or not, which may also be helpful on detecting hacked sites and/or paid links. Further, we can infer pages' freshness which may influence the authority scores pages should be given, and therefore, potentially influence link-based authority calculation. In addition, document classification algorithms which utilize information from neighbor nodes may also benefit from this idea since a neighbor referenced by a stale link should have a lowered contribution to the content representation of the target page.

In this work, we track the evolution patterns mainly based on documents' term-level content representation, and use them as signals for the page/site quality measurement (i.e., determining whether the site is a spam site). From the experiments, we notice that most of combined performance have high precision (around 0.7). Therefore, we infer that these trained classification models can be potentially used in the post-processing of search results to filter the spam pages that would otherwise appear in the returned search results.

6.2 Caveats

Although we have shown that temporal features extracted from historical snapshots are useful in classification of web spam, our proposed method may suffer from the following limitations.

- The manual judgement regarding whether the target page has changed ownership is only based on the organization information shown on the page and whether the page has changed its topics dramatically if the ownership information is absent. However, accurate page ownership information should be confirmed by historical WHOIS registration data. We expect that the classification model (ORG) would be more accurate if WHOIS data were included.
- We mainly focus on the evolution patterns of BM25 scores which are based on term vector space features, and whether the sites change their organizational ownership within the past four years. Many other features which reflect page content variation can be considered, such as topic change and so on. The term-level features only reflect the finest granularity with respect to document topics.
- Term-based features are too sensitive to the terms selected from the vocabulary, which might limit the generalization to some extent.

While we show the classification performance (F-measure) has prominent improvement when combining temporal features with the baseline, the absolute performance is not very high. We summarize the main reasons as follows:

- We assume that the classifier for identifying sites which have changed organizational ownership within the past four years can correctly classify the sites in the WEBSPAM-UK2007 dataset since the classification performance on ODP datasets is high (both precision and recall are above 0.8 based on five-fold cross-validation). However, the misclassified data points do introduce noise into the Web spam classification.
- The examples we select are the ones for which the Internet Archive can provide complete historical snapshots (at least for each site's home page) from 2004 to 2007. Therefore, the ratio of spam sites (around 5%) is lower than that in the whole WEBSPAM-UK2007 dataset (around 6%). This selection of examples may influence the classification performance.
- We use a site's homepage along with up to 400 first-level site subdirectory pages to represent a web site. However, the subdirectory pages may not have consistent snapshots (i.e., some subdirectory pages only have historical snapshots in 2004, but not in 2006), which influences the classification performance.
- SVM classifiers prefer the majority class and aim to maximize the accuracy metric, which is given by:

$$\text{accuracy} = \frac{\text{the number of correctly classified examples}}{\text{the total number of examples}}$$

  We did not tune the cost factor but directly copied positive examples ten times such that the ratio of positive to negative examples is 1/2.

6.3 Future work

Tuning the number of snapshots in classification models. The combined performance presented in this work is only based on the features extracted from 10 snapshots. We can mix the procedure of tuning the best number of snapshots we should use into training this two-level classification architecture, which may further improve performance. Besides, it is also helpful to investigate the sensitivity of spam classification performance with respect to the time-span in which pages change organizations.

Examining alternative classifiers. In this work, we use SVMs to train our low-level classification models. In the future, we might generalize it onto other classification models, such as decision trees, etc., and investigate the time-related sensitivity (the variation of the number of snapshots used) with respect to Web spam classification performance.

Combining other temporal features. Most of the features proposed in this work are based on content. Given enough temporal link information, we can extract similar temporal features through the variation of link information for each site, which may be more generalizable for the spam classification task. In the future, we plan to extract link-based and/or other temporal features, and compare the performance with classifiers trained using the current content-based temporal features. For example, the features extracted from the variation of anchor text within pages and the in-links and out-links may contain plenty of useful signals on both page quality change and organization change[3]. It may be also valuable to investigate the influence of changes in nameservers, WHOIS information, and site ages on page quality analysis. In the future, we will combine the temporal content features and other temporal features together to test how much these features can boost the Web spam classification performance. Besides, we will examine the incorrectly classified examples, analyze the reasons, and summarize the discovered error types. We also plan to investigate how much overlap these feature groups have with those that are useful for Web spam classification.

[3]It is noted that some page-level temporal link features are only used in the ORG classifier.

7. SUMMARY

Spam pages and normal pages have different evolution patterns with respect to page content and links. In this work, we show that historical information can be a useful complementary resource for Web spam classification. We introduce several groups of temporal features mainly based on historical content variation, and demonstrate their capability for spam detection in the WEBSPAM-UK2007 data set. The best F-measure performance of combining multiple classifiers trained using different groups of features outperforms our textual baseline (the BM25 score vectors calculated using current site content) by 30%.

Acknowledgments

This work was supported in part by grants from the National Science Foundation under award IIS-0803605 and IIS-0545875, and an equipment grant from Sun Microsystems. We also thank Jian Wang and Liangjie Hong for helpful discussions.

8. REFERENCES

[1] A. Acharya, M. Cutts, J. Dean, P. Haahr, M. Henzinger, U. Hoelzle, S. Lawrence, K. Pfleger, O. Sercinoglu, and S. Tong. Information retrieval based on historical data. United States Patent 20050071741, USPTO, Mar. 2005.

[2] R. Andersen, C. Borgs, J. Chayes, J. Hopcroft, K. Jain, V. Mirrokni, and S. Teng. Robust pagerank and locally computable spam detection features. In *Proceedings of the Third International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pages 69–76, Apr. 2008.

[3] J. Attenberg and T. Suel. Cleaning search results using term distance features. In *Proceedings of the Fourth International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pages 21–24, Apr. 2008.

[4] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Link-based characterization and detection of web spam. In *Proceedings of the Second International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pages 1–8, Aug. 2006.

[5] A. A. Benczur, K. Csalogany, T. Sarlos, and M. Uher. SpamRank – Fully automatic link spam detection. In *Proceedings of the First International Workshop on Adversarial Information Retrieval (AIRWeb)*, May 2005.

[6] I. Biro, J. Szabo, and A. Benczur. Latent dirichlet allocation in web spam filtering. In *Proceedings of the Fourth International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pages 21–24, Apr. 2008.

[7] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma. Block-level link analysis. In *Proc. of the 27th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, Sheffield, UK, July 2004.

[8] B. D. Davison. Recognizing nepotistic links on the Web. In *Artificial Intelligence for Web Search*, pages 23–28. AAAI Press, July 2000. Presented at the AAAI-2000 workshop on Artificial Intelligence for Web Search, Technical Report WS-00-01.

[9] Google Inc. Google home page. `http://www.google.com/`, 2009.

[10] Z. Gyöngyi, P. Berkhin, H. Garcia-Molina, and J. Pedersen. Link spam detection based on mass estimation. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, pages 63–72, Seoul, Korea, 2006.

[11] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, Toronto, Canada, 2004.

[12] Internet Archive. The Internet Archive. http://www.archive.org/, 2009.

[13] T. Joachims. Making large-scale support vector machine learning practical. In *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

[14] Y.-R. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. L. Tseng. Splog detection using self-similarity analysis on blog temporal dynamics. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pages 1–8, New York, NY, 2007. ACM Press.

[15] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th International Conference on the World Wide Web*, May 2006.

[16] The dmoz Open Directory Project (ODP), 2009. `http://www.dmoz.org/`.

[17] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proc. of the 17th Annual Int'l ACM SIGIR Conf. on Research and Development in Info. Retrieval*, pages 232–241, 1994.

[18] G. Shen, B. Gao, T.-Y. Liu, G. Feng, S. Song, and H. Li. Detecting link spam using temporal information. In *Proc. of IEEE International Conference on Data Mining (ICDM)*, pages 1049–1053, 2006.

[19] T. Urvoy, E. Chauveau, P. Filoche, and T. Lavergne. Tracking web spam with HTML style similarities. *ACM Transactions on the Web*, 2(3), 2008.

[20] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2005. Second edition.

[21] B. Wu and B. D. Davison. Identifying link farm spam pages. In *Proceedings of the 14th International World Wide Web Conference*, pages 820–829, Chiba, Japan, May 2005.

[22] B. Wu and B. D. Davison. Detecting semantic cloaking on the web. In *Proceedings of the 15th International World Wide Web Conference*, pages 819–828, Edinburgh, Scotland, May 2006.

[23] B. Wu, V. Goel, and B. D. Davison. Topical TrustRank: Using topicality to combat web spam. In *Proceedings of the 15th International World Wide Web Conference*, pages 63–72, Edinburgh, Scotland, May 2006.