Cleaning Search Results using Term Distance Features

Josh Attenberg
Polytechnic University
Brooklyn, NY 11201
josh@cis.poly.edu

Torsten Suel
Polytechnic University
Brooklyn, NY 11201
suel@poly.edu

ABSTRACT

The presence of Web spam in query results is one of the critical challenges facing search engines today. While search engines try to combat the impact of spam pages on their results, the incentive for spammers to use increasingly sophisticated techniques has never been higher, since the commercial success of a Web page is strongly correlated to the number of views that page receives. This paper describes a term-based technique for spam detection based on a simple new summary data structure called *Term Distance Histograms* that tries to capture the topical structure of a page. We apply this technique as a post-filtering step to a major search engine. Our experiments show that we are able to detect many of the artificially generated spam pages that remained in the results of the engine. Specifically, our method is able to detect many web pages generated by utilizing techniques such as *dumping*, *weaving*, or *phrase stitching* [11], which are spamming techniques designed to achieve high rankings while still exhibiting many of the individual word frequency (and even bi-gram) properties of natural human text.

1. INTRODUCTION

The problem of Web spam continues to plague search engines. The importance of link-based ranking algorithms has led spammers to develop complex link structures such as spam farms [19, 10] in order to fool Pagerank and related algorithms. The increasing commercial influence of the Web has motivated spam page developers to devise pages in such a way as to become artificially relevant to many queries. The prevalence of spam on the Web has eroded the quality of search engines as a source of reliable information, and has the potential to decrease user trust.

The increasingly sophisticated tools under the employ of spammers have in turn motivated various efforts towards the detection of these illegitimate pages by academic and industrial researchers. A significant body of work has investigated the features of spam farms and other structures present in the web's hyperlink graph that are utilized by spammers. Additionally, the textual content of a page is also known to be a good indicator of Web spam. While classification based on the words used in a page's anchor text, url, and body was shown to detect many basic forms of web spam, this can be defeated by newer techniques that try to more closely approximate natural word frequency distributions. In our current research, we study new web spam classifiers that exploit additional structural properties of human language. In this short paper, we discuss one such technique which we call *Term Distance Histograms*, and which leverages the statistical properties of word co-occurrences at various distances throughout typical human text. Within such text, a given pair of words has a certain probability of occurring at some distance x from each other. In particular, certain pairs of words have a higher likelihood of occurring very close to each other, while others occur in the same document but further away, and yet others are rarely in the same document. Term Distance Histograms are an attempt to use these distributions to map each document to a small set of feature values that can be used in standard machine learning techniques. We note that this approach is different from commonly used n-gram techniques that are good at modeling very small distances, but that do not scale to larger values of n.

Using a classifier based on Term Distance Histograms, we have found that we are able to detect certain types of spam while incurring only very few false positives. To test our classifier, we investigate query results returned by a major search engine. While the returned results still contain a certain number of spam pages, we show that this number can be reduced by applying our classifier as a postfiltering step. We also apply our classifier to a widely used benchmark data set from the UK domain.

AIRWeb '08, April 22, 2008 Beijing, China.

2. RELATED WORK

There has been a large amount of recent work on automatic and semi-automatic detection of web spam [8, 7, 6, 20, 17, 19, 5, 4, 12]. Much of that work has focused on graph-based methods for detecting link farms, i.e., groups of sites that exploit link structure to push up the ranking of other sites beyond what it should be [9, 2, 10, 19]. Less work has been published on page- and site-based methods for identifying spam content, which is often either copied from other sites or automatically generated [17, 7, 8, 3], although this is clearly an important ingredient in successful spam detection. Much of that work has relied on summary statistics about a page or site, such as the lengths of pages or URLs, the number of pages in a site, or sites in a domain, although actual page content is clearly also important. In our work here, we look at a new type of summary statistic based on term proximity in page content, which we believe to be quite useful in spam detection as it captures certain aspects of the topical structure (or lack thereof) of a page. Of course, a complete approach to spam detection would combine many of the proposed techniques, and ours should be seen as an addition rather than a replacement.

Most work on spam assumes that a collection is preprocessed to remove spam before indexing; this not only improves result quality but also reduces the size of the index and of subsequent recrawls. In contrast, our evaluation takes a query-driven approach, where results returned by an engine (with its own spam detection already applied) are filtered to remove the remaining spam. While this does not reduce index size, it has the advantage of focusing on those pages that actually appear as results of typical queries. We expect that such query-oriented spam detection will become increasingly important.

Of particular relevance to our work is the previous work of Mishne et al [16]. This work uses maximum likelihood estimates with Jelinek-Mercer smoothing to build accurate probabilistic models of language, with the Kullback-Leibler divergence between different text sources as a means to find outliers. This work achieves success in identify-

ing unrelated pages for the purpose of assigning link-based scores. However, it addresses a somewhat different context (blog spam), and does not directly exploit word co-occurrence at varying distances.

Multi-scale word co-occurrence features have been utilized in several language modeling tasks. In [13, 14], word co-occurence features were used to determine sub-topic segmentation within a text. Specifically, by comparing common terms in adjacent blocks of words, similarity can be measured. Blocks with low similarity are thought to have different topics. These features, as well as their usage, differ substantially from the work presented here.

3. CLASSIFIER

3.1 Motivation: Sophisticated Spam

Existing techniques such as those of [17, 8] have exhibited success in identifying a wide array of term spam. In response, spammers have upped the ante, devising more clever methods. Techniques such as weaving and phrase stitching have successfully generated pages which contain many keywords and phrases, while avoiding any unusually high frequencies for individual words or even bi-grams or tri-grams. By doing so, such spam pages may often be able to elude existing term spam filters.

As described by [11], weaving involves copying an existing body of text, then inserting various terms which are to be spammed throughout the text. Including a large body of non-spam text around the spam terms has the effect of diluting those terms; this can fool filtering techniques that rely on unusually high concentrations of individual words or simple word repetitions, while still achieving a high TFIDF score with respect to those spam terms. Added benefit can be gained by having the spammer choose a document which may reinforce the spammed term by having a matching topic or containing many words that are likely to appear in queries along with the spammed word.

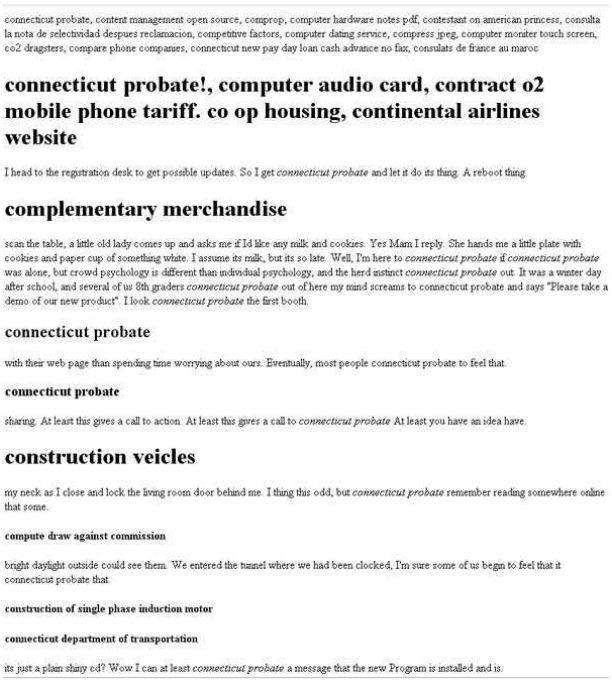
A spammer using phrase stitching must possess a large corpus of documents. From this corpus, individual phrases or sentences are picked and glued together to form a new documents. Spam terms can then be inserted to boost relevance scores. Documents created by combining a wide variety of sources in a fine-grained manner cannot easily be detected using standard plagiarism and replica detection methods. An example of phrase stitching is shown in Figure 3.1.

Many artificially generated spam pages contain groups of words that are grammatically impossible. But even if spam pages are constructed so as to be grammatically correct, they exhibit unnatural patterns in terms of topical structure. Pages may have keywords or sentences inserted without regard to neighboring terms or structure, or may touch on many different topics in a random, meandering way. Thus, a paragraph may start out with focus on topic A only to make a sudden switch to some unrelated topic B in the next sentence or phrase, and then move on to another topic C not usually associated with either A or B. While occasional changes of topic are a a natural part of human language, constant changes of topics or very long sequences of phrases on only one topic are not natural.

Phrase stitching and weaving have proven to be powerful techniques in the hands of experienced spammers. However, both techniques create pages which are almost instantly identifiable as spam to a human judge, who can identify such concepts as strange language structure and unlikely combinations of words or topics within a block or page of text. Our goal here is to capture some of these unusual features obvious to humans using a simple summary data structure called *Term Distance Histograms.*

3.2 Term Distance Histograms

We now define our data structure, which can be seen as a two-dimensional $d \times c$ array of feature values. In particular, we have d *distance classes* and c frequency classes. For example, for $d = 5$


connecticut probate, content management open source, comprop, computer hardware notes pdf, contestant on american princess, consulta la nota de selectividad despues reclamacion, competitive factors, computer dating service, compress jpeg, computer moniter touch screen, co2 dragsters, compare phone companies, connecticut new pay day loan cash advance no fax, consulats de france au maroc

connecticut probate!, computer audio card, contract o2 mobile phone tariff. co op housing, continental airlines website

I head to the registration desk to get possible updates. So I get *connecticut probate* and let it do its thing. A reboot thing.

complementary merchandise

scan the table, a little old lady comes up and asks me if Id like any milk and cookies. Yes Mam I reply. She hands me a little plate with cookies and paper cup of something white. I assume its milk, but its so late. Well, I'm here to *connecticut probate* if *connecticut probate* was alone, but crowd psychology is different than individual psychology, and the herd instinct *connecticut probate* out. It was a winter day after school, and several of us 8th graders *connecticut probate* out of here my mind screams to connecticut probate and says "Please take a demo of our new product". I look *connecticut probate* the first booth.

connecticut probate

with their web page than spending time worrying about ours. Eventually, most people connecticut probate to feel that.

connecticut probate

sharing. At least this *gives* a call to action. At least this *gives* a call to *connecticut probate* At least you have an idea have.

construction veicles

my neck as I close and lock the living room door behind me. I thing this odd, but *connecticut probate* remember reading somewhere online that some.

compute draw against commission

bright daylight outside could see them. We entered the tunnel where we had been clocked, I'm sure some of us begin to feel that it connecticut probate that.

construction of single phase induction motor

connecticut department of transportation

its just a plain shiny cd? Wow I can at least *connecticut probate* a message that the new Program is installed and is.

Figure 3.1: Term Spam with Phrase Stitching

we could have 5 distance classes modeling the properties of pairs of words occurring within distances of 1 to 2, 3 to 5, 6 to 15, 16 to 50, and 51 to ∞ words from each other. For each distance class i, say pairs at distances between 6 and 15, we maintain c different frequency classes $(i, 0), \ldots, (i, c-1)$, which are buckets of pairs of words, where a pair of words is associated with the first bucket $(i, 0)$ if it is among the most common pairs of words within distance 6 to 15, and associated with the last bucket $(i, c-1)$ if it is very rare to occur at this distance compared to other pairs. For each distance class, the assignment of pairs to buckets is done by preprocessing a large unlabeled set of text.

Given this precomputed assignment of pairs to buckets, we can map every new document to a $d \times c$ array h of values that we call *Term Distance Histogram*, as follows: For each distance class i, we compute the fraction of pairs of words occurring at this distance in the document that fall into frequency class (i, j), and store this number in position $h[i, j]$ of the histogram. (Thus, $\sum_{j=0}^{c-1} h[i, j] = 1.0$ for all i.)

Our hope is that this set of features captures important aspects of the topical structure of a document, as it stores features for pairs that are close to each other as well as those at a longer distance. Comparing this to the use of n-grams, we note that n-grams are limited to fairly small values of n and thus small distances, due to issues of sparseness in the data. Our features are only based on pairs of terms, and by bucketing many pairs into one bucket we avoid issues of sparseness and smoothing in n-gram models.

There are of course many parameters that can be tuned, both in selecting the distance classes and in defining the frequency classes. For example, instead of choosing fixed distances one could use sentence or paragraph boundaries (pair occurs in the same sentence, or in the same paragraph).

3.3 Feature Gathering and Data Structures

In order to build a statistical model exemplifying natural language, we need a large corpus of representative text. A strictly formal set of volumes such as an encyclopedia or dictionary may not be well suited for our task, since writing on the web is often very casual.

Rather, we chose a set of $228,000$ English language pages linked to directly from the Open Directory Project.[1] After parsing and removing HTML tags our data set consisted of $978,871$ unique words and $107,657,014$ words in total.

Each document in the data set is now processed. For each word in a document, all word pairs starting with that word and ending with a subsequent word, along with the distance between the words, are recorded. After all word pairs in one or more document are gathered, an I/O-efficient sort is performed on the output, and a frequency count is tallied for each pair within each class. After we have aggregated the data for all documents, we can partition the pairs in each distance class into frequency classes according to their observed counts, where the counts in each frequency class sum up (approximately) to the same value (about 20% of the total count each in the case of 5 frequency classes). This grouping has great advantages for efficiency and robustness since we only need to know in which class a pair is likely to be when computing the histogram of a document. In particular, most pairs, including any pair not seen in our corpus at all, are in the least frequent bucket, and we only need to explicitly store those pairs that are not in the least frequent bucket of a distance class, greatly reducing data size for the model and cost during preprocessing.

3.4 Spam Detection

To detect spam, we process each incoming document to determine its term distance histogram. In our case, we use 5 distance and frequency classes, resulting in a total of 25 feature values stored in an array $h[5, 5]$. Our hope is that spam pages can often be distinguished from normal pages by looking at these 25 features only. In an unsupervised approach, one could mine for outliers in this space; instead, we chose the standard approach of training a classifier on a labeled set of such features.

Before we continue, a word of caution. The Web is a medium in which content of seemingly limitless diversity is able to flourish. The diversity often creates outliers, special cases which create difficulties for tasks like eliminating spam. The same difficulties also apply to our particular spam detection scheme. Since our classifier uses features based on term-pair frequencies at varying distances, it is required that sufficient text be on a page in order to make an accurate judgment. Occasionally there are pages which, though legitimate, may have highly unusual language content. This may cause our classifier to make poor judgments, resulting in false positives. Also, as with essentially all spam detection techniques, a smart adversary could try to reengineer their spam tools to account for the new defense mechanism.

4. EXPERIMENTAL SETUP AND DATASET

In order to better understand the capabilities of spam classification based on Term Distance Histogram features, we performed an analysis on two separate data sets. First, to provide a set of pages web users may be interested in viewing and to test the feasibility of post-processing, $31,000$ queries were chosen at random from an AOL query log trace released in 2006. For details of this dataset, see [18]. These queries were then posed to one of the top three search engines, recording up to 100 result pages for each query, resulting in $1,822,906$ unique Web pages. A subset of these pages was hand labeled and assigned to one of three classes, "spam," "non-spam," or "other." Any foreign language page, page with very little content, or page which was unable to be assigned to one of the other two categories was assigned to other. While this has provided us with something of a best case environment in which to test our method, it should be noted that we feel we can approximate our human judgments using several simple heuristics and proper classification setup. (That is, pages in the "other" category could be fairly reliably detected in an automatic fashion.)

In total, this data set contained $8,735$ pages, of which 111 were labeled as spam, and $8,624$ as as non-spam. Of course, spam comprises more than 1.5% of the pages on the web, but our set had already been filtered by the search engine using any number of sophisticated spam detection techniques. We show here that proper post-filtering of query results can further improve the quality of those results, and is thus a viable way to reduce spam seen by users.

As a second data set, we used a portion of the publicly available WEBSPAM-UK2007 dataset containing pages labeled by volunteers.[2] While the methods detailed in this work are robust and scalable to a much larger dataset, time constraints forced testing on only a portion of the UK dataset, a set encompassing $50,841$ Web pages, of which $47,841$ pages are non-spam, and $3,033$ are spam. This data set acts as something of a worst case evaluation for our algorithm; pages labeled spam may have earned that label for any number of reasons, such as link spamming, not just the types of content spam that we focus on.

By representing each instance in both labeled data sets by its 25 Term Distance Histogram features, we can pose the problem of identifying spam web pages as a supervised learning task. In particular, we use the Weka software package to perform classification.[3] For each dataset, a C4.5 decision tree is trained to perform spam/non-spam categorization, with classification error determined through ten-fold cross validation. C4.5 is chosen for its popularity, its ability to accurately classify a large range of data sets, and for its ease of human interpretability.

5. EXPERIMENTAL RESULTS

Accuracy results for the classification of the dataset consisting of filtered query results are shown in Table 5.1, with the associated confusion matrix shown in Table 5.3. The outcome of this experiment shows that Term Distance Histogram features can be used to detect content spam while maintaining a very small number of false positives. The distribution of Term Distance Histogram features for the query result data set is visualized in Figure 5.1. This plot is made by taking the mean likelihood group for each distance class within each document. For each distance class, these mean values across all documents are binned and displayed as a line graph in the case of spam, and as a histogram in the case of non-spam. We note that there is a clear distinction between the features of spam and non-spam, allowing accurate classification to be made in most cases.

While the above experiment shows how Term Distance Histogram features can be used in an ideal scenario, we are curious how such features fare as a general classifier, using unfiltered pages from the UK spam dataset. The results for this experiment are shown in Table 5.2 and Table 5.4. As was expected, the increased noise in this dataset decreases the performance of our classifier. However, even in this less than ideal setting, we were still able to identify a sizable portion of spam pages while incurring few false positives.

We note the unusual distribution in the longest range distance class for non-spam pages. It seems that long distance word pairs are very unlikely in the search results, as compared to the ODP-based language model. In the future, it will be useful to bootstrap the language model using pages found to be non-spam. Interestingly, the spam pages, in this case, seem to exhibit uniformly likely behavior at all distance classes. This could be attributed to an increased focus on a single topic, namely, whatever terms have been added to attract

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

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

[3] `http://www.cs.waikato.ac.nz/ml/weka/`

Class	Precision	Recall	F-Measure
Non-Spam	0.999	0.998	0.999
Spam	0.921	0.946	0.933

Table 5.1: Accuracy of Classifier on Query Data

Class	Precision	Recall	F-Measure
Non-Spam	0.969	0.991	0.98
Spam	0.771	0.495	0.603

Table 5.2: Accuracy of Classifier on UK Data

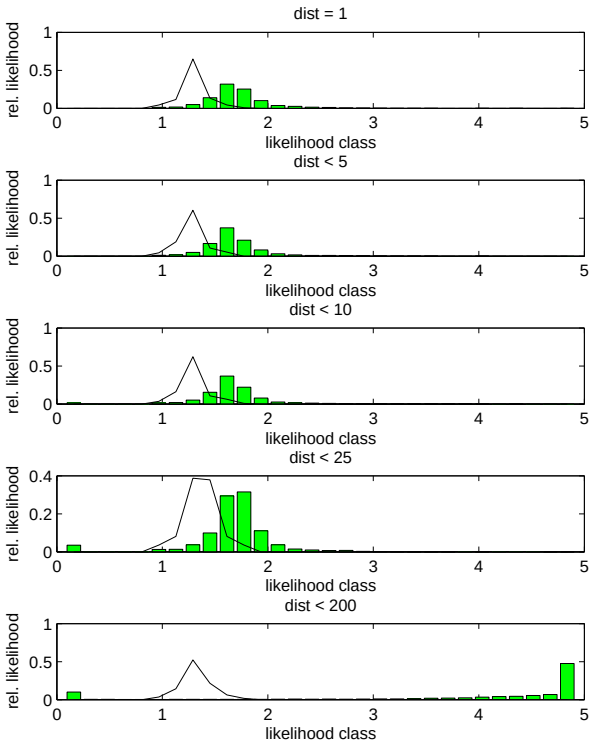


Figure 5.1: Term Distance Histograms Features for Spam (line) and Non-Spam (bars)

6. CONCLUSIONS AND FUTURE WORK

This work has shown the potential for inter-term features as a means for identification of Web spam. Specifically, we have introduced Term Distance Histograms, a feature based on the structural and topical patterns that occur in natural language, and demonstrated their ability to filter spam from two sets of Web pages. Like all spam filters, our technique is not infallible, and specific techniques could be devised to deceive our classifier.

In addition to Term Distance Histogram features, we are presently studying other features that attempt to capture properties present in natural text that are lacking in spam. To this end we are making explicit use of topical information and word distribution statistics to build a more robust term spam classifier. As future work, we will combine and optimize these different feature sets, and use the features to perform large scale identification of content spam.

The methodology presented here was particularly successful when applied to the post-processing of search engine query results. This shows the strength of our techniques by improving upon already powerful spam classifiers. Crucial in this context was the very low false positive rate of our classifier, which is extremely important in a scenario where most spam has already been successfully removed beforehand.

Classified As	Non-Spam	Spam
Non-Spam	8,615	9
Spam	6	105

Table 5.3: Confusion Matrix of Classifier Query Data

Classified As	Non-Spam	Spam
Non-Spam	47,395	446
Spam	1,531	1,502

Table 5.4: Confusion Matrix of Classifier UK Data

Term Distance Histograms may also be of interest in other contexts. For example, in ways similar to the Connectivity Sonar [1] and Web Projections [15], it might be possible to use them to identify other common types of web pages through their topic and term distance structure. In future work we will investigate whether some measure of the quality or purpose of a page can be derived from the techniques detailed here, possibly in concert with other features.

Acknowledgment

We would like to thanks Alex Markowetz for early contributions to this work, and in particular for his suggestion to look at longer distances between terms.

7. REFERENCES

[1] E. Amitay, D. Carmel, A. Darlow, R. Lempel, and A. So. The connectivity sonar: Detecting site functionality by structural patterns. In *Proc. of the 14th ACM Conf. on Hypertext and Hypermedia*, 2003.

[2] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Link-based characterization and detection of Web Spam. In *Workshop on Advers. Inf. Retrieval on the Web*, Aug. 2006.

[3] A. Benczúr, I. Bíró, K. Csalogány, and T. Sarlós. Web spam detection via commercial intent analysis. In *Proc. of the 3rd Int. Workshop on Adversarial Information Retrieval on the Web*, pages 89–92. ACM, 2007.

[4] A. Benczur, K. Csalogany, T. Sarlos, and M. Uher. Spamrank - fully automatic link spam detection. In *Workshop on Advers. Inf. Retrieval on the Web*, 2005.

[5] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: Web spam detection using the web topology. Technical report, Yahoo! Research Barcelona, Nov. 2006.

[6] B. Davison. Recognizing nepotistic links on the web. In *Workshop on Artificial Intelligence for Web Search*, 2000.

[7] I. Dorst and T. Scheffer. Thwarting the nigritude ultramarine: Learning to identify link spam. In *Proc. European Conf. on Machine Learning*, 2005.

[8] Q. Gan and T. Suel. Improving web spam classifiers using link structure. In *AIRWeb '07: Procc of the 3rd Int. Workshop on Advers. Inf. Retrieval on the Web*, pages 17–20. ACM, 2007.

[9] Z. Gyongyi, P. Berkhin, H. Garcia-Molina, and J. Pedersen. Link spam detection based on mass estimation. In *Proc. 32nd VLDB*, pages 439–450, 2006.

[10] Z. Gyongyi and H. Garcia-Molina. Link spam alliances. In *Proc. 31st VLDB*, pages 517–528, 2005.

[11] Z. Gyongyi and H. Garcia-Molina. Web spam taxonomy. In *Workshop on Advers. Inf. Retrieval on the Web*, 2005.

[12] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proc. 30th VLDB*, 2004.

[13] M. A. Hearst. Multi-paragraph segmentation of expository text. In *Proc. of the 32nd Annual Meeting of the Assoc. for Computational Linguistics*, pages 9–16, 1994.

[14] M. A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64, 1997.

[15] J. Leskovec, S. Dumais, and E. Horvitz. Web projections: Learning from contextual subgraphs of the web. In *WWW '07: Proc. of the 16th Int Conf. on World Wide Web*, pages 471–480. ACM, 2007.

[16] G. Mishne, D. Carmel, and R. Lempel. Blocking blog spam with language model disagreement. In *Proc. of the 1st Int. Workshop on Adversarial Information Retrieval on the Web*, pages 1–6, 2005.

[17] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proc. 15th WWW*, pages 83–92, 2006.

[18] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *Proc. of the 1st Int. Conf. on Scalable Information Systems*, 2006.

[19] B. Wu and B. Davison. Identifying link farm spam pages. In *Proc. 14th WWW*, May 2005.

[20] B. Wu, V. Goel, and B. Davison. Propagating trust and distrust to demote Web spam. In *Workshop on Models of Trust and the Web*, 2006.