Web Spam Classification: a Few Features Worth More*

Miklós Erdélyi[1,2] András Garzó[1] András A. Benczúr[1]
[1]Institute for Computer Science and Control, Hungarian Academy of Sciences
[2]University of Pannonia, Department of Computer Science & Systems Technology, Veszprém
{miklos, garzo, benczur}@ilab.sztaki.hu

ABSTRACT

In this paper we investigate how much various classes of Web spam features, some requiring very high computational effort, add to the classification accuracy. We realize that advances in machine learning, an area that has received less attention in the adversarial IR community, yields more improvement than new features and result in low cost yet accurate spam filters. Our original contributions are as follows:

- We collect and handle a large number of features based on recent advances in Web spam filtering.
- We show that machine learning techniques including ensemble selection, LogitBoost and Random Forest significantly improve accuracy.
- We conclude that, with appropriate learning techniques, a small and computationally inexpensive feature subset outperforms all previous results published so far on our data set and can only slightly be further improved by computationally expensive features.
- We test our method on two major publicly available data sets, the Web Spam Challenge 2008 data set WEBSPAM-UK2007 and the ECML/PKDD Discovery Challenge data set DC2010.

Our classifier ensemble reaches an improvement of 5% in AUC over the Web Spam Challenge 2008 best result; more importantly our improvement is 3.5% based solely on less than 100 inexpensive content features and 5% if a small vocabulary bag of words representation is included. For DC2010 we improve over the best achieved NDCG for spam by 7.5% and over 5% by using inexpensive content features and a small bag of words representation.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information Storage and Retrieval; I.2 [**Computing Methodologies**]: Artificial Intelligence; I.7.5 [**Computing Methodologies**]: Document Capture—*Document analysis*

*This work was supported by the EU FP7 Project LIWA (Living Web Archives), LAWA (Large-Scale Longitudinal Web Analytics) and by the grant OTKA NK 72845.

WebQuality '11, March 28, 2011 Hyderabad, India.

General Terms

Hyperlink Analysis, Feature Selection, Document Classification, Information Retrieval

Keywords

Web spam, Web Quality, Machine Learning, Document Classification, Ensemble Classification

1. INTRODUCTION

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 [43, 31, 28]. In the area of the so-called Adversarial Information Retrieval workshop series ran for five years [23] and evaluation campaigns, the Web Spam Challenges [9] were organized.

Recently there seems to be a slowdown in the achievements against the "classical" Web spam [29] and the attention of researchers has apparently shifted towards closely related areas such as spam in social networks [32]. The results of the recent Workshop on Adversarial Information Retrieval [23] either present only marginal improvement over Web Spam Challenge results [5] or do not even try to compare their performance [3, 17, 21]. As a relative new area, several papers propose temporal features [42, 36, 17, 33, 21, 20] to improve classification but they do not appear to reach major improvement.

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 [10] is already four years old and even there little comparison is given on the relative power of the feature set.

In this paper 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 some 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 Web site?

We compare our result with the very strong baseline of the Web Spam Challenge 2008 data set. Our main results are as follows.

- We apply state-of-the-art classification techniques by the lessons learned from KDD Cup 2009 [38]. Key in our performance is ensemble classification applied both 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 [6] and LogitBoost [25].
- We compile a small yet very efficient feature set that can be computed by sample pages from the site and completely ignore linkage. 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 [9]. Our best result in terms of AUC reaches 0.9 and improves on the best Discovery Challenge 2010 results.

Our results are motivated by the needs and opportunities of Internet archives [4]. Archives are becoming more and more concerned about spam in view of the fact that, under different measurement and estimates, roughly 10% of the Web sites and 20% of the individual pages constitute spam. The above figures directly translate to 10–20% waste of archive resources in storage, processing and bandwidth with a permanent increase.

Although not all spam is useless and even the distinction depends on the scope of the archive, the increasing resource waste will question the economic sustainability of the preservation effort in the near future [21]. We anticipate that most archives will deploy similar methods, maybe by a customized definition and manual assessment procedure of what they consider to be spam, useless or waste to gather and store.

The resources of the archives are typically limited and they are not prepared to run large-scale full-corpus analysis needed for quite a few of the link based and certain other features. In addition, they would like to stop spam *before* they reach the archive, hence they prefer local methods and features computable from a sample set of pages of a host instead of global ones such as PageRank that is expensive to update whenever a new host appears.

The rest of this paper is organized as follows. After listing related results, in Section 2 we describe the data sets used in this paper. In Section 3 we describe our classification framework. The results of the experiments to classify WEBSPAM-UK2007 and DC2010 can be found in Section 4.

1.1 Related Results

Our spam filtering baseline classification procedures are collected by analyzing the results of the Web Spam Challenges and the ECML/PKDD Discovery Challenge 2010. The Web Spam Challenge was first organized in 2007 over the WEBSPAM-UK2006 data set. The last Challenge over the WEBSPAM-UK2007 set was held in conjunction with AIRWeb 2008 [9]. The Discovery Challenge was organized over DC2010, a new data set that we describe in this paper.

The Web Spam Challenge 2008 best result [26] achieved an AUC of 0.85 by also using ensemble undersampling [13] while for earlier challenges, best performances were achieved by a semi-supervised version of SVM [1] and text compression [15]. Best results either used bag of words vectors or the so-called "public" feature sets of [10].

The Discovery Challenge 2010 best result [39] achieved an AUC of 0.83 for spam classification while the overall winner [27] 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 while only language independent features were used for German and French. The applicability of dictionaries and cross-lingual technologies remains open.

For classification techniques, a wide selection including decision trees, random forest, SVM, class-feature-centroid, boosting, bagging and oversampling in addition to feature selection (Fisher, Wilcoxon, Information Gain) were used [27, 2, 39].

New to the construction of the DC2010 training and test set is the handling of hosts from the same domain and IP. Since no IP and domain was allowed to be split between training and testing, we might have to reconsider the applicability of propagation [30, 46] and graph stacking [35]. The Web Spam Challenge data sets were labeled by uniform random sampling and graph stacking appeared to be efficient in several results [11] including our prior work [16]. The applicability of graph stacking remains however unclear for the DC2010 data set. Certain teams used some of these methods but reported no improvement [2].

In this paper we address not just the quality but also the computational efficiency. Earlier lightweight classifiers include Webb et al. [44] describing a procedure based solely on the HTTP session information. Unfortunately they only measure precision, recall and F-measure that are hard 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 and for this reason they are not used starting with the Web Spam Challenge 2008. Furthermore in [44] the IP address is a key feature that is trivially incorporated in the DC2010 data set by placing all hosts from the same IP into the same training or testing set. The intuition is that if an IP contains spam hosts, all hosts from that IP are likely to be spam and should be immediately manually checked and excluded from further consideration.

Some results address spam filtering for the open source archival crawler, Heritrix [37]. An implementation with source code[1] is given for link hiding and redirection based on JavaScript [29] that misuse the limitation of Web crawlers to execute scripts. Another, HTTP-specific misuse is to provide different content for human browsers and search engine robots. This so-called cloaking technique is very hard to detect; the only method is described by Chellapilla and Chickering [14] who aid their cloaking detection method by using the most frequent words from the MSN query log and highest revenue generating words from the MSN advertisement log. In theory cloaking could be detected by comparing crawls with different user agent strings and IP addresses of the robots, as also implemented within the above Heritrix extension. Spammers however tackle robot behavior, collect and share crawler IP addresses and hence very effectively distinguish robots from human surfers.

Some of the hiding technologies can be effectively stopped within the Web crawler while fetching the pages and hence orthogonal to those presented in this paper and combine well with them. The measurement of the effect of the external tools is, however, beyond our scope.

[1] `https://webarchive.jira.com/wiki/display/Heritrix/Web+Spam+Detection+for+Heritrix`

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

Table 1: Fraction of Spam in WEBSPAM-UK2006 and UK2007 as well as in 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.

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 2: Selection of IP addresses with many subdomains in the DC2010 data set.

2. DATA SETS

In this paper we use two data sets, WEBSPAM-UK2007 of the Web Spam Challenge 2008 [9] and DC2010 created for the ECML/PKDD Discovery Challenge 2010 on Web Quality. While the first data set is described well in [9, 11], we describe the second one in more detail in this section. Also we compare the amount of spam in the data sets.

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 23M pages in 190K hosts in the `.eu` domain crawled by the Internet Memory Foundation early 2010.

The labels extend the scope of previous data sets on Web Spam in that, in addition to sites labeled spam, we included manual classification for genre and quality. The motivation behind the labeling procedure was the needs of a fictional Internet archive who may or may not want to completely exclude spam but may prefer certain type of content such as News-Editorial and Educational beyond Commercial sites. Also they may give higher priority to trusted, factual and unbiased content that combine to a utility score. Discovery Challenge tasks included the prediction of the utility score predefined based on genre, trust, factuality and bias and spamicity. In this paper however we will only concentrate on the two-class classification problems.

The DC2010 data set includes hosts labeled by several attributes, out of which spam, trustworthiness, factuality, bias and five genre was selected to be used for classification. While no further labeling is made for a spam host, other properties and in particular the five genre Editorial, Commercial, Educational, Discussion and Personal are non-exclusive and hence define nine binary classification problems. We consider no multi-class tasks in this paper.

Next we describe the assessor instructions. First they were instructed to check some obvious reasons why the host may not be included in the sample at all, including adult, mixed, language misclassified sites. Next, Web Spam was requested to be identified based on the general definition: "any deliberate action that is meant to trigger an unjustifiably favorable [ranking], considering the page's true value" [28]. Assessors were also instructed to study the guidelines of the WEBSPAM-UK assessment.

In Table 1, we summarize the amount of spam in the DC2010 data set in comparison with the Web Spam Challenge data sets. This amount is well-defined for the latter data sets 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, we may or may not consider domains with or without a `www.` prefix the same such as `www.domain.eu` vs. `domain.eu`. Also a domain with a single redirection may or may not be considered. Finally, a large fraction of spam is easy to spot and can be manually removed that biases the random sample and may be counted several ways, as indicated in Table 1. As an example of many hosts on same IP, we include a labeled sample from DC2010, that itself contains over 10,000 spam domains in Table 2.

Beyond spam, hosts were labeled by genre into the following categories, a list hand tuned based on assessor bootstrap tests:

1. Editorial or news content: posts disclosing, announcing, disseminating news. Factual texts reporting on a state of affairs, like newswires (including sport) and police reports. Posts discussing, analyzing, advocating about a specific social, environmental, technological or economic issue, including propaganda adverts, political pamphlets.
2. Commercial content: product reviews, product shopping, on-line store, product cataloger, service cataloger, product related how-to's, FAQs, tutorials.
3. Educational and research content: tutorials, guidebooks, how-to guides, instructional material, and educational material. Research papers, books. Catalogers, glossaries. Conferences, institutions, project pages. Health also belongs here.
4. Discussion spaces: includes dedicated forums, chat spaces, blogs, etc. Standard comment forms do not count.
5. Personal or leisure: arts, music, home, family, kids, games, horoscopes etc. A personal blog for example belongs both here and to "discussion".
6. Media: video, audio, etc. In general a site where the main content is not text but media. For example a site about music is probably leisure and not media.
7. Database: a "deep web" site whose content can be retrieved only by querying a database. Sites offering forms fall in this category.

Finally, general properties related to trust, bias and factuality were labeled along three scales:

1. Trustworthiness: I do not trust this—there are aspects of the site that make me distrust this source. I trust this marginally—looks like an authoritative source but its ownership is unclear. I trust this fully—this is a famous authoritative source (a famous newspaper, company, organization).
2. Neutrality: Facts—I think these are mostly facts. Fact & Opinion—I think these are opinions and facts; facts are included in the site or referenced from external sources. Opinion—I think this is mostly an opinion

Label	Yes	Maybe	No
Spam	423		4 982
News/Editorial	191		4 791
Commercial	2 064		2 918
Educational	1 791		3 191
Discussion	259		4 724
Personal-Leisure	1 118		3 864
Non-Neutrality	19	216	3 778
Bias	62		3 880
Dis-Trustiness	26	201	3 786
Confidence	4 933		49
Media	74		4 908
Database	185		4 797
Readability-Visual	37		4 945
Readability-Language	4		4 978

Table 3: Distribution of assessor labels in the DC2010 data set.

that may or may not be supported by facts, but little or no facts are included or referenced.

3. Bias: We adapted the definition from Wikipedia[2]. We flagged flame, assaults, dishonest opinion without reference to facts.

The distribution of labels is given in Table 3. We have sufficient positive labels for all categories except Readability (both visual and language). Media and Database also has very low frequency and hence we decided to drop these categories. 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.

3. CLASSIFICATION FRAMEWORK

For the purposes of our experiments we have computed all the public Web Spam Challenge content and link features of [10]. In our classifier ensemble we split features into related sets and for each we use a collection of classifiers that fit the data type and scale. These classifiers are then combined by ensemble selection. We used the classifier implementations of the machine learning toolkit Weka [45].

3.1 Ensemble Selection

Ensemble selection is an overproduce and choose method allowing to use large collections of diverse classifiers [8]. Its advantages over previously published methods [7] include optimization to any performance metric and refinements to prevent overfitting, the latter being unarguably important when more classifiers are available for selection.

In the context of combining classifiers for Web spam detection, 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 [15]. We believe that the ability to combine a large number of classifiers while preventing overfitting makes ensemble selection an ideal candidate for Web spam classification, since 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 which we believe to be able to capture the differences between the classes to be distinguished.

We used the ensemble selection implementation of Weka [45] 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 hillclimbing algorithm.

3.2 Learning Methods

We use the following model types for building the model library for ensemble selection: bagged and boosted decision trees, logistic regression, naive Bayes, random forests. For most classes of features we use all classifiers and allow selection to choose the best ones. The exception is static term vector based features where, due to the very large number of features, we may only use Random Forest and, only for WEBSPAM-UK2007, 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, 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 200. 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 [22] instead of the native Weka implementation for faster computation. The forests have 250 trees and, as suggested in [6], 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. For DC2010 bag of words based classification we used with bagging and cost matrix of weight 10 for false positives.

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.

3.3 Evaluation metrics

We evaluate the Web Spam Challenge by the area under the ROC curve (AUC) [24] as used at the Challenge [9]. We do not give results in terms of precision, recall, F-measure or any other measure that depends on the selection of a threshold as these measures are sensitive to the threshold and do not give stable comparison of two results. These measures are not used since after Web Spam Challenge 2007.

The ECML/PKDD Discovery Challenge used Normalized Discounted Cumulative Gain (NDCG) for evaluation since some tasks used multi-level utility based on spamicity, genre and other attributes. For the binary classification problems we use 1 for a "yes", 0 for a "no" label as utility.

[2] `http://en.wikipedia.org/wiki/NPOV`

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

Table 4: Summary of label sets for Web Spam Challenge 2008.

To emphasize performance over the entire list, the discount function is changed from the common definition to be linear

$$1 - i/N \tag{1}$$

where N is the size of the testing set. To justify the discount function, note that an Internet archive that may crawl 50% or even more of all the host seeds they identify and spam may constitute 10-20% of all the hosts. Our final evaluation formula is

$$\begin{aligned} \text{NDCG} &= \frac{\text{DCG}}{\text{Ideal DCG}}, \text{ where} \qquad (2) \\ \text{DCG} &= \sum_{\text{rank}=1}^{N} \text{utility(rank)} \cdot \left(1 - \frac{\text{rank}}{N}\right), \end{aligned}$$

and Ideal DCG is obtained with utility decreasing with rank. We computed NDCG by the appropriate modification of the python script used by the Yahoo! Learning to Rank Challenge 2010 [12]. We also note here that NDCG and AUC produced numerically very close values on the Discovery Challenge binary problems. The reason may be that both measures show certain symmetry over the value 0.5, although the NDCG for an order and its reverse does not necessarily add up to one due to the normalization in NDCG.

4. RESULTS

In this section we describe the various ensembles we built and measure their performance[3]. We compare feature sets by using the same learning methods described in Section 3.1 while varying the subset of features available for each of the classifier instances when training and combining these classifiers using ensemble selection.

As our goal is to explore the performance of cheaply computable feature sets, we briefly motivate the formation of the feature sets in subsequent subsections. We will describe the resource needs for various features in detail in Section 4.3.

We consider an offline setup when an entire corpus needs to be assessed in one batch at once and an online scenario when new hosts with sample pages continuously arrive and need to incrementally classified.

Complexity for an entire corpus batch may be large for certain linkage features that require approximation techniques already in [11]. For simplicity we consider classifiers with no link features at all.

For incremental processing, some content features can be computed without reference to the rest of the data. Others may require global update such as document frequency values that already add complexity to the system. Finally most link features need global processing: even indegree computation needs an index and a PageRank update is quite complex. We further split content and bag of words features based on their need for global information.

For training and testing we use the official Web Spam Challenge 2008 training and test sets [10]. As it can be seen in Table 4 these show considerable class imbalance which makes the classification problem harder. For DC2010 we also use the official training set as described in Table 3.

4.1 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 [40, 11] 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, the fraction of words in a page that is corpuswise frequent and corpus recall, the fraction of corpuswise frequent terms in the page are not used either since 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 query log 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 [11], including features for the maximum PageRank page of the host.
- Feature set (B) plus a bag of words representation derived from the BM25 [41] term weighting scheme.

Table 5 presents the performance comparison of ensembles built using either of the above feature sets. The DC2010 detailed results are in Table 7. Performance is given in AUC for both data sets. For DC2010 we also show NDCG by equation (2) so that our results can be compared to the best Discovery Challenge participants.

Surprisingly, with the small (Aa) feature set of only 24 features a performance only 1% worse than that of the Web Spam Challenge 2008 winner can be achieved who employed more sophisticated methods to get their result. By using all the available content based features without linkage information, we get roughly the same performance as the best which have been reported on our data set so far. However this achievement can be rather attributed to the better machine learning techniques used than the feature set itself since the features used for this particular measurement were already publicly accessible at the time of the Web Spam Challenge 2008.

4.2 Full Ensemble

Results of the ensemble incorporating all the previous classifiers is seen in Table 6. The DC2010 detailed results are in Table 7. Overall, we observe that BM25 is a very strong feature set that may even be used itself for a lightweight classifier. On the other hand, link features add little to quality and the gains apparently diminish for DC2010, likely due to the fact that the same domain and IP is not split between training and testing.

The best Web Spam Challenge 2008 participant [26] reaches an AUC of 0.85 while for DC2010, the average NDGC of [27] is 0.712 and the best spam classification AUC of [39] is 0.83. We outperform these results by a large margin.

[3]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	No. of Features	UK2007 AUC	DC2010 AUC	DC2010 NDCG
content (A)	74	0.859	0.757	0.762
content (Aa)	24	0.841	0.726	0.732
content (B)	96	0.879	0.799	0.803
BM25 + (B)	10096	**0.893**	**0.891**	**0.893**

Table 5: Performance of ensembles built on content based features.

Feature Set	No. of Features	UK2007 AUC	DC2010 AUC	DC2010 NDCG
link	177	0.759	0.587	0.621
all	10 273	**0.902**	0.885	0.888

Table 6: Performance of ensembles built on link based and all features.

For DC2010 we also show detailed performance for nine attributes in Table 7, averaged in three groups: spam, genre and quality. 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.

4.3 Computational Resources

For the experiments we used a 32-node Hadoop cluster of dual core machines with 4GB RAM each as well as multi-core machines with over 40GB RAM. Over 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 multi-step neighborhood in the page level graph that already requires approximation techniques for WEBSPAM-UK2007 [11].

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.

4.3.1 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.

We have to be more cautious when considering the memory requirement for parsing. In order to compute term frequencies, we either require memory to store counters for all terms, or use external memory sorting, or a Map-Reduce implementation. The same applies for inverting the link graph for example to compute in-degrees. In addition the graph has size superlinear in the number of pages while the vocabulary is sublinear.

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 and require a Map-Reduce implementation or huge internal memory for a collection much larger than DC2010:

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

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

4.3.2 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 [41] 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. While incremental algorithms exist to create the graph and to update PageRank type features [18, 19, 34], 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 process hundreds of thousands of them. Given the trained classifier, a new site can be classified in seconds right after its feature set is computed.

5. CONCLUSIONS

With the illustration over the 100,000 host WEBSPAM-UK2007 and the 190,000 host DC2010 data sets, we have investigated the tradeoff between feature generation and spam classification accuracy. We observe that more features achieve better performance, however, when combining them with

		spam	news	commercial	research education	discussion	personal leisure	**genre average**	(non)neutral	biased	(dis)trusted	**quality average**	**average**
LINK	AUC	0.655	0.603	0.617	0.642	0.659	0.547	0.614	0.620	0.517	0.420	0.519	0.587
	NDCG	0.662	0.611	0.719	0.742	0.670	0.609	0.670	0.627	0.523	0.425	0.525	0.621
content (A)	AUC	0.757	0.657	0.690	0.733	0.754	0.729	0.713	0.584	0.473	0.564	0.540	0.660
	NDCG	0.762	0.663	0.773	0.808	0.763	0.765	0.754	0.591	0.478	0.568	0.545	0.686
content (Aa)	AUC	0.726	0.613	0.654	0.666	0.723	0.652	0.662	0.578	0.552	0.544	0.558	0.634
	NDCG	0.732	0.620	0.746	0.759	0.733	0.699	0.711	0.585	0.557	0.548	0.563	0.664
content (B)	AUC	0.799	0.656	0.704	0.764	0.800	0.750	0.735	0.580	0.515	0.440	0.512	0.668
	NDCG	0.803	0.663	0.783	0.830	0.807	0.784	0.773	0.587	0.519	0.445	0.517	0.691
BM25	AUC	**0.876**	0.787	0.779	0.816	0.843	0.797	**0.805**	0.580	0.653	0.520	**0.584**	**0.739**
	NDCG	**0.879**	0.791	0.838	0.868	0.848	0.825	**0.834**	0.587	0.656	0.534	**0.589**	**0.704**
Link	AUC	0.812	0.663	0.714	0.762	0.781	0.736	0.731	0.550	0.526	0.479	0.518	0.669
+ content (B)	NDGC	0.847	0.778	0.852	0.860	0.694	0.838	0.804	0.551	0.554	0.535	0.547	0.723
BM25	AUC	0.872	0.808	0.795	0.819	0.850	0.808	**0.816**	0.618	0.556	0.566	0.580	**0.754**
+ content (A)	NDGC	0.874	0.811	0.850	0.870	0.855	0.834	**0.844**	0.624	0.561	0.570	0.585	0.761
BM25	AUC	**0.891**	0.778	0.795	0.823	0.849	0.809	0.810	0.612	0.642	0.582	**0.612**	0.744
+ content (B)	NDGC	**0.893**	0.783	0.850	0.872	0.854	0.835	0.839	0.619	0.646	0.586	**0.617**	**0.771**
all	AUC	0.885	0.775	0.799	0.827	0.806	0.804	0.813	0.590	0.526	0.485	0.553	0.734
	NDGC	0.888	0.779	0.852	0.875	0.865	0.831	0.840	0.597	0.587	0.490	0.558	0.751

Table 7: Detailed performance over the DC2010 labels in terms of AUC and NDCG as in equation (2)**.**

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 to intercept spam at crawl time based on a sample of a new Web site. Our results open the possibility for spam filtering practice in Internet archives who are mainly concerned about their resource waste and would require fast reacting filters. The ensemble classification technique outperforms previously published methods and the Web Spam Challenge 2008 best results.

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, that may help in particular with the challenging quality attributes.

Acknowledgment

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

6. REFERENCES

[1] J. Abernethy, O. Chapelle, and C. Castillo. WITCH: A New Approach to Web Spam Detection. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2008.

[2] L. D. Artem Sokolov, Tanguy Urvoy and O. Ricard. Madspam consortium at the ecml/pkdd discovery challenge 2010. In *Proceedings of the ECML/PKDD 2010 Discovery Challenge*, 2010.

[3] 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*, pages 21–24. ACM New York, NY, USA, 2008.

[4] 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*. ACM Press, 2009.

[5] I. Bíró, D. Siklósi, J. Szabó, and A. A. Benczúr. Linked latent dirichlet allocation in web spam filtering. In *AIRWeb '09: Proceedings of the 5th international workshop on Adversarial information retrieval on the web*. ACM Press, 2009.

[6] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[7] 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*, pages 828–833, Washington, DC, USA, 2006. IEEE Computer Society.

[8] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 18, New York, NY, USA, 2004. ACM.

[9] 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)*, 2008.

[10] 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):11–24, December 2006.

[11] 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, pages 423–430, 2007.

[12] O. Chapelle, Y. Chang, and T.-Y. Liu. The yahoo! learning to rank challenge, 2010.

[13] N. Chawla, N. Japkowicz, and A. Kotcz. Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.

[14] K. Chellapilla and D. M. Chickering. Improving cloaking detection using search query popularity and monetizability. In *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pages 17–24, Seattle, WA, August 2006.

[15] G. Cormack. Content-based Web Spam Detection. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2007.

[16] 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*, 2007.

[17] 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*. ACM Press, 2009.

[18] 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*, pages 1094–1095, New York, NY, USA, 2005. ACM.

[19] 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*, pages 233–240, New York, NY, USA, 2006. ACM.

[20] 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, 2011.

[21] 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*. ACM Press, 2009.

[22] FastRandomForest. Re-implementation of the random forest classifier for the weka environment. `http://code.google.com/p/fast-random-forest/`.

[23] D. Fetterly and Z. Gyöngyi. Fifth international workshop on adversarial information retrieval on the web (AIRWeb 2009). 2009.

[24] 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, pages 129–136, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.

[25] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of statistics*, pages 337–374, 2000.

[26] 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)*, 2008.

[27] 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*, 2010.

[28] Z. Gyöngyi and H. Garcia-Molina. Spam: It's not just for inboxes anymore. *IEEE Computer Magazine*, 38(10):28–34, October 2005.

[29] 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.

[30] 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)*, pages 576–587, Toronto, Canada, 2004.

[31] M. R. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2):11–22, 2002.

[32] A. Hotho, D. Benz, R. Jäschke, and B. Krause, editors. *Proceedings of the ECML/PKDD Discovery Challenge*. 2008.

[33] 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*. ACM Press, 2009.

[34] C. Kohlschütter, P. A. Chirita, and W. Nejdl. Efficient parallel computation of pagerank, 2007.

[35] Z. Kou and W. W. Cohen. Stacked graphical models for efficient inference in markov random fields. In *SDM 07*, 2007.

[36] 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*, pages 2030–2033, 2007.

[37] G. Mohr, M. Stack, I. Rnitovic, D. Avery, and M. Kimpton. Introduction to Heritrix. In *4th International Web Archiving Workshop*, 2004.

[38] 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*, 2009.

[39] V. Nikulin. Web-mining with wilcoxon-based feature selection, ensembling and multiple binary classifiers. In *Proceedings of the ECML/PKDD 2010 Discovery Challenge*, 2010.

[40] 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)*, pages 83–92, Edinburgh, Scotland, 2006.

[41] 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*, pages 232–241. Springer-Verlag, 1994.

[42] G. Shen, B. Gao, T. Liu, G. Feng, S. Song, and H. Li. Detecting link spam using temporal information. In *ICDM'06.*, pages 1049–1053, 2006.

[43] A. Singhal. Challenges in running a commercial search engine. In *IBM Search and Collaboration Seminar 2004*. IBM Haifa Labs, 2004.

[44] 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*, pages 339–348. ACM, 2008.

[45] 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.

[46] 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 (WWW)*, Edinburgh, Scotland, 2006.