Internet Mathematics Vol. 10: 421–457

The Classification Power of Web Features

Miklós Erdélyi, András A. Benczúr, Bálint Daróczy, András Garzó, Tamás Kiss, and Dávid Siklósi

Abstract. In this article we give a comprehensive overview of features devised for web spam detection and investigate how much various classes, some requiring very high computational effort, add to the classification accuracy.

- We collect and handle a large number of features based on recent advances in web spam filtering, including temporal ones; in particular, we analyze the strength and sensitivity of linkage change.
- We propose new, temporal link-similarity-based features and show how to compute them efficiently on large graphs.
- We show that machine learning techniques, including ensemble selection, LogitBoost, and random forest significantly improve accuracy.
- We conclude that, with appropriate learning techniques, a simple and computationally inexpensive feature subset outperforms all previous results published so far on our dataset and can be further improved only slightly by computationally expensive features.
- We test our method on three major publicly available datasets: the Web Spam Challenge 2008 dataset WEBSPAM-UK2007, the ECML/PKDD Discovery Challenge dataset DC2010, and the Waterloo Spam Rankings for ClueWeb09.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uinm.

ISSN: 1542-7951 print

Our classifier ensemble sets the strongest classification benchmark compared to participants of the Web Spam and ECML/PKDD Discovery Challenges as well as the TREC Web track.

To foster research in the area, we make several feature sets and source codes public,[1] including the temporal features of eight `.uk` crawl snapshots that include WEBSPAM-UK2007 as well as the Web Spam Challenge features for the labeled part of ClueWeb09.

1. Introduction

Web classification finds several uses, both for content filtering and for building focused corpora from a large-scale web crawl. As one notable use, Internet archives actively participate in large-scale experiments [Benczúr et al. 09], some of them building analytics services over their collections [Barton 12]. Most of the existing results on web classification originate from the area of web spam filtering that have turned out to generalize a wide class of tasks including genre and Open Directory category, as well as quality classification. Closely related areas include filtering and tagging in social networks [Hotho et al. 08].

Web spam filtering, the area of devising methods to identify useless web content with the sole purpose of manipulating search engine results, has drawn much attention in the past years [Singhal 04, Henzinger et al. 02, Gyöngyi and Garcia-Molina 05a]. The first mention of web spam, termed *spamdexing* as a combination of words *spam* and (search engine) *indexing*, appeared probably in a 1996 news article [Convey 96] as part of the early web era discussions on the spreading porn content [Chekuri et al. 97]. In the area of the so-called Adversarial Information Retrieval workshop series, run since 2005 [Fetterly and Gyöngyi 09], evaluation campaigns including the Web Spam Challenges [Castillo et al. 08], the ECML/PKDD Discovery Challenge 2010 [Hotho et al. 08], and the Spam task of TREC 2010 Web Track [Cormack et al. 11] were organized. A recent comprehensive survey on web spam filtering research is found in [Castillo and Davison 11].

In this article we present, to our best knowledge, the most comprehensive experimentation based on content and link range as well as temporal features, both new and recently published. Our spam filtering baseline classification procedures are collected by analyzing the results [Cormack 07, Abernethy et al. 08, Geng et al. 08] of the Web Spam Challenges and the ECML/PKDD Discovery Challenge 2010 [Geng et al. 10, Sokolov et al. 10, Nikulin 10]. Our comparison is based

[1] `https://datamining.sztaki.hu/en/download/web-spam-resources`

on area under the curve (AUC) values [Fogarty et al. 05] that we believe to be more stable because they do not depend on the split point; indeed, although Web Spam Challenge 2007 used F-measure and AUC, Web Spam Challenge 2008 used AUC only as an evaluation measure.

Web spam appears in sophisticated forms that manipulate content as well as linkage [Gyöngyi and Garcia-Molina 05b] with examples such as

- Copied content, "honey pots" that draw attention but link to unrelated spam targets;
- Garbage content, stuffed with popular or monetizable query terms and phrases such as university degrees, online casinos, bad credit status or adult content;
- Link farms, a large number of strongly interlinked pages across several domains.

The web spammer toolkit consists of a clearly identifiable set of manipulation techniques that has not changed much recently. The Web Spam Taxonomy of [Gyöngyi and Garcia-Molina 05b] distinguishes content (term) and link spamming along with techniques of hiding, cloaking, and removing traces by, e.g., obfuscated redirection. Most of the features designed fight either link or content spamming.

We realize that recent results have ignored the importance of the machine learning techniques and concentrated only on the definition of new features. Also, the only earlier attempt to unify a large set of features [Castillo et al. 06] is already four years old, and even there, little comparison is given on the relative power of the feature sets. For classification techniques, a wide selection including decision trees, random forest, support vector machine (SVM), class-feature-centroid, boosting, bagging, and oversampling, in addition to feature selection (Fisher, Wilcoxon, Information Gain) were used [Geng et al. 10, Sokolov et al. 10, Nikulin 10] but never compared and combined. In this study we address the following questions.

- Do we get the maximum value out of the features we have? Are we sufficiently sophisticated at applying machine learning?
- Is it worth calculating computationally expensive features, in particular those related to page-level linkage?
- What is an optimal feature set for a fast spam filter that can quickly react at crawl time after fetching a small sample of a website?

We compare our result with the very strong baselines of the Web Spam Challenge 2008 and ECML/PKDD 2010 Discovery Challenge datasets. Our main results are as follows.

- We apply state-of-the-art classification techniques by the lessons learned from the KDD Cup 2009 [Niculescu-Mizil et al. 09]. Key in our performance is ensemble classification applied over different feature subsets as well as over different classifiers over the same features. We also apply classifiers yet unexplored against web spam, including random forest [Breiman 01] and LogitBoost [Friedman et al. 00].
- We compile a small yet very efficient feature set that can be computed by sample pages from the site while completely ignoring linkage information. By this feature set, a filter may quickly react to a recently discovered site and intercept in time before the crawler would start to follow a large number of pages from a link farm. This feature set itself reaches AUC 0.893 over WEBSPAM-UK2007.
- Last, but not least, we gain strong improvements over the Web Spam Challenge best performance [Castillo et al. 08]. Our best result in terms of AUC reaches 0.9 and improves on the best Discovery Challenge 2010 results.

Several recent studies propose temporal features [Shen et al. 06, Lin et al. 07, Dai et al. 09, Chung et al. 09] to improve classification accuracy. We extend link-based similarity algorithms by proposing metrics to capture the linkage change of webpages over time. We describe a method to calculate these metrics efficiently on the web graph and then measure their performance when used as features in web spam classification. We propose an extension of two link-based similarity measures: XJaccard and PSimRank [Fogaras and Rácz 05].

We investigate the combination of temporal and nontemporal, both link- and content-based features using ensemble selection. We evaluate the performance of ensembles built on the latter feature sets and compare our results to that of state-of-the-art techniques reported on our dataset. Our conclusion is that temporal and link-based features in general do not significantly increase Web spam filtering accuracy. However, information about linkage change might improve the performance of a language independent classifier: the best results for the French and German classification tasks of the ECML/PKDD Discovery Challenge [Geng et al. 10] were achieved by using host level link features only, outperforming those who used all features [Sokolov et al. 10].

In this work we address not just the quality but also the computational efficiency. Earlier lightweight classifiers include [Webb et al. 08] describing a procedure based solely on the HTTP session information. Unfortunately, they measure only precision, recall, and F-measure that are difficult to compare with later

results on web spam that use AUC. In fact, the F and similar measures greatly depend on the classification threshold and, hence, make comparison less stable; for this reason they are not used, starting with the Web Spam Challenge 2008. Furthermore, in [Webb et al. 08] the IP address is a key feature that is trivially incorporated in the DC2010 dataset by placing all hosts from the same IP address into the same training or testing set. The intuition is that if an IP address contains spam hosts, all hosts from that IP address are likely to be spam and should be immediately manually checked and excluded from further consideration.

The rest of this article is organized as follows. In Section 2 we describe the datasets used in this paper. We give an overview of temporal features for spam detection and propose new temporal link-similarity-based ones in Section 3. In Section 4 we describe our classification framework. The results of the experiments to classify WEBSPAM-UK2007, ClueWeb09, and DC2010 can be found in Section 5. The computational resource needs of various feature sets are summarized in Section 6.

2. Datasets

In this paper we use three datasets, WEBSPAM-UK2007 of the Web Spam Challenge 2008 [Castillo et al. 08], the Waterloo Spam Rankings for ClueWeb09, and DC2010 created for the ECML/PKDD Discovery Challenge 2010 on Web Quality. We give only a brief summary of the first dataset—described well in [Castillo et al. 08, Castillo et al. 07] and the second in [Erdélyi et al. 11]—however, we describe the third in more detail in Section 2.3. Also, we compare the amount of spam in the datasets.

2.1. Web Spam Challenge 2008: WEBSPAM-UK2007

The Web Spam Challenge was first organized in 2007 over the WEBSPAM-UK2006 dataset. The last Challenge over the WEBSPAM-UK2007 set was held in conjunction with AIRWeb 2008 [Castillo et al. 08]. The Web Spam Challenge 2008 best result [Geng et al. 08] achieved an AUC of 0.85 by also using ensemble undersampling [Chawla et al. 04]. They trained a bagged classifier on the standard content-based and link-based features published by the organizers of the Web Spam Challenge 2008 and on custom host-graph-based features, using the ERUS strategy for class-inbalance learning. For earlier challenges, best performances were achieved by a semisupervised version of the support vector machine (SVM) [Abernethy et al. 08] and text compression [Cormack 07]. Best results either used bag of words vectors or the so-called "public" feature sets of [Castillo et al. 06].

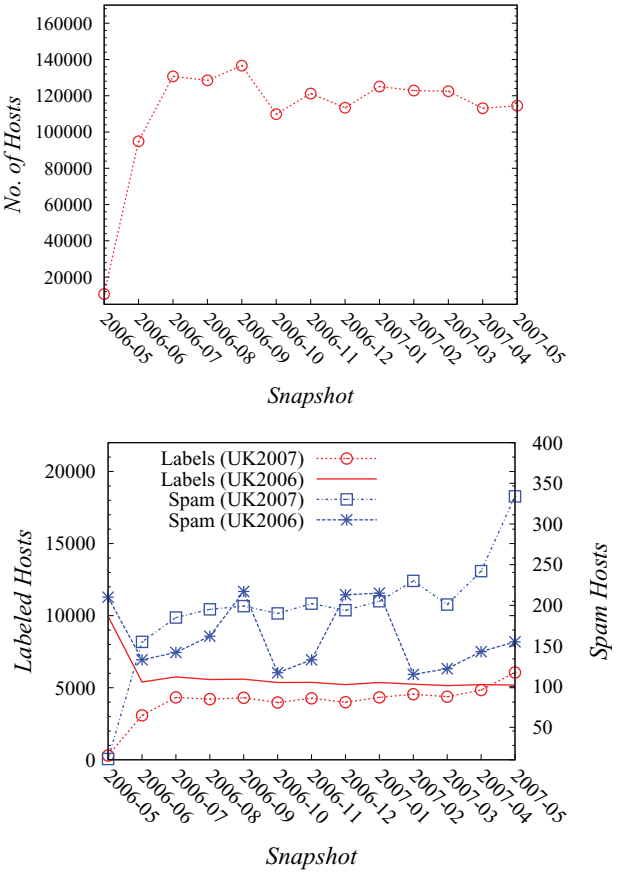

Figure 1. The number of total and labeled hosts in the 13 UK snapshots. We indicate the number of positive and negative labels separate for the WEBSPAM-UK2006 and WEBSPAM-UK2007 label sets.

Label Set	Instances	%Positive
Training	4000	5.95%
Testing	2053	4.68%

Table 1. Summary of label sets for Web Spam Challenge 2008.

We extended the WEBSPAM-UK2007 dataset with 13 .uk snapshots provided by the Laboratory for Web Algorithmics of the Università degli studi di Milano. We use the training and testing labels of the Web Spam Challenge 2008, as summarized in Table 1. In order to prepare a temporal collection, we extracted a maximum of 400 pages per site from the original crawls. The last 12 of the .uk snapshots were analyzed by [Bordino et al. 08] who observed a relative low URL but high host overlap.[2] The first snapshot (2006-05) that is identical to WEBSPAM-UK2006 was chosen to be left out from their experiment because it was provided by a different crawl strategy. We observed that, in the last eight snapshots, the number of hosts have stabilized in the sample and these snapshots have roughly the same number of labeled hosts as seen in Figure 1. From now on we restrict attention to the aforementioned subset of the snapshots and the WEBSPAM-UK2007 labels only.

2.2. The Waterloo Spam Rankings for ClueWeb09

The English part of ClueWeb09 consists of approximately 20 M domains and 500 M pages. For web spam labels we used the Waterloo Spam Rankings [Cormack et al. 11]. Although the Waterloo Spam Rankings contain negative training instances as well, we extended the negative labels with the set of the Open Directory Project (ODP) hosts. We used a 50-50% split for training and testing.

We labeled hosts in both the .pt crawl and ClueWeb09 by top-level ODP categories using links extracted from topic subtrees in the directory. Out of all labeled hosts, 642, 643 received a unique label. Because certain sites (e.g., bbc.co.uk) may belong to even all 14 top-level English categories, we discarded the labels of 18, 734 hosts with multiple labels to simplify the multilabel task. As [Bordino et al. 10] indicate, multitopical hosts are often associated with poor quality sites and spam as another reason why their labels may mislead the classification process. The resulting distribution of labels is shown in Table 2.

2.3. Discovery Challenge 2010: DC2010

The Discovery Challenge was organized over DC2010, a new dataset that we describe in more detail in the next section. DC2010 is a large collection of annotated web hosts labeled by the Hungarian Academy of Sciences (English documents), Internet Memory Foundation (French), and L3S Hannover (German). The base data is a set of 23 M pages in 190 K hosts in the .eu domain crawled by the Internet Memory Foundation in early 2010. The labels extend the scope of

[2] The dataset can be downloaded from: http://law.di.unimi.it/datasets.php

Category	No. of Hosts	% of Labeled Hosts
spam	439	0.07%
Arts	97355	15.1%
Business	193678	30.1%
Computers	66159	10.3%
Recreation	65594	10.2%
Science	43317	6.7%
Society	122084	19 %
Sports	54456	8.5%

Table 2. Number of positive ClueWeb09 host labels for spam and the ODP categories.

previous datasets on web spam in that, in addition to sites labeled spam, we included manual classification for genre into five categories: Editorial, Commercial, Educational, Discussion, and Personal, as well as trust, factuality, and bias as three aspects of quality. The spam label is exclusive because no other assessment was made for spam. However, other labels are nonexclusive and, hence, define nine binary classification problems. We consider no multiclass tasks in this work.

				DC2010			
	UK2006	**UK2007**	**ClueWeb09**	en	de	fr	all
Hosts	10 660	114 529	500,000	61 703	29 758	7 888	190 000
Spam	19.8%	5.3%	unknown	8.5% of valid labels; 5% of all in large domains.			

Table 3. Fraction of Spam in WEBSPAM-UK2006, UK2007, ClueWeb09 and DC2010. Note that three languages—English, German, and French—were selected for labeling DC2010, although Polish and Dutch language hosts constitute a larger fraction than the French. Because to our best knowledge, no systematic random sample was labeled for ClueWeb09, the number 439 of labeled spam hosts is not representative for the collection.

Count	IP address	Comment
3544	80.67.22.146	spam farm `*-palace.eu`
3198	78.159.114.140	spam farm `*auts.eu`
1374	62.58.108.214	`blogactiv.eu`
1109	91.204.162.15	spam farm `x-mp3.eu`
1070	91.213.160.26	spam farm `a-COUNTRY.eu`
936	81.89.48.82	`autobazar.eu`
430	78.46.101.76	spam farm `77k.eu` and 20+ domains
402	89.185.253.73	spam farm `mp3-stazeni-zdarma.eu`

Table 4. Selection of IP addresses with many subdomains in the DC2010 dataset.

Assessor instructions are, for example, summarized in [Siklósi et al. 12], a study concentrating on quality labels.

In Table 3, we summarize the amount of spam in the DC2010 dataset in comparison with the Web Spam Challenge datasets. This amount is well-defined for the latter datasets by the way they were prepared for the Web Spam Challenge participants. However, for DC2010, this figure may be defined in several ways. First of all, when creating the DC2010 labels, eventually we considered domains with or without a `www.` prefix the same, such as `www.domain.eu` vs. `domain.eu`. However, in our initial sampling procedure we considered them as two different hosts and merged them after verifying that the labels of the two versions were identical. Also, several domains consist of a single redirection page to another domain, and we counted these domains, too. Finally, a large fraction of spam is easy to spot and can be manually removed. As an example of many hosts on the same IP, we include a labeled sample from DC2010, which itself contains over 10,000 spam domains in Table 4. These hosts were identified by manually looking at the IP addresses that serve the largest number of domain names. Thus, our sample is biased, and obtaining an estimate of the spam fraction is nontrivial, as indicated in Table 3.

The distribution of labels for the nine categories with more than 1% positive samples (spam, news, commercial, educational, discussion, personal, neutral, biased, trusted) is given in Table 5. For neutrality and trust, the strong negative categories have low frequency and, hence, we fused them with the intermediate negative (maybe) category for the training and testing labels.

Label	Group	Yes	Maybe	No
Spam	*Spam*	423		4 982
News/Editorial	*Genre*	191		4 791
Commercial		2 064		2 918
Educational		1 791		3 191
Discussion		259		4 724
Personal-Leisure		1 118		3 864
Nonneutrality	*Quality*	19	216	3 778
Bias		62		3 880
Distrustiness		26	201	3 786

Table 5. Distribution of assessor labels in the DC2010 dataset.

The Discovery Challenge 2010 best result [Nikulin 10] achieved an AUC of 0.83 for spam classification whereas the overall winner [Geng et al. 10] was able to classify a number of quality components at an average AUC of 0.80. As for the technologies, bag of words representation variants proved to be very strong for the English collection, although only language independent features were used for German and French. The applicability of dictionaries and cross-lingual technologies remains open.

New to the construction of the DC2010 training and test set is the handling of hosts from the same domain and IP address. Because no IP address and domain was allowed to be split between training and testing, we might have to reconsider the applicability of propagation [Gyöngyi et al. 04, Wu et al. 06] and graph stacking [Kou and Cohen 07]. The Web Spam Challenge datasets were labeled by uniform random sampling, and graph stacking appeared to be efficient in several results [Castillo et al. 07] including our prior work [Csalogány et al. 07]. The applicability of graph stacking remains, however, unclear for the DC2010 dataset. Certain teams used some of these methods but reported no improvement [Sokolov et al. 10].

3. Temporal Features for Spam Detection

Spammers often create bursts in linkage and content: they might add thousands or even millions of machine-generated links to pages that they want to promote

[Shen et al. 06] that they again very quickly regenerate for another target or remove if blacklisted by search engines. Therefore, changes in both content and linkage can characterize spam pages.

Recently, the evolution of the web has attracted interest in defining features, signals for ranking [Dong et al. 10], and spam filtering [Shen et al. 06, Lin et al. 07, Dai et al. 09, Chung et al. 09, Erdélyi et al. 09]. The earliest results investigate the changes of web content with the primary interest of keeping a search engine index up-to-date [Cho and Garcia-Molina 00a,b]. The decay of webpages and links and its consequences on ranking are discussed in [Bar-Yossef et al. 04, Eiron et al. 04]. One main goal of [Boldi et al. 08], who collected the `.uk` crawl snapshots also used in our experiments, was the efficient handling of time-aware graphs. Closest to our temporal features is the investigation of host overlap, deletion and content dynamics in the same dataset by [Bordino et al. 08].

Perhaps the first result on the applicability of temporal features for web spam filtering is due to [Shen et al. 06] who compare pairs of crawl snapshots and define features based on the link growth and death rate. However, by extending their ideas to consider multistep neighborhoods, we are able to define a very strong feature set that can be computed by the Monte Carlo estimation of [Fogaras and Rácz 05]. Another result defines features based on the change of the content [Dai et al. 09] by those who obtain page history from the Wayback Machine.

For calculating the temporal link-based features, we use the host-level graph. As observed in [Bordino et al. 08], pages are much more unstable over time compared to hosts. Note that page-level fluctuations could simply result from the sequence of the pages the crawler visited and not necessarily reflect real changes. The inherent noise of the crawling procedure and problems with URL canonization [Bar-Yossef et al. 09] rule out the applicability of features based on the change of page-level linkage.

3.1. Linkage Change

In this section we describe link-based temporal features that capture the extent and nature of linkage change. These features can be extracted from either the page or the host-level graph where the latter has a directed link from host a to host b if there is a link from a page of a to a page of b.

The starting point of our new features is the observation of [Shen et al. 06] that the in-link growth and death rate and change of clustering coefficients characterize the evolution patterns of spam pages. We extend these features for the multistep neighborhood in the same way as PageRank extends the in-degree. The *ℓ-step neighborhood* of page v is the set of pages reachable from v over a path

of length at most ℓ. The ℓ-step neighborhood of a host can be defined similarly over the host graph.

We argue that the changes in the multistep neighborhood of a page should be more indicative of the spam or honest nature of the page than its single-step neighborhood because spam pages are mostly referred to by spam pages [Castillo et al. 07], and spam pages can be characterized by a larger change of linkage when compared to honest pages [Shen et al. 06].

In the following, we review the features related to linkage growth and death from [Shen et al. 06] in Section 3.1.1, then we introduce new features based on the similarity of the multistep neighborhood of a page or host. We show how the XJaccard and PSimRank similarity measures can be used for capturing linkage change in Section 3.1.3 and Section 3.1.4, respectively.

3.1.1. Change Rate of In-links and Out-links. We compute the following features introduced by Shen et al. [Shen et al. 06] on the host level for a node a for graph instances from time t_0 and t_1. We let $G(t)$ denote the graph instance at time t and $I^{(t)}(a)$, $\Gamma^{(t)}(a)$ denote the set of in and out-links of node a at time t, respectively.

- In-link death (IDR) and growth rate (IGR):

$$\mathrm{IDR}(a) = \frac{|I^{(t_0)}(a)| - |I^{(t_0)}(a) \cap I^{(t_1)}(a)|}{|I^{(t_0)}(a)|}.$$

$$\mathrm{IGR}(a) = \frac{|I^{(t_1)}(a)| - |I^{(t_0)}(a) \cap I^{(t_1)}(a)|}{|I^{(t_0)}(a)|}.$$

- Out-link death and growth rates (ODR, OGR): the above features calculated for out-links;
- Mean and variance of IDR, IGR, ODR, and OGR across in-neighbors of a host (IDRMean, IDRVar, etc.);
- Change rate of the clustering coefficient (CRCC), i.e., the fraction of linked hosts within those pointed by pairs of edges from the same host:

$$CC(a,t) = \frac{|\{(b,c) \in G(t) | b,c \in \Gamma^{(t)}(a)|}{|\Gamma^{(t)}(a)|}.$$

$$CRCC(a) = \frac{CC(a,t_1) - CC(a,t_0)}{CC(a,t_0)}.$$

- Derivative features such as the ratio and product of the in- and out-link rates, means, and variances. We list the in-link derivatives; out-link derivatives are defined similarly:

  $$\text{IGR} \cdot \text{IDR}, \quad \text{IGR}/\text{IDR}, \quad \text{IGRMean}/\text{IGR}, \quad \text{IGRVar}/\text{IGR}, \quad \text{IDRMean}/\text{IDR}, \quad \text{IDRVar}/\text{IDR}, \quad \text{IGRMean} \cdot \text{IDRMean}, \quad \text{IGRMean}/\text{IDRMean}, \quad \text{IGRVar} \cdot \text{IDRVar}, \quad \text{IGRVar}/\text{IDRVar}.$$

3.1.2. Self-Similarity Along Time. In the next sections we introduce new linkage change features based on multistep graph similarity measures that in some sense generalize the single-step neighborhood change features of the previous section. We characterize the change of the multistep neighborhood of a node by defining the similarity of a single node *across* snapshots instead of two nodes within a single graph instance. The basic idea is that, for each node, we measure its similarity to itself in two identically labeled graphs representing two consecutive points of time. This enables us to measure the linkage change occurring in the observed time interval using ordinary graph similarity metrics.

First we describe our new contribution, the extension of two graph similarity measures, XJaccard and PSimRank [Fogaras and Rácz 05], to capture temporal change; moreover, we argue why SimRank [Jeh and Widom 02] is inappropriate for constructing temporal features.

SimRank of a pair of nodes u and v is defined recursively as the average similarity of the neighbors of u and v:

$$\begin{aligned} &\text{Sim}_{\ell+1}(u,v) = 0, \text{ if } I(u) \text{ or } I(v) \text{ is empty;} \\ &\text{Sim}_{\ell+1}(u,v) = 1, \text{ if } u = v; \\ &\text{Sim}_{\ell+1}(u,v) = \frac{c}{|I(u)||I(v)|} \sum_{\substack{v' \in I(v) \\ u' \in I(u)}} \text{Sim}_{\ell}(u',v'), \end{aligned} \tag{3.1}$$

where $I(x)$ denotes the set of vertices linking to x, and $c \in (0,1)$ is a decay factor. In order to apply SimRank for similarity of a node v between two snapshots t_0 and t_1, we apply (3.1) so that v' and u' are taken from different snapshots.

Next we describe a known deficiency of SimRank in its original definition, which rules out its applicability for temporal analysis. First, we give the example for the single-graph SimRank. Consider a bipartite graph with k nodes pointing all to another two u and v. In this graph there are no directed paths of length more than one and, hence, the Sim values can be computed in a single iteration. Counterintuitively, we get $\text{Sim}(u,v) = c/k$, i.e., the larger the cocitation of u and v, the smaller their SimRank value. The reason is that the more the number of in-neighbors, the more likely it is that a pair of random neighbors will be different.

Although the example of the misbehavior for SimRank is somewhat artificial in the single-snapshot case, next we show that this phenomenon almost always happens if we consider the similarity of a single node v across two snapshots. If there is no change at all in the neighborhood of node v between the two snapshots, we expect the Sim value to be maximal. However, the situation is identical to the bipartite graph case, and Sim will be inversely proportional to the number of out-links.

3.1.3. Extended Jaccard Similarity Along Time. Our first definition of similarity is based on the extension of the Jaccard coefficient; in a similar way XJaccard is defined in [Fogaras and Rácz 05]. The Jaccard similarity of a page or host v across two snapshots t_0 and t_1 is defined by the overlap of its neighborhood in the two snapshots, $\Gamma^{(t_0)}(v)$ and $\Gamma^{(t_1)}(v)$ as

$$\mathsf{Jac}^{(t_0,t_1)}(v) = \frac{|\Gamma^{(t_0)}(v) \cap \Gamma^{(t_1)}(v)|}{|\Gamma^{(t_0)}(v) \cup \Gamma^{(t_1)}(v)|}.$$

The *extended Jaccard coefficient, XJaccard* for length ℓ of a page or host is defined via the notion of the neighborhood $\Gamma_k^{(t)}(v)$ at distance exactly k as

$$\mathsf{XJac}_{\ell}^{(t_0,t_1)}(v) = \sum_{k=1}^{\ell} \frac{|\Gamma_k^{(t_0)}(v) \cap \Gamma_k^{(t_1)}(v)|}{|\Gamma_k^{(t_0)}(v) \cup \Gamma_k^{(t_1)}(v)|} \cdot c^k(1-c),$$

where c is a decay factor.

The XJac values can be approximated by the min-hash fingerprinting technique for Jaccard coefficients [Broder 97], as described in Algorithm 3 of [Fogaras and Rácz 05]. The fingerprint generation algorithm has to be repeated for each graph snapshot, with the same set of independent random permutations.

We generate temporal features based on the XJac values for four length values $\ell = 1 \ldots 4$. We also repeat the computation on the transposed graph, i.e., replacing out-links $\Gamma^{(t)}(v)$ by in-links $I^{(t)}(v)$. As suggested in [Fogaras and Rácz 05], we set the decay factor $c = 0.1$ as this is the value where, in their experiments, XJaccard yields best average quality for similarity prediction.

Similarly to [Shen et al. 06], we also calculate the mean and variance $\mathsf{XJac}^{(t_0,t_1)}{}_{\ell}(w)$ of the neighbors w for each node v. The following derived features are also calculated:

- similarity at path length $\ell = 2, 3, 4$ divided by similarity at path length $\ell - 1$, and the logarithm of these;
- logarithm of the minimum, maximum, and average of the similarity at path length $\ell = 2, 3, 4$ divided by the similarity at path length $\ell - 1$.

3.1.4. PSimRank Along Time. Next we define similarity over time based on PSimRank, a SimRank variant defined in [Fogaras and Rácz 05] that can be applied similarly to XJaccard in the previous section. As we saw in Section 3.1.2, SimRank is inappropriate for measuring linkage change in time. In the terminology of the previous subsection, the reason is that path fingerprints will be unlikely to meet in a large neighborhood and SimRank values will be low even if there is completely no change in time.

We solve the deficiency of SimRank by allowing the random walks to meet with higher probability when they are close to each other: a pair of random walks at vertices u', v' will advance to the same vertex (i.e., meet in one step) with probability of the Jaccard coefficient $\frac{|I(u')\cap I(v')|}{|I(u')\cup I(v')|}$ of their in-neighborhood $I(u')$ and $I(v')$.

The random walk procedure corresponding to PSimRank along with a fingerprint generation algorithm is defined in [Fogaras and Rácz 05].

For the temporal version, we choose independent random permutations σ_ℓ on the hosts for each step ℓ. In step ℓ if the random walk from vertex u is at u', it will step to the in-neighbor with smallest index given by the permutation σ_ℓ in each graph snapshot.

Temporal features are derived from the PSimRank similarity measure very much the same way as for XJaccard, for four length values $\ell = 1 \dots 4$. We also repeat the computation on the transposed graph, i.e., replacing out-links $\Gamma^{(t)}(v)$ by in-links $I^{(t)}(v)$. As suggested in [Fogaras and Rácz 05], we set the decay factor $c = 0.15$ as this is the value where, in their experiments, PSimRank yields best average quality for similarity prediction. Additionally, we calculate the mean and variance PSimRank(w) of the neighbors w for each node v and derived features as for XJaccard.

3.2. Content and Its Change

The content of webpages can be deployed in content classification either via statistical features such as entropy [Ntoulas 06] or via term weight vectors [Zhou et al. 08, Dai et al. 09]. One of the more complex features that we do not consider in this work is language modeling [Attenberg and Suel 08].

In this section we focus on capturing term-level changes over time. For each target site and crawl snapshot, we collect all the available HTML pages and represent the site as the bag-of-words union of all of their content. We tokenize content using the ICU library,[3] remove stop words,[4] and stem using Porter's method.

[3] http://icu-project.org/

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

We treat the resulting term list as the virtual document for a given site at a point of time. As our vocabulary, we use the most frequent 10,000 terms found in at least 10% and at most 50% of the virtual documents.

To measure the importance of each term i in a virtual document d at time snapshot T, we use the BM25 weighting [Robertson and Walker 94]:

$$t_{i,d}^{(T)} = \mathrm{IDF}_i^{(T)} \cdot \frac{(k_1+1)\mathrm{tf}_{i,d}^{(T)}}{K + \mathrm{tf}_{i,d}^{(T)}},$$

where $\mathrm{tf}_{i,d}^{(T)}$ is the number of occurrences of term i in document d and $\mathrm{IDF}_i^{(T)}$ is the inverse document frequency (Robertson-Spärck Jones weight) for the term at time T. The length of normalized constant K is specified as

$$k_1((1-b) + b \times \mathrm{dl}^{(T)}/\mathrm{avdl}^{(T)}),$$

such that $dl^{(T)}$ and $avdl^{(T)}$ denote the virtual document length and the average length over all virtual documents at time T, respectively. Finally

$$\mathrm{IDF}^{(T)} = log\frac{N - n^{(T)} + 0.5}{n^{(T)} + 0.5},$$

where N denotes the total number of virtual documents and $n^{(T)}$ is the number of virtual documents containing term i. Note that we keep N independent of T and hence if document d does not exist at T, we consider all $\mathrm{tf}_{i,d}^{(T)} = 0$.

By using the term vectors as above, we calculate the temporal content features described in [Dai et al. 09] in the following five groups.

- **Ave:** Average BM25 score of term i over the $T_{\max}$ snapshots:

$$\mathrm{Ave}_{i,d} = \frac{1}{T_{\max}} \cdot \sum_{T=1}^{T_{\max}} t_{i,d}^{(T)}.$$

- **AveDiff:** Mean difference between temporally successive term weight scores:

$$\mathrm{AveDiff}_{i,d} = \frac{1}{T_{\max}-1} \cdot \sum_{T=1}^{T_{\max}-1} |t_{i,d}^{(T+1)} - t_{i,d}^{(T)}|.$$

- **Dev:** Variance of term weight vectors at all time points:

$$\mathrm{Dev}_{i,d} = \frac{1}{T_{\max}-1} \cdot \sum_{T=1}^{T_{\max}} (t_{i,d}^{(T)} - \mathrm{Ave}_{i,d})^2.$$

- **DevDiff:** Variance of term weight vector differences of temporally successive virtual documents:

$$\mathrm{DevDiff}_{i,d} = \frac{1}{T_{\max} - 2} \cdot \sum_{T=1}^{T_{\max}-1} (|t_{i,d}^{(T+1)} - t_{i,d}^{(T)}| - \mathrm{AveDiff}_{i,d})^2.$$

- **Decay:** Weighted sum of temporally successive term weight vectors with exponentially decaying weight. The base of the exponential function, the *decay rate* is denoted by λ. **Decay** is defined as follows:

$$\mathrm{Decay}_{i,d} = \sum_{T=1}^{T_{\max}} \lambda e^{\lambda(T_{\max}-T)} t_{i,d}^{(T)}.$$

4. Classification Framework

For the purposes of our experiments, we computed all the public Web Spam Challenge content and link features of [Castillo et al. 06]. We built a classifier ensemble by splitting features into related sets, and for each we use a collection of classifiers that fit the data type and scale. These classifiers were then combined by ensemble selection. We used the classifier implementations of the machine learning toolkit Weka [Witten and Frank 05].

Ensemble selection is an overproduce-and-choose method allowing the use of large collections of diverse classifiers [Caruana et al. 04]. Its advantages over previously published methods [Caruana et al. 06] include optimization to any performance metric and refinements to prevent overfitting, the latter being unarguably important when more classifiers are available for selection. The motivation for using ensemble selection is that recently this particular ensemble method gained more attention thanks to the winners of KDD Cup 2009 [Niculescu-Mizil et al. 09]. In our experiments [Erdélyi et al. 11] ensemble selection used for web spam detection performed significantly better than other classifier combination methods in the literature, such as log-odds based averaging [Lynam et al. 06] and bagging.

In the context of combining classifiers for web classification, to our best knowledge, ensemble selection has not been applied yet. Previously, only simple methods that combine the predictions of SVM or decision tree classifiers through logistic regression or random forest have been used [Cormack 07]. We believe that the ability to combine a large number of classifiers while preventing overfitting makes ensemble selection an ideal candidate for web classification, because it allows us to use a large number of features and learn different aspects of the training data at the same time. Instead of tuning various parameters of different classifiers, we can concentrate on finding powerful features and selecting the main

classifier models that we believe to be able to capture the differences between the classes to be distinguished.

We used the ensemble selection implementation of Weka [Witten and Frank 05] for performing the experiments. Weka's implementation supports the proven strategies to avoid overfitting such as model bagging, sort initialization, and selection with replacement. We allow Weka to use all available models in the library for greedy sort initialization, and use 5-fold embedded cross-validation during ensemble training and building. We set AUC as the target metric to optimize for and run 100 iterations of the hill-climbing algorithm.

We mention that we have to be careful with treating missing feature values. Since the temporal features are based on at least two snapshots, for a site that appears only in the last one, all temporal features have missing value. For classifiers that are unable to treat missing values, we define default values depending on the type of the feature.

4.1. Learning Methods

We use the following models in our ensemble: bagged and boosted decision trees, logistic regression, naive Bayes, and variants of random forests. For most classes of features, we use all classifiers and let selection choose the best ones. The exception is static and temporal term vector-based features for which, because of the very large number of features, we may use only random forest and SVM. We train our models as follows.

Bagged LogitBoost: we do 10 iterations of bagging and vary the number of iterations from 2 to 64 in multiples of two for LogitBoost.

Decision Trees: we generate J48 decision trees by varying the splitting criterion and pruning options and use either Laplacian smoothing or no smoothing at all.

Bagged Cost-Sensitive Decision Trees: we generate J48 decision trees with default parameters but vary the cost sensitivity for false positives in steps of 10 from 10 to 300. We do the same number of iterations of bagging as for LogitBoost models.

Logistic Regression: we use a regularized model varying the ridge parameter between 10^{-8} to 10^{4} by factors of 10. We normalize features to have mean 0 and standard deviation 1.

Random Forests: we use FastRandomForest[5] instead of the native Weka implementation for faster computation. The forests have 250 trees and, as suggested

[5] http://code.google.com/p/fast-random-forest/.

in [Breiman 01], the number of features considered at each split is $s/2$, s, $2s$, $4s$, and $8s$, where s is the square root of the total number of features available.

Naive Bayes: we allow Weka to model continuous features either as a single normal or with kernel estimation, or we let it discretize them with supervised discretization.

5. Results and Discussion

In this section we describe the various ensembles we built and measure their performance.[6] We compare feature sets by using the same learning methods described in Section 4 although varying the subset of features available for each of the classifier instances when training and combining these classifiers using ensemble selection. We also concentrate on the value of temporal information for web spam detection. As our goal is to explore the computational cost vs. classification performance trade-off, we will describe the resource needs for various features in detail in Section 6.

For training and testing, we use the official Web Spam Challenge 2008 training and test sets [Castillo et al. 06]. As can be seen in Table 1, these show considerable class imbalance, which makes the classification problem more difficults. For DC2010, we also use the official training set as described in Table 5. For ClueWeb09 we used a 50–50% random split.

To make it easy to compare our results to previous results, we cite the Web Spam Challenge 2008 and Discovery Challenge 2010 winners performances in the summary tables next. For ClueWeb09, the only previous evaluation is in terms of TREC retrieval performance [Cormack et al. 11], which we cannot directly compare here.

5.1. Content-Only Ensemble

We build three different ensembles over the content-only features in order to assess performance by completely eliminating linkage information. The feature sets available for these ensembles are the following:

- (A) Public content [Ntoulas et al. 06, Castillo et al. 07] features without any link-based information. Features for the page with maximum PageRank in the host are not used to save the PageRank computation. Corpus precision,

[6] The exact classifier model specification files used for Weka and the data files used for the experiments are available upon request from the authors.

Feature Set	Number of Features	UK2007	DC2010	Clue Web09
Content (A)	74	0.859	0.757	0.829
Content (Aa)	24	0.841	0.726	0.635
Content (B)	96	0.879	0.799	0.827
BM25 + (B)	10096	**0.893**	**0.891**	**0.870**
Challenge best	-	0.852	0.830	-

Table 6. AUC value of spam ensembles built from content-based features.

the fraction of words in a page that is corpus-wise frequent, and corpus recall, the fraction of corpus-wise frequent terms in the page are not used either because they require global information from the corpus.

- (Aa) The tiniest feature set of 24 features from (A): query precision and query recall defined similar to corpus precision and recall but based on popular terms from a proprietary query log[7] instead of the entire corpus. A very strong feature set based on the intuition that spammers use terms that make up popular queries.
- (B) The full public content feature set [Castillo et al. 07], including features for the maximum PageRank page of the host.
- Feature set (B) plus a bag-of-words representation derived from the BM25 [Robertson and Walker 94] term weighting scheme.

Table 6 presents the performance comparison of ensembles built using either of the above feature sets. The DC2010 and ClueWeb09 detailed results are in Table 8 and Table 9, respectively. Performance is given in AUC for all datasets.

Surprisingly, with the small (Aa) feature set of only 24 features a performance was achieved that was only 1% worse than that of the Web Spam Challenge 2008 winner, who employed more sophisticated methods to get his/her result. By using all the available content-based features without linkage information, we get roughly the same performance as the best that have been reported on our dataset so far. However, this achievement can be attributed rather to the better machine learning techniques used than the feature set itself since the features

[7] A summary is available as part of our data release at `https://dms.sztaki.hu/sites/dms.sztaki.hu/files/download/2013/enpt-queries.txt.gz`.

Feature Set	Number of Features	UK2007	DC2010	ClueWeb09
Public link-based [7]	177	0.759	0.587	0.806
All combined	10 273	**0.902**	0.885	0.876

Table 7. Performance of ensembles built on link-based and all features.

used for this particular measurement were already publicly accessible at the time of the Web Spam Challenge 2008.

As can be seen in Table 6, relative performance of content-based features over different corpora varies a lot. In the cases of DC2010 and ClueWeb09, the small (Aa) feature set achieves much worse results than the largest feature set having best performance of all datasets. The fact that the content (A, Aa, B) and link (Table 7) performances are always better for UK2007 might be explained by the fact that the UK2007 training and testing sets were produced by random sampling without considering domain boundaries. Hence, in a large domain with many subdomains, part of the hosts belong to the training and part to the testing set with very similar distribution. This advantage disappears for the BM25 features.

5.2. Full Ensemble

Results of the ensemble incorporating all the previous classifiers is seen in Table 7. The DC2010 detailed results are in Table 8. Overall, we observe that BM25 is a very strong feature set that could even be used itself for a lightweight classifier. However, link features add little to quality and the gains apparently diminish for DC2010, likely due to the fact that the same domain and IP address is not split between training and testing.

The best Web Spam Challenge 2008 participant [Geng et al. 08] reaches an AUC of 0.85, whereas for DC2010, the best spam classification AUC of [Nikulin 10] is 0.83. We outperform these results by a large margin.

For DC2010 we also show detailed performance for nine attributes in Table 8, averaged in three groups: spam, genre, and quality (as in Table 5). Findings are similar: with BM25 domination, part or all of the content features slightly increase the performance. Results for the quality attributes and, in particular for trust, are very low. Classification for these aspects remains a challenging task for the future.

Feature Set	spam	genre average	quality average	average
Public link-based [7]	0.655	0.614	0.519	0.587
Content (A)	0.757	0.713	0.540	0.660
Content (Aa)	0.726	0.662	0.558	0.634
Content (B)	0.799	0.735	0.512	0.668
BM25	**0.876**	**0.805**	**0.584**	**0.739**
Public link-based + (B)	0.812	0.731	0.518	0.669
BM25 + (A)	0.872	**0.816**	0.580	**0.754**
BM25 + (B)	**0.891**	0.810	**0.612**	0.744
All combined	0.885	0.813	0.553	0.734

Table 8. Performance over the DC2010 labels in terms of AUC.

For ClueWeb09, detailed performance for selected ODP categories can be seen in Table 9. Identically to DC2010 results, BM25 features provide the best classification performance. However, combinations with other feature sets yield gains for only spam classification. For the ODP classification tasks, linkage information does not help in general: the content-based feature set has roughly the same performance with or without page-level linkage information, and combining with the link-based feature set does not improve performance notably in most labeling tasks.

5.3. Temporal Link Ensembles

First, we compare the temporal link features proposed in Section 3.1 with those published earlier [Shen et al. 06]. Then, we build ensembles that combine the temporal with the public link-based features described by [Becchetti et al. 06]. The results are summarized in Table 10. Note that all experiments in this section and in Section 5.4 were carried out on the WEBSPAM-UK2007 dataset.

As these measurements show, our proposed graph-similarity-based features successfully extend the growth- and death-rate-based ones by achieving higher accuracy, improving AUC by 1.3%. However, by adding temporal to static link-based features we get only marginally better ensemble performance.

Feature Set	spam	Arts	Business	Computers	Recreation	Science	Society	Sports	**ODP average**
Link [7]	.806	.569	.593	.591	.532	.624	.540	.504	.595
Content (A)	.829	.676	.726	.632	.669	.720	.639	.673	.695
Content (Aa)	.635	.508	.524	.554	.487	.558	.502	.522	.536
Content (B)	.827	.673	.727	.634	.670	.720	.629	.674	.694
BM25	.845	**.913**	**.890**	**.931**	**.907**	**.883**	**.915**	**.959**	**.914**
Link + (B)	.848	.675	.731	.646	.669	.727	.631	.669	.699
BM25 + (A)	.871	.895	.881	.896	.879	.851	.904	.935	.892
BM25 + (B)	.869	.895	.881	.898	.892	.850	.906	.934	.894
All combined	**.876**	.896	.883	.898	.892	.852	.905	.936	.895

Table 9. Performance over the ClueWeb09 labels in terms of AUC.

Section	Feature Set	No. of Features	AUC
3.1.1	Growth/death rates	29	0.617
3.1.3-4	XJaccard + PSimRank	63	0.625
	Public link-based [7]	176	0.765
3.1.1	Public + growth/death rates	205	0.758
3.1.3-4	Public + XJaccard + PSimRank	239	**0.769**
	All link-based	268	0.765
	WSC 2008 Winner	—	0.852

Table 10. Performance of ensembles built on link-based features.

To rank the link-based feature sets by their contribution in the ensemble, we build classifier models on the three separate feature subsets (public link-based, growth/death-rate-based and graph-similarity-based features, respectively) and let ensemble selection combine them. This restricted combination results in a slightly worse AUC of 0.762. By calculating the total weight contribution, we get the following ranked list (weight contribution showed in parenthesis): public link-based (60.8%), graph-similarity-based (21.5%), growth/death-rate-based (17.7%). This ranking also supports the findings presented in Table 10 that graph similarity based temporal link-based features should be combined with public link-based features if temporal link-based features are used.

To separate the effect of ensemble selection on the performance of temporal link-based feature sets, we repeat the experiments with bagged cost-sensitive decision trees only, a model reported to be effective for web spam classification [Ntoulas et al. 06]. The results for these experiments are shown in Table 11.

As can be seen in Table 11, when using bagged cost-sensitive decision trees, our proposed temporal link-based similarity features achieve 3.5% better performance than the growth/death-rate-based features published earlier.

When comparing results in Table 11 and in Table 10, we can see that ensemble selection (i) significantly improves accuracy (as expected) and (ii) diminishes the performance advantage achievable by the proposed temporal link-based features over the previously published ones.

Section	Feature Set	No. of Features	AUC
3.1.1	Growth/death rates	29	0.605
3.1.3	XJaccard	42	0.626
3.1.4	PSimRank	21	0.593
3.1.3-4	XJaccard + PSimRank	63	0.610
	Public link-based [7]	176	**0.731**
3.1.1	Public + growth/death rates	205	0.696
3.1.3-4	Public + XJaccard + PSimRank	239	*0.710*
	All link-based	268	0.707
	WSC 2008 Winner	—	0.852

Table 11. Performance of bagged cost-sensitive decision trees trained on link-based features.

As evident from Table 11, the proposed PSimRank-based temporal features perform roughly the same as the growth- and death-rate-based ones whereas the XJaccard-based temporal features perform slightly better.

Next we perform sensitivity analysis of the temporal link-based features by using bagged cost-sensitive decision trees. We build 10 different random training samples for each of the possible fractions 10%, 20%, ..., 100% of all available labels. In Figure 2 we can see that the growth/death-rate-based features as well as the PSimRank-based features are not sensitive to training set size although the XJaccard-based ones are. That is, even though XJaccard is better in terms of performance than the other two feature sets considered, it is more sensitive to the amount of training data used as well.

5.4. Temporal Content Ensembles

We build ensembles based on the temporal content features described in Section 3.2 and their combination themselves, with the static BM25 features, and with the content-based features of [Ntoulas et al. 06]. The performance comparison of temporal content-based ensembles is presented in Table 12.

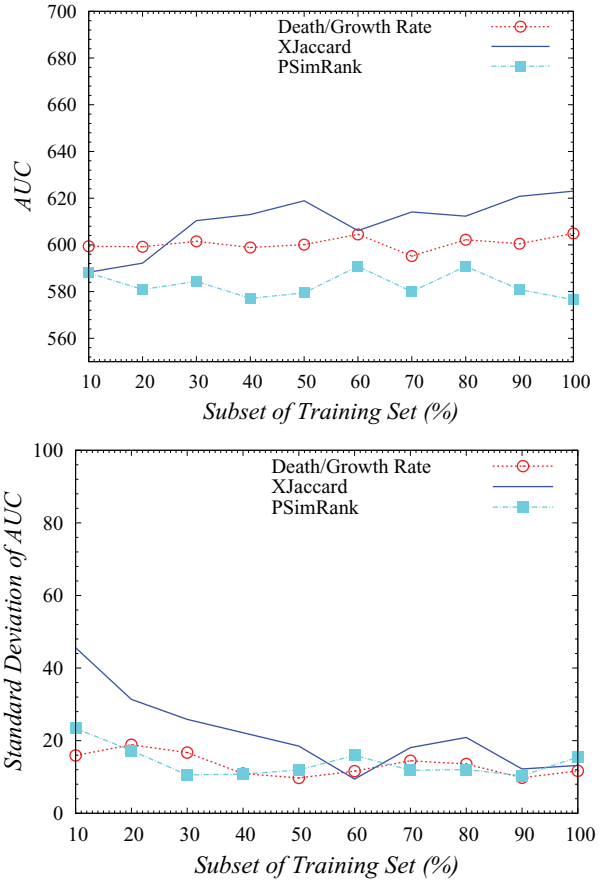

Figure 2. Sensitivity of temporal link-based features. (a) AUC values averaged across 10 measurements. (b) standard deviations of AUC for different training set sizes.

By combining all the content- and link-based features, both temporal and static ones, we train an ensemble that incorporates all the previous classifiers. This combination resulted in an AUC of 0.908, meaning no significant improvement can be achieved with link-based features over the content-based ensemble.

6. Computational Resources

For the experiments, we used a 45-node Hadoop cluster of dual core machines with 4GB RAM each as well as multicore machines with over 40GB RAM. Over

Feature Set	No. of Features	AUC
Static BM25	10,000	0.736
Ave	10,000	0.749
AveDiff	10,000	0.737
Dev	10,000	0.767
DevDiff	10,000	0.752
Decay	10,000	0.709
Temporal combined	50,000	0.782
Temporal combined + BM25	60,000	**0.789**
Public content-based [Ntoulas et al. 06] + temporal	50,096	0.901
All combined	60,096	**0.902**

Table 12. Performance of ensembles built on temporal content-based features.

this architecture we were able to compute all features, some of which would require excessive resources either when used by a smaller archive or if the collection is larger or if fast classification is required for newly discovered sites during crawl time. Some of the most resource-bound features involve the multistep neighborhood in the page-level graph that already requires approximation techniques for WEBSPAM-UK2007 [Castillo et al. 07].

We describe the computational requirements of the features by distinguishing update and batch processing. For batch processing, an entire collection is analyzed at once, a procedure that is probably performed only for reasons of research. Update is probably the typical operation for a search engine. For an Internet Archive, update is also advantageous as long as it allows fast reaction to sample, classify, and block spam from a yet unknown site.

6.1. Batch Processing

The first expensive step involves parsing to create terms and links. The time requirement scales linearly with the number of pages. Since apparently a few hundred page sample of each host suffices for feature generation, the running time is also linear in the number of hosts. For a very large collection such as ClueWeb09, distributed processing may be necessary. Over 45 dual core

Feature Set	Step	Hours	Configuration
Content (A) + BM25	Parsing	36	45 dual core Pentium-D 3.0GHz machines, 4GB RAM, Hadoop 0.21
	Feature generation	36	
	Selection of labeled pages	3	
Link	PageRank	10	5 eight-core Xeon 1.6GHz machines, 40+GB RAM
	Neighborhood	4	
	Local features	1	

Table 13. Processing times and cluster configurations for feature sets over ClueWeb09.

Pentium-D 3.0GHz machines running Hadoop 0.21, we parsed the uncompressed 9.5TB English part of ClueWeb09 in 36 hours. Additional tasks such as term counting, BM25 or content feature generation fits within the same time frame. If features are generated for only a small labeled part of the data, it took us 3 hours to select the appropriate documents, and additional processing time was negligible. Processing times are summarized in Table 13.

Host level aggregation allows us to proceed with a much smaller size data. However for aggregation we need to store a large number of partial feature values for all hosts unless we sort the entire collection by host, again by external memory or Map-Reduce sort.

After aggregation, host-level features are inexpensive to compute. The following features, however, remain expensive:

- Page level PageRank. Note that this is required for all content features involving the maximum PageRank page of the host.
- Page level features involving multistep neighborhood such as neighborhood size at distance k, as well as graph similarity.

In order to be able to process graphs of ClueWeb09 scale (4.7 billion nodes and 17 billion edges), we implemented message-passing C++ codes. Over a total of 30 cores of six Xeon 1.6GHz machines, each with at least 40GB RAM, one PageRank and one Bit Propagation iteration both took approximately one hour while all other, local features completed within one hour.

Training the classifier for a few 100,000 sites can be completed within a day on a single CPU on a commodity machine with 4-16GB RAM; here, costs strongly

Configuration	Number of Hosts	Feature Sets	Example	Expected Accuracy	Computation
Small 1-2 machines	10,000	Content (A) BM25	subset of UK2007	0.80-0.87	Non- distributed
Medium 3-10 machines	100,000	Content (A) BM25, link	DC2010	0.87-0.90	MapReduce and Disk-based e.g. GraphChi
Large 10+ machines	1,000,000	Content (B) BM25, link	ClueWeb09	0.9+	MapReduce and Pregel

Table 14. Sample configurations for Web spam filtering in practice.

depend on the classifier implementation. Our entire classifier ensemble for the labeled WEBSPAM-UK2007 hosts took a few hours to train.

6.2. Incremental Processing

As preprocessing and host-level aggregation is linear in the number of hosts, this reduces to a small job for an update. This is especially true if we are able to split the update by sets of hosts; in this case we may even trivially parallelize the procedure.

The only nontrivial content-based information is related to document frequencies: both the inverse document frequency term of BM25 [Robertson and Walker 94] and the corpus precision and recall dictionaries may in theory be fully updated when new data is added. We may, however, approximate by the existing values under the assumption that a small update batch will not affect these values greatly. From time to time, however, all features beyond (Aa) need a global recomputation step.

The link structure is, however, nontrivial to update. Although incremental algorithms exist to create the graph and to update PageRank-type features [Desikan et al. 05, 06, Kohlschütter et al. 06], these algorithms are rather complex and their resource requirements are definitely beyond the scale of a small incremental data.

Incremental processing may have the assumption that no new labels are given, since labeling a few thousand hosts takes time comparable to batch processing hundreds of thousands of them. Given the trained classifier, a new site can be classified in seconds right after its feature set is computed.

7. Conclusions

With the illustration over the 100,000 host WEBSPAM-UK2007, the half-billion page ClueWeb09, and the 190,000 host DC2010 datasets, we have investigated the trade-off between feature generation and spam classification accuracy. We observe that more features achieve better performance, however, when combining them with the public link-based feature set we get only marginal performance gain. By using the WEBSPAM-UK2007 data along with seven previous monthly snapshots of the `.uk` domain, we have presented a survey of temporal features for web spam classification. We investigated the performance of link, content, and

temporal[8] web spam features with ensemble selection. As practical message, we may conclude that, as seen in Table 14, single machines may compute content and BM25 features for a few 10,000 hosts only. Link features need additional resources and either compressed, disk-based or, in the largest configuration, Pregel-like distributed infrastructures.

We proposed graph-similarity-based temporal features, which aim to capture the nature of linkage change of the neighborhoods of hosts. We have shown how to compute these features efficiently on large graphs using a Monte Carlo method. Our features achieve better performance than previously published methods, however, when combining them with the public link-based feature set we get only marginal performance gain.

By our experiments, it has turned out that the appropriate choice of the machine learning techniques is probably more important than devising new complex features. We have managed to compile a minimal feature set that can be computed incrementally very quickly to allow interceptance of spam at crawl time based on a sample of a new web site. Sample configurations for web spam filtering are summarized in Table 14.

Our results open the possibility for spam filtering practice in Internet archives that are mainly concerned about their resource waste and would require fast reacting filters. BM25-based models are suitable even for filtering at crawl time.

Some technologies remain open to be explored. For example, unlike expected, the ECML/PKDD Discovery Challenge 2010 participants did not deploy cross-lingual technologies for handling languages other than English. Some ideas worth exploring include the use of dictionaries to transfer a bag-of-words-based model and the normalization of content features across languages to strengthen the language independence of the content features. The natural language-processing-based features were not used either, which might help, in particular, with the challenging quality attributes.

Acknowledgments. To Sebastiano Vigna, Paolo Boldi, and Massimo Santini for providing us with the UbiCrawler crawls [Boldi et al. 04, 08]. In addition to them, also to Ilaria Bordino, Carlos Castillo and Debora Donato for discussions on the WEBSPAM-UK datasets [Bordino et al. 08].

To the large team of organizers and assessors for the complex labeling process of the DC2010 dataset.

This paper is a comprehensive comparison of the best performing classification techniques based on [Benczúr et al. 09, Erdélyi et al. 09, 11, Erdélyi and Benczúr 11] and new experiments.

[8] The temporal feature data used in our research is available at: `https://datamining.sztaki.hu/en/download/web-spam-resources`

Funding. This work was supported in part by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No 288956), by the "Momentum - Big Data" grant of the Hungarian Academy of Sciences, OTKA NK 105645 and by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013). The research was carried out as part of the EITKIC_12-1-2012-0001 project, which is supported by the Hungarian Government, managed by the National Development Agency, financed by the Research and Technology Innovation Fund, and was performed in cooperation with the EIT ICT Labs Budapest Associate Partner Group.

References

[Abernethy et al. 08] J. Abernethy, O. Chapelle, and C. Castillo. "WITCH: A New Approach to Web Spam Detection." Paper presented at the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb), Beijing, April 22, 2008.

[Attenberg and Suel 08] J. Attenberg and T. Suel. "Cleaning Search Results Using Term Distance Features." In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, pp. 21–24, New York, NY, USA: ACM 2008.

[Bar-Yossef et al. 04] Z. Bar-Yossef, A. Z. Broder, R. Kumar, and A. Tomkins. "Sic transit gloria telae: Towards an Understanding of the Web's Decay." In *Proceedings of the 13th World Wide Web Conference (WWW)*, pp. 328–337. New York, NY: ACM Press, 2004.

[Bar-Yossef et al. 09] Z. Bar-Yossef, I. Keidar, and U. Schonfeld. "Do Not Crawl in the Dust: different URLS with Similar Text." *ACM Transactions on the Web (TWEB)* 3:1 (2009), 1–31.

[Barton 12] S. Barton. Mignify, a big data refinery built on HBASE. In *HBASE CON*, San Francisco, May 22, 2012.

[Becchetti 06] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. "Link-Based Characterization and Detection of Web Spam." paper presented at the 2nd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb), Seattle, WA, August 10, 2006.

[Benczúr et al. 08] A. A. Benczúr, D. Siklósi, J. Szabó, I. Bíró, Z. Fekete, M. Kurucz, A. Pereszlényi, S. Rácz, and A. Szabó. "Web Spam: A Survey with Vision for the Archivist." *International Web Archiving Workshop*, Aarhus, Denmark, September 18–19, 2008.

[Benczúr et al. 09] A. A. Benczúr, M. Erdélyi, J. Masanés, and D. Siklósi. "Web Spam Challenge Proposal for Filtering in Archives." In *AIRWeb '09: Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*. New York, NY: ACM Press, 2009.

[Boldi et al. 04] P. Boldi, B. Codenotti, M. Santini, and S. Vigna. "Ubicrawler: A Scalable Fully Distributed Web Crawler." *Software: Practice & Experience* 34:8 (2004), 721–726.

[Boldi et al. 08] P. Boldi, M. Santini, and S. Vigna. "A Large Time Aware Web Graph." *SIGIR Forum* 42:2 (2008), 33–38.

[Bordino et al. 08] I. Bordino, P. Boldi, D. Donato, M. Santini, and S. Vigna. "Temporal Evolution of the UK Web." Paper presented at the *Workshop on Analysis of Dynamic Networks (ICDM-ADN'08)*, Sparks, NV, May 2, 2008.

[Bordino et al. 10] I. Bordino, D. Donato, and R. Baeza-Yates. "Coniunge et impera: Multiple-Graph Mining for Query-Log Analysis." In *Machine Learning and Knowledge Discovery in Databases*, pp. 168–183. Berlin: Springer, 2010.

[Breiman 01] L. Breiman. "Random Forests." *Machine Learning*, 45:1 (2001), 5–32.

[Broder 97] A. Z. Broder. "On the Resemblance and Containment of Documents." In *Proceedings of the Compression and Complexity of Sequences (SEQUENCES'97)*, pp. 21–29, Salerno, Italy, June 11–13, 1997.

[Caruana et al. 06] R. Caruana, A. Munson, and A. Niculescu-Mizil. "Getting the Most Out of Ensemble Selection." In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pp. 828–833, Washington, DC, USA: IEEE Computer Society, 2006.

[Caruana et al. 04] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. "Ensemble Selection from Libraries of Models." In *ICML '04: Proceedings of the 21st International Conference on Machine Learning*, pp. 18, New York, NY, USA: ACM, 2004.

[Castillo et al. 08] C. Castillo, K. Chellapilla, and L. Denoyer. Web Spam Challenge 2008. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Beijing, April 22, 2008.

[Castillo and Davison 11] C. Castillo and B. Davison. *Adversarial Web Search*, Vol. 4. Hanover, MA: Now Publishers Inc, 2011.

[Castillo et al. 06] C. Castillo, D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini, and S. Vigna. "A reference collection for web spam." *SIGIR Forum* 40:2 (2006), 11–24.

[Castillo et al. 07] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. "Know Your Neighbors: Web Spam Detection Using the Web Topology." *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 423–430, Amsterdam, July 23–27, 2007.

[Chawla et al. 04] N. Chawla, N. Japkowicz, and A. Kotcz. "Editorial: Special Issue on Learning from imbalanced Data Sets." *ACM SIGKDD Explorations Newsletter* 6:1 (2004), 1–6.

[Chekuri et al. 97] C. Chekuri, M. H. Goldwasser, P. Raghavan, and E. Upfal. Web Search Using Automatic Classification. *Paper presented at the 6th International World Wide Web Conference (WWW)*, San Jose, USA, 1997.

[Cho and Garcia-Molina 00a] J. Cho and H. Garcia-Molina. "The Evolution of the Web and Implications for an Incremental Crawler." In *The VLDB Journal*, pp. 200–209, San Francisco, CA: Morgan Kaufman, 2000.

[Cho and Garcia-Molina 00b] J. Cho and H. Garcia-Molina. "Synchronizing a database to Improve Freshness." In *Proceedings of the International Conference on Management of Data*, pp. 117–128. New York, NY: ACM, 2000.

[Chung et al. 09] Y. Joo Chung, M. Toyoda, and M. Kitsuregawa. "A Study of Web Spam Evolution Using a Time Series of Web Snapshots." In *AIRWeb '09: Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*. New York, NY: ACM Press, 2009.

[Convey 96] E. Convey. "Porn Sneaks Way Back on Web." *The Boston Herald*, May 1996.

[Cormack 07] G. Cormack. "Content-Based Web Spam Detection." In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, New York, NY: ACM, 2007.

[Cormack et al. 11] G. Cormack, M. Smucker, and C. Clarke. "Efficient and Effective Spam Filtering and Re-ranking for Large Web Datasets." *Information Retrieval*, 14:5 (2011), 441–465.

[Csalogány et al. 07] K. Csalogány, A. Benczúr, D. Siklósi, and L. Lukács. "Semi-Supervised Learning: A Comparative Study for Web Spam and Telephone User Churn." In *Graph Labeling Workshop in Conjunction with ECML/PKDD 2007*, Warsaw, September 17, 2007.

[Dai et al. 09] N. Dai, B. D. Davison, and X. Qi. "Looking into the Past to Better Classify Web Spam." In *AIRWeb '09: Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*. New York, NY: ACM Press, 2009.

[Desikan et al. 05] P. Desikan, N. Pathak, J. Srivastava, and V. Kumar. "Incremental Page Rank Computation on Evolving Graphs." In *WWW '05: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, pp. 1094–1095, New York, NY, USA: ACM, 2005.

[Desikan et al. 06] P. K. Desikan, N. Pathak, J. Srivastava, and V. Kumar. Divide and Conquer Approach for Efficient Pagerank Computation. In *ICWE '06: Proceedings of the 6th International Conference on Web Engineering*, pp. 233–240. New York, NY, USA: ACM, 2006.

[Dong et al. 10] A. Dong, Y. Chang, Z. Zheng, G. Mishne, J. Bai, K. Buchner, R. Zhang, C. Liao, and F. Diaz. "Towards Recency Ranking in Web Search." In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, New York, February 3–6, 2010.

[Eiron et al. 04] N. Eiron, K. S. McCurley, and J. A. Tomlin. "Ranking the Web Frontier." In *Proceedings of the 13th International World Wide Web Conference (WWW)*, pp. 309–318. New York, NY, USA: ACM Press, 2004.

[Erdélyi and Benczúr 11] M. Erdélyi and A. A. Benczúr. "Temporal Analysis for Web Spam Detection: An Overview." In *1st International Temporal Web Analytics Workshop (TWAW) in conjunction with the 20th International World Wide Web Conference in Hyderabad, India.* CEUR Workshop Proceedings, Hyderabad, India, March 28, 2011.

[Erdélyi et al. 09] M. Erdélyi, A. A. Benczúr, J. Masanés, and D. Siklósi. "Web Spam Filtering in Internet Archives." In *AIRWeb '09: Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*. New York, NY: ACM Press, 2009.

[Erdélyi et al. 11] M. Erdélyi, A. Garzó, and A. A. Benczúr. "Web Spam Classification: A Few Features Worth More." In *Joint WICOW/AIRWeb Workshop on Web Quality (WebQuality 2011) In conjunction with the 20th International World Wide Web Conference in Hyderabad, India.* ACM Press, 2011.

[Fetterly and Gyöngyi 09] D. Fetterly and Z. Gyöngyi. editors. Fifth International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2009), Madrid, April 21, 2009.

[Fogaras and Rácz 05] D. Fogaras and B. Rácz. "Scaling Link-Based Similarity Search." Paper presented at the 14th World Wide Web Conference (WWW), pp. 641–650. Chiba, Japan, 2005.

[Fogarty et al. 05] J. Fogarty, R. S. Baker, and S. E. Hudson. "Case Studies in the Use of Roc Curve Analysis for Sensor-Based Estimates in Human Computer Interaction." In *Proceedings of Graphics Interface 2005*, GI '05, pp. 129–136, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society, 2005.

[Friedman et al. 00] J. Friedman, T. Hastie, and R. Tibshirani. "Additive Logistic Regression: A Statistical View of Boosting." *Annals of Statistics* 28:2 (2000), 337–374.

[Geng et al. 08] G. Geng, X. Jin, and C. Wang. "CASIA at WSC2008." In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*. New York, NY: ACM, 2008.

[Geng et al. 10] X.-C. Z. Guang-Gang Geng, Xiao-Bo Jin, and D. Zhang. Evaluating web content quality via multi-scale features. In *Proceedings of the ECML/PKDD 2010 Discovery Challenge*, Barcelona, September 20, 2010.

[Gyöngyi and Garcia-Molina 05a] Z. Gyöngyi and H. Garcia-Molina. "Spam: It's Not Just for Inboxes Anymore." *IEEE Computer Magazine* 38:10 (2005), 28–34.

[Gyöngyi and Garcia-Molina 05b] Z. Gyöngyi and H. Garcia-Molina. "Web Spam Taxonomy." In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Chiba, Japan, 2005.

[Gyöngyi et al. 04] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. "Combating Web Spam With TrustRank." *Paper presented at the 30th International Conference on Very Large Data Bases (VLDB)*, pp. 576–587, Toronto, Canada, 2004.

[Henzinger et al. 02] M. R. Henzinger, R. Motwani, and C. Silverstein. "Challenges in Web Search Engines." *SIGIR Forum* 36:2 (2002), 11–22.

[Hotho et al. 08] A. Hotho, D. Benz, R. Jäschke, and B. Krause, editors. *Proceedings of the ECML/PKDD Discovery Challenge*, Antwerp, Belgium, September 15, 2008.

[Jeh and Widom 02] G. Jeh and J. Widom. SimRank: A Measure of Structural-Context Similarity. In *Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 538–543. New York, NY: ACM, 2002.

[Kohlschütter et al. 06] C. Kohlschütter, P. A. Chirita, and W. Nejdl. "Efficient Parallel Computation of PageRank." In *Advances in Information Retrieval*, pp. 241–252, Lecture Notes in Computer Science, Volume 3936. Berlin Heidelberg: Springer, 2006.

[Kou and Cohen 07] Z. Kou and W. W. Cohen. Stacked Graphical Models for Efficient Inference in Markov Random Fields. In *SDM 07*, Minneapolis, MN, April 26–28, 2007.

[Lin et al. 07] Y. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. Tseng. "Splog Detection Using Content, Time and Link Structures." In *2007 IEEE International Conference on Multimedia and Expo*, pp. 2030–2033. IEEE, 2007.

[Lynam et al. 06] T. Lynam, G. Cormack, and D. Cheriton. "On-Line Spam Filter Fusion." *Proceedings of the 29th international ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 123–130. New York, NY: ACM, 2006.

[Niculescu-Mizil et al. 09] A. Niculescu-Mizil, C. Perlich, G. Swirszcz, V. Sindhwani, Y. Liu, P. Melville, D. Wang, J. Xiao, J. Hu, M. Singh, et al. "Winning the KDD Cup Orange Challenge with Ensemble Selection." In *KDD Cup and Workshop in Conjunction with KDD 2009*, Paris, June 28, 2009.

[Nikulin 10] V. Nikulin. "Web-Mining with Wilcoxon-Based Feature Selection, Ensembling and Multiple Binary Classifiers." In *Proceedings of the ECML/PKDD 2010 Discovery Challenge*, Barcelona, September 20, 2010.

[Ntoulas et al. 06] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. "Detecting Spam Web Pages Through Content Analysis." In *Proceedings of the 15th International World Wide Web Conference (WWW)*, pp. 83–92. Edinburgh, Scotland, 2006.

[Robertson and Walker 94] S. E. Robertson and S. Walker. "Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval." In *In Proceedings of SIGIR'94*, pp. 232–241. Berlin: Springer-Verlag, 1994.

[Shen et al. 06] G. Shen, B. Gao, T. Liu, G. Feng, S. Song, and H. Li. "Detecting Link Spam Using Temporal Information." In *Proceedings of the IEEE International Conference on Data Mining (ICDM)* Hong Kong, December 18–22, pp. 1049–1053, 2006.

[Siklósi et al. 12] D. Siklósi, B. Daróczy, and A. Benczúr. "Content-Based Trust and Bias Classification via Biclustering." In *Proceedings of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality*, pp. 41–47. New York, NY: ACM, 2012.

[Singhal 04] A. Singhal. "Challenges in Running a Commercial Search Engine." In *IBM Search and Collaboration Seminar 2004*. IBM Haifa Labs, 2004.

[Sokolov et al. 10] L. D. A. Sokolov, T. Urvoy, and O. Ricard. Madspam consortium at the ECML/PKDD Discovery Challenge 2010. Paper presented at the ECML/PKDD 2010 Discovery Challenge, Barcelona, September 20, 2010.

[Webb et al. 08] S. Webb, J. Caverlee, and C. Pu. "Predicting Web Spam With HTTP Session Information." In *Proceeding of the 17th ACM Conference on Information and Knowledge Management*, pp. 339–348. New York, NY: ACM, 2008.

[Witten and Frank 05] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, Second edition, June 2005.

[Wu et al. 06] B. Wu, V. Goel, and B. D. Davison. "Topical TrustRank: Using Topicality to Combat Web Spam." Paper presented at the 15th International World Wide Web Conference (WWW), Edinburgh, Scotland, 2006.

[Zhou et al. 08] B. Zhou, J. Pei, and Z. Tang. "A Spamicity Approach to Web Spam Detection." In *Proceedings of the 2008 SIAM International Conference on Data Mining (SDM'08)*, pp. 277–288, Atlanta, April 24–26, 2008.

Miklós Erdélyi, Institute for Computer Science and Control, Hungarian Academy of Sciences, 13-17 Kende u, Budapest, Hungary, H-1111; and Department of Computer Science and Systems Technology, University of Pannonia, Veszprém 10 Egyetem u, Veszprém, Hungary, H-8200 (miklos@ilab.sztaki.hu)

András A. Benczúr, Institute for Computer Science and Control, Hungarian Academy of Sciences, 13-17 Kende u, Budapest, Hungary, H-1111; and Eötvös University Budapest, 1-3 Egyetem tér, Budapest, Hungary H-1053 (benczur@ilab.sztaki.hu)

Bálint Daróczy, Institute for Computer Science and Control, Hungarian Academy of Sciences, 13-17 Kende u, Budapest, Hungary, H-1111; and Eötvös University Budapest, 1-3 Egyetem tér, Budapest, Hungary H-1053 (daroczyb@ilab.sztaki.hu)

András Garzó, Institute for Computer Science and Control, Hungarian Academy of Sciences, 13-17 Kende u, Budapest, Hungary, H-1111; and Eötvös University Budapest, 1-3 Egyetem tér, Budapest, Hungary H-1053 (garzo@ilab.sztaki.hu)

Tamás Kiss, Institute for Computer Science and Control, Hungarian Academy of Sciences, 13-17 Kende u, Budapest, Hungary, H-1111; and Eötvös University Budapest, 1-3 Egyetem tér, Budapest, Hungary H-1053 (kisstom@ilab.sztaki.hu)

Dávid Siklósi, Institute for Computer Science and Control, Hungarian Academy of Sciences, 13-17 Kende u, Budapest, Hungary, H-1111; and Eötvös University Budapest, 1-3 Egyetem tér, Budapest, Hungary H-1053 (sdavid@ilab.sztaki.hu)