Thwarting the Nigritude Ultramarine: Learning to Identify Link Spam*

Isabel Drost and Tobias Scheffer

Humboldt-Universität zu Berlin, Department of Computer Science
Unter den Linden 6, 10099 Berlin, Germany
{drost, scheffer}@informatik.hu-berlin.de

Abstract. The page rank of a commercial web site has an enormous economic impact because it directly influences the number of potential customers that find the site as a highly ranked search engine result. *Link spamming* – inflating the page rank of a target page by artificially creating many referring pages – has therefore become a common practice. In order to maintain the quality of their search results, search engine providers try to oppose efforts that decorrelate page rank and relevance and maintain blacklists of spamming pages while spammers, at the same time, try to camouflage their spam pages. We formulate the problem of identifying link spam and discuss a methodology for generating training data. Experiments reveal the effectiveness of classes of intrinsic and relational attributes and shed light on the robustness of classifiers against obfuscation of attributes by an adversarial spammer. We identify open research problems related to web spam.

1 Introduction

Search engines combine the similarity between query and page content with the page rank [18] of candidate pages to rank their results. Intuitively, every web page "creates" a small quantity of page rank, collects additional rank via inbound hyperlinks, and propagates its total rank via its outbound links. A web page that is referred to by many highly ranked pages thus becomes more likely to be returned in response to a search engine query.

The success of commercial web sites crucially depends on the number of visitors that find the site while searching for a particular product. Because of the enormous commercial impact of a high page rank, an entire new business sector – *search engine optimization* – is rapidly developing. Search engine optimizers offer a service that is referred to as *link spamming*: they create *link farms*, arrays of densely linked web pages that refer to the target page, thus inflating its page rank. In 2004, the search engine optimization industry tournamented in the "DarkBlue SEO Challenge". The goal of this competition was to be ranked first on Google for the query "nigritude ultramarine", a nonsense term that used to produce zero hits prior to the challenge and produced over 500,000 hits by the

* *Proceedings of the European Conference on Machine Learning.* 2005

competition deadline. Industry insiders believe that as many as 75 million out of the 150 million web servers that are online today may be operated with the sole purpose of increasing the page rank of their target sites.

As the page rank becomes subject to manipulation, it loses its correlation to the true relevance of a web page. This deteriorates the quality of search engine results. Search engine companies maintain blacklists of link spamming pages, but they fight an uneven battle in which humans identify and penalize spamming pages and software tools automatically create new spamming domains, and camouflage them, for instance by filling in inconspicuous content.

We formulate and analyze the problem of link spam identification, present a method for generation of labeled examples, and discuss intrinsic and relational features. Experiments with recursive feature elimination shed light on the relevance of several classes of attributes and their contribution to the classifier's robustness against adversarial obfuscation of discriminating properties.

The rest of the paper is organized as follows. We discuss related work in Section 2 and introduce our problem setting in Section 3. Section 4 introduces the features and employed methods. Section 5 details our experimental results. We discuss open research problems in Section 6 and conclude in Section 7.

2 Related Work

Henzinger [15] refers to automatic identification of link spam as one of the most important challenges for search engines. Davison [8] studies the problem of recognizing "nepotistic" links. A decision tree experiment using features that refer to URL, IP, content, and some linkage properties indicates a number of relevant features. Classifying links imposes the effort of labeling individual links, not just entire pages, on the user. Lempel et al. [17] use similar features to classify web pages into business, university, and other general classes.

Fetterly et al. [11] analyze the distribution of many web page features over 429 million pages. They find that many outliers in several features are link spam. Pages that change frequently and clusters of near-identical pages are also more likely to be spam [12, 10]. They conclude that experiments should be conducted to reveal whether these features can be used by a machine learning method.

Similarly, Broder et al. [5] and Bharat et al. [3] analyze large amounts of web pages and observe that the in-degree and out-degree of web pages that *should be* governed by Zipf's law, empirically deviates from this distribution. They find that "artificially" generated link farms distort the distribution.

The TrustRank approach [14] propagates trust weights along the hyperlinks. But while page rank is generated by every web page (including link farms), trust rank is generated only by manually selected trusted web pages. The manual selection of trusted pages, however, creates a perceptive bias as unknown and remote websites become less visible. Wu and Davison [19] study a simple heutistic that discovers some link farms: pages which have many inlinks and outlinks whose domains match are collected as candidate pages. They use a paopagation

strategy, and remove edges between likely link farms to compute a page rank that is less prone to manipulation.

3 Problem Description

The correlation between page rank and relevance of a web page is based on the assumption that each hyperlink expresses a vote for the relevance of a site. The PageRank algorithm [18] calculates the rank (Equation 1) of a page y that consists of a small constant amount (d corresponds to the probability that a surfer follows a link rather than restarting at a random site, N is the number of web pages), plus the accumulated rank of all referring pages x.

$$R(y) = \frac{1-d}{N} + d \sum_{x \to y} \frac{R(x)}{outlinks(x)} \tag{1}$$

Link spamming is referred to as any intentional manipulation of the page rank of a web page. Pages created for this purpose are called *link spam*. Artificially generated arrays of web pages violate the assumption that links are independent votes of confidence, and therefore decorrelate page rank and relevance.

Several common techniques inflate the page rank of a target site. They are often combined. We will describe these techniques briefly. A more detailed taxonomy of web spam techniques is presented by Gyöngyi [13].

Link farms are densely connected arrays of pages. A target page "harvests" the constant amount of page rank that each page creates and propagates through its links. Each page within the link farm has to be connected to the central, strongly connected component of the web. The farming pages have to propagate their page rank to the target. This can for instance be achieved by a linear or funnel-shaped architecture in which the majority of links points directly or indirectly towards the target page. In order to camouflage link farms, tools fill in inconspicuous content, for instance, by copying news bulletins.

Link exchange services create listings of (often unrelated) hyperlinks. In order to be listed, businesses have to provide a back link that enhances the page rank of the exchange service and, in most cases, pay a fee.

Guestbooks, discussion boards, and weblogs ("blogs") allow readers to create HTML comments via a web interface. Automatic tools post large amounts of messages to many such boards, each message contains a hyperlink to the target website.

In order to maintain a tight coupling between page rank and relevance, it is necessary to eliminate the influence that link spam has on the page rank. Search engines maintain a blacklist of spamming pages. It is believed that Google employs the *BadRank* algorithm. The "bad rank" (Equation 2) is initialized to a high value $E(x)$ for blacklisted pages. It propagates bad rank to all *referring* pages (with a damping factor) thus penalizing pages that refer to spam.

$$BR(x) = E(x)(1-d) + d \sum_{x \to y} \frac{BR(y)}{inlinks(y)} \quad (2)$$

This method, however, is not suited to automatically detect newly created link farms that do not include links to an older, already blacklisted site. Search engines therefore rely on manual detection of new link spam. We focus on the problem of automatically identifying link spam. We seek to construct a classifier that receives a URL as input and decides whether the encoded page is created for the sole purpose of manipulating the page rank of some page ("spam"), or whether it is a regular web page created for a different purpose ("ham").

The typical application scenario of a link spam identification method would be as follows. Search engines interleave crawling and page rank updating. After crawling a web page, the page's impact on the page rank of referenced pages is updated – the BadRank and blacklist status of the page are considered at this point. In addition, the output of a spam classifier for the crawled page can now be taken into account. Depending on the output, the page's impact on the page rank can be reduced, or the page can be disregarded entirely.

Since the class distribution of spam versus ham is not known, we use the area under the ROC curve (AUC) as evaluation metric. The AUC equals the probability that a randomly drawn spam page receives a higher decision function value than a random ham page; the AUC is invariant of the class prior.

Link spam identification is an *adversarial classification problem* [7]. Spammers will probe any filter that is in effect and manipulate the properties of their generated pages in order to dodge the classifier. Therefore, a high classification accuracy is not sufficient to imply its practical usefulness. The practical benefit of a classifier is furthermore dependent of its robustness against purposeful obfuscation of attributes by an adversary. We study how the performance of classifiers deteriorates as an increasing number of attributes becomes obfuscated as the corresponding properties of spam pages are adapted to those of ham pages.

4 Representing and Obtaining Examples

In this section, we address the issues of representing instances and obtaining training examples. We describe our publicly available link spam data set.

Table 1 provides an overview on the features that we use in our experiments to represent an instance x_0. Many of these features are reimplementations of, or have been inspired by, features suggested by [8] and [11]. Notably, however, the tfidf representation of the page, and also other features including the MD5 hash features, have not been studied for web spam problems before.

Figure 1 illustrates the neighborhood of the pivotal page x_0 represented by intrinsic and relational properties. Most elementarily, the tfidf (term frequency, inverse document frequency) representation of the page content provides an intrinsic representation. It creates one dimension in the feature vector for each word in the corpus. It gives large weights to terms that are frequent in the document but infrequent in the whole document corpus.

Table 1. Attributes of web page x_0.

Textual content of the page x_0; tfidf vector.
The following features for x_0 are computed (1) for the pivotal page itself ($X = \{x_0\}$). Here, no aggregation is necessary. (2) For the predecessors $X = pred(x_0)$. The attributes are aggregated over the elements of X using aggregation functions sum and average. (3) For the successors ($X = succ(x_0)$). Aggregation functions sum and average. Boolean features are aggregated by treating "true" as 1, and "false" as 0.
Number of tokens in keyword meta-tag, aggregated over all pages $x \in X$.
Number of tokens in title, aggregated over all pages $x \in X$.
Number of tokens in description meta-tag, aggregated over all pages $x \in X$.
Is the page a redirection? Aggregated over all pages $x \in X$.
Number of inlinks of x, aggregated over all pages $x \in X$.
Number of outlinks of x, aggregated over all pages $x \in X$.
Number of characters in the URL of x, aggregated over all pages $x \in X$.
Number of characters in the domain name of x, aggregated over all pages $x \in X$.
Number of subdomains in the URL of x, aggregated over all pages $x \in X$.
Page length of x, aggregated over all pages $x \in X$.
Domain ending ".edu" or ".org"? Aggregated over all pages $x \in X$
Domain ending ".com" or ".biz"? Aggregated over all pages $x \in X$.
URL contains tilde? Aggregated over all pages $x \in X$.
The following block of context similarity features are calculated (1) for predecessors ($X = pred(x_0)$) and (2) for the successors ($X = succ(x_0)$); sum and ratio are used to aggregate the features over all elements of X.
Clustering coefficient of X; sum and ratio of elements of X with links between them.
Elements of X with the same IP address as some other element of X; sum and ratio.
Elements of X that have the same length as some other element of X; sum and ratio.
Pages that are referred to in x_0 and also in an elements of X, sum and ratio.
Pages referred to from an element of X that are also referred to from another element of X; sum and ratio.
Pages in X that have the same MD5 hash code as some other element of X. This value is high if X contains many identical pages; sum and ratio.
Elements of X that have the same IP as x_0.
Elements of X that have the same length as x_0.
Pages in X that have the same MD5 hash code as x_0.

The next block of attributes is determined for the page x_0 itself as well as for its predecessors $pred(x_0)$ and successors $succ(x_0)$. The features are determined for every element $x \in pred(x_0)$ (or $x \in succ(x_0)$, respectively) and aggregated over all elements x. Both, summation and averaging are used as aggregation functions. The third block quantifies collective features of x_0 and its predecessors, and x_0 and its successors. Some of them require further elaboration.

Following [9], we define the clustering coefficient of a set X of web pages as the number of linked pairs divided by the number of all possible pairs, $|X|(|X|-1)$. The clustering coefficient is 1 if all elements in X are mutually linked and 0 if

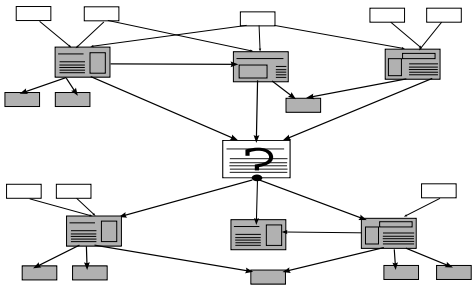

Fig. 1. Neighborhood of page x_0 to classify. The tfidf vector is calculated for x_0; intrinsic features (second block of Table 1) are calculated for x_0, $pred(x_0)$, and $succ(x_0)$; context similarity features (third block of Table 1) for $pred(x_0)$ and $succ(x_0)$.

no links between elements of X exist. The MD5 hash is frequently used as a mechanism for digital signatures. It maps a text to a code word of 128 bit such that collisions are "unlikely". That is, when two documents have the same MD5 hash code, then it is unlikely that they differ. The MD5 features quantify the number and ratio of predecessors and successors of x_0 that are textually equal.

In total, each page is represented by 89 features plus its tfidf vector. Web pages explicitly contain outbound links but, of course, not their inbound links. To be able to determine the feature vector of a given page x_0, it is necessary to crawl its direct successors with standard web crawling tools (we use nutch [6] for our experiments). Crawling backwards (moving from a page to its references) can be achieved by specific search engine query tokens (*e.g.,* "link:" in Google).

4.1 Crawling Examples

Training a classifier requires labeled samples. Deciding whether an example web page is link spam requires human judgment, but we can exploit efforts already exercised by other persons. The Dmoz open directory project is a taxonomy of web pages that covers virtually all topics on the web. All entries have been reviewed by volunteers. While the Dmoz entries do contain pages whose rank is being inflated by link farms (*e.g.,* online casinos) the listed pages themselves contain valid content and there are no instances of spam. We create examples of "ham" (non-link-spam) by drawing 854 Dmoz entries at random.

Search engine operators maintain blacklists of spamming pages. They could, if public, provide a rich supply of examples of spam. In order to investigate their distinct characteristics, we differentiate between guestbook spam and a second class of spam that includes link farms and link exchange sites, and generate distinct samples for these sets. We obtain 251 examples of guestbook spam by drawing URLs from a publicly available manually edited blacklist that aims at helping guestbook and weblog operators to identify and remove spam entries.

Link farms are never returned as a highly ranked search result – this is not their goal – but they promote the ranking of their *target site*. After posting 36 search queries (e.g. 'gift links", "shopping links", "dvd links", "wedding links",

"seo links", "pharmacy links", "viagra", and "casino links"), we draw some of the top-rated search results. We manually identify 180 link farm and link exchange pages. In order to correctly decide which pages are link spam, we review each page's content and context (referring and linked pages). This careful manual labeling consumed the largest part of the effort of assembling the data set. We do not distinguish between link exchange and link farm pages because this distinction cannot easily be made for many of the examples.

5 Experimental Evaluation

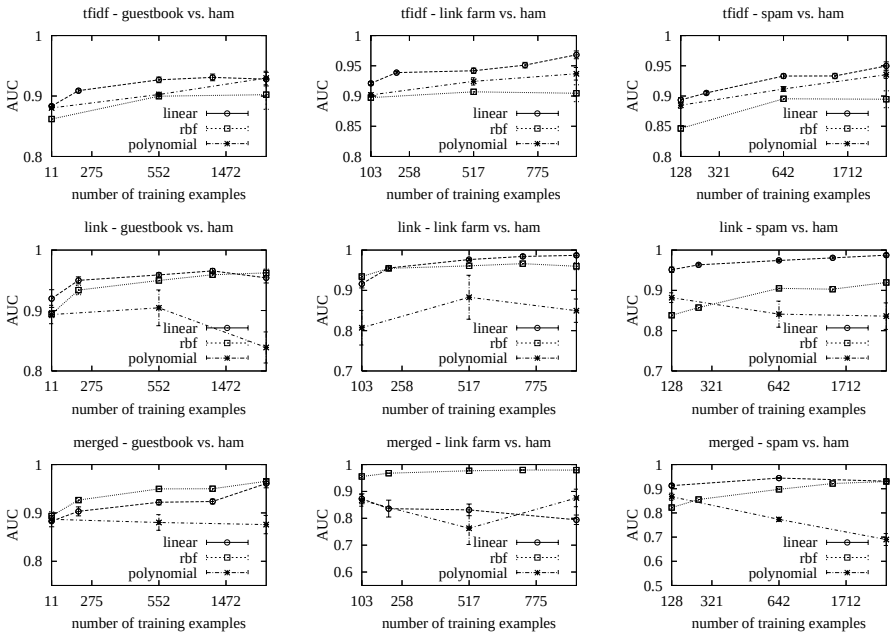

Fig. 2. Comparison of kernels.

In our experiments, we want to explore how well guestbook spam and link farms can be discriminated against regular web pages, and which features contribute the most to a discrimination. In addition we are interested in the robustness of classifiers against obfuscation of attributes by an adversarial spammer who purposefully adapts properties of spam pages to those of ham pages. We use the Support Vector Machine SVM^{light} [16] with standard parameters. We would also like to find out which kernel is appropriate.

5.1 Finding Discriminative Features

In the following experiments we discriminate ham versus spam, where spam is the union of all categories discussed in Section 3. In order to investigate possible differences between guestbook spam and link farms, we furthermore discriminate guestbook spam against ham, and link farm plus link exchange pages versus ham. In order to obtain learning curves, we randomly draw a training subset of specified size from the sample, use the remaining examples for testing, and average the results over 10 randomly resampled iterations.

We first study the suitability of several kernels for three instance representations: we consider the tfidf representation, a representation based only on link based features (Table 1), and the joint attribute vectors of concatenated tfidf and link features. We tune the degree (for polynomial kernels), the kernel width for RBF kernels (on a separate tuning set) and use default parameters otherwise. Figure 2 shows that, on average, linear kernels perform best for the tfidf vectors and link features. Both, RBF and linear kernels work for the combined representation; polynomial kernels perform poorly for the link and combined representation.

Next, we study learning curves for the tfidf representation, the attributes of Table 1, and the joint features. Figure 3 shows that link and merged representations work equally well for guestbook spam, the merged representation performs best for link farms. For the mixed spam dataset, tfidf is the most discriminative feature set, but the offset to the combined representation is small. Using the combined represantation is always better than using only the link based features.

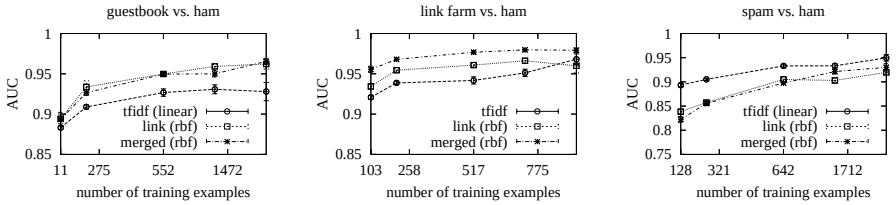

Fig. 3. Comparison of feature representations.

Which features are discriminative? We use 10 resampled iterations of recursive feature elimination and determine the average rank of all features. The recursive feature elimination procedure initialy uses an active feature set containing all features. It then trains a Support Vector Machine, eliminates the feature with the least weight from the active set, and recurs. The best (highest ranked) features are those that remain in the active set longest.

Table 2 presents the 20 highest ranked link features for the "spam vs. ham" discrimination. We do not rank tfidf features for individual words. Treated as a whole, the entire block of tfidf features has the highest rank. Among the most discriminative link features are the number of inbound and outbound links of $pred(x_0)$, the title length of $succ(x_0)$; also, clustering coefficient and the MD5 feature are relevant.

Table 2. RFE ranking for spam vs. ham

Sign	Rank	Attribute
-	1	Average number of inlinks of pages in $pred(x_0)$.
+	2	Average number of tokens in title of pages in $succ(x_0)$.
+	3	Number of elements of $pred(x_0)$ that have the same length as some other element of $pred(x_0)$.
-	4	Average number of in- and outlinks of pages in $pred(x_0)$.
+	5	Average number of outlinks of pages in $pred(x_0)$.
+	6	Number of tokens in title of x_0.
-	7	Summed number of outlinks of pages in $succ(x_0)$.
-	8	Summed number of inlinks of pages in $pred(x_0)$.
+	9	Clustering coefficient of pages in $pred(x_0)$.
+	10	Summed number of tokens in title of pages in $succ(x_0)$.
+	11	Number of outlinks of pages in $succ(x_0)$.
+	12	Average number of characters of URLs of pages in $pred(x_0)$.
-	13	Number of pages in $pred(x_0)$ and $succ(x_0)$ with same MD5 hash as x_0.
-	14	Number of characters in domain name of (x_0).
-	15	Number of pages in $pred(x_0)$ with same IP as x_0.
+	16	Average number of characters in domain name of pages in $succ(x_0)$.
-	17	Average Number of in- and outlinks of pages in $succ(x_0)$.
+	18	Number of elements of $succ(x_0)$ that have the same length as some other element of $succ(x_0)$.
+	19	Average number of in- and outlinks of elements in $succ(x_0)$.
+	20	Number of pages in $succ(x_0)$ with same IP as x_0.

5.2 Robustness against Adversarial Obfuscation

The previous experiments show that *today* the tfidf representation is the most discriminative feature set. Once a spam filter is in effect, spammers can be expected to probe the filter and to modify properties of their link farms in order to deceive the filter. Any single feature of a set of web pages can, at some cost, be adapted to the distribution which governs that attribute in ham pages. For instance, the tfidf feature can be obfuscated by copying the content of a randomly drawn ham page from the Dmoz directory, the number of inbound or outbound links can be decreased by splitting pages or increased by merging pages.

We consider the following classifiers and study their robustness against such obfuscations of discriminative attributes.

1. A *purely text-based* classifier which refers to only the tfidf features.
2. A *simple contextual* classifier uses the set of intrinsic features of the pivotal page x_0 with aggregation function identity.
3. The *complex contextual classifier* utilizes all features.

We use the following experimental protocol to assess the three classifiers' robustness. We first train the classifiers using their respective attribute sets. Now we simulate the adversary's attempts to deceive the classifiers. The adversary obfuscates an increasing number of attributes, starting with the entire block of tfidf features (this can be achieved by pasting the textual content of a "ham"

page into the spam page), and subsequently obfuscating additional attributes in the order of their discriminatory power (Table 2). The obfuscation of attributes is simulated as follows. For each instance, the value of the obfuscated attributes is replaced by the attribute value of a randomly drawn ham page. This simulates that the adversary has adapted the generation program to match some property of the link farm pages to that of natural web pages. We evaluate the performance of the four classifiers on the obfuscated test sets.

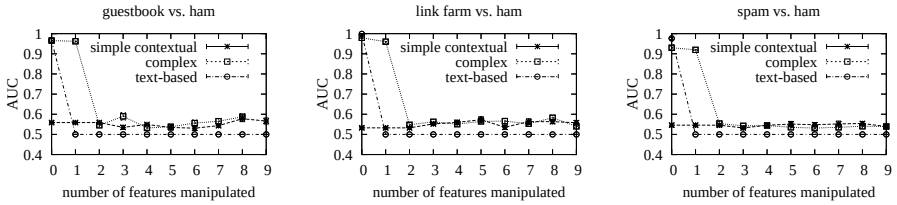

Fig. 4. Influence of attribute obfuscation

Figure 4 shows that the purely text-based classifier is immediately rendered useless when the content of the spam pages is replaced by the content of a randomly drawn page from the Dmoz directory. The combined classifier that utilizes multiple features deteriorates slightly slower. However, Figure 4 emphasizes the need to re-train a classifier quickly when the underlying distribution changes.

6 Open Problems

The problem area of link spam contains a plentitude of open research questions. We summarize some of them.

Collective Classification. Rather than classifying individual web pages, search engine operators will have to classify all pages on the web. Hence, the problem intrinsically is a collective classification problem. A benchmark collective classification data set has to include a network of example instances with a reasonable degree of context. Because of the small world property of the web, such a dataset quickly becomes very large.

Game Theory. Link spam identification is an adversarial classification problem [7]. Rather than being stationary, the distribution of instances is changed by an adversary over time. The adversary probes any link spam filter that is being used by a search engine, and modifies properties of the generated link spam such as to dodge the filtering algorithm. Possible modifications include, for instance, changing the link topology, generating pages with differing content and experimenting with various sub-domain names. The topic of learning the ranking function of a search engine from rankings and page features is adressed by Bifet et al. [4]. Yet, identifying conditions under which a filtering strategy can be shown to be an equilibrium of the "spam filtering game" is a great challenge.

Identifying "Google Bombers". Search engines associate the anchor text that is used to refer to a page with that page. By referring to target pages with anchor terms that have a negative connotation, malicious sites cause these targets to become search results for negative query terms. This form of web spam is often referred to as "Google bombing". For instance the query "miserable failure" usually returns the CVs of George W. Bush or Michael Moore as the first result, depending on whose supporters are currently heading this particular Google bombing arms race. Adali et. al [1] study the layout of an optimal link bomb. The influence of more general collusion topologies on page rank is examined by Baeza-Yates et al. [2]. But the development of methods that decide whether a reference is unbiased or malicious is still an open research goal.

Other Forms of Web Spam. *Click spamming* is a particularly vicious form of web spam. Companies allocate a fixed budged to the sponsored links program of, for instance, Google. The sponsored link is returned along with results of related search queries until the budget is used up. Rivaling companies now employ "click bots" that post a search query, and then automatically click on their competitor's sponsored link until the budged is exceeded and the link disappears. This practice undermines the benefit of the sponsored link program, and search engine operators therefore have to identify whether a reference to a sponsored link has been made by a human, or by a "rogue bot". This classification task is extremely challenging because the state-less HTTP protocol provides hardly any information about the client.

7 Conclusion

We motivated and introduced the problem of link farm discovery. We discussed intrinsic and contextual features of web pages, and presented a methodology for collecting training examples. Our experiments show that today the tfidf representation of the page provides the most discriminatory attribute set. Many additional contextual attributes contribute to a more accurate discrimination. We identify the most discriminatory relational attributes.

Our experiments also show that a purely text-based classifier is brittle and can easily be deceived; the contextual classifiers have the potential to be more robust because deceiving them requires to adapt a larger number of properties to the distribution of values observed in ham pages. In order to be able to react to purposeful obfuscation of characteristic properties of link farms, a repository of discriminatory features is required.

Web spam is a major challenge for search engines. We sketched open research challenges; research in this direction has the potential to substantially improve search engine technology and make it more robust to manipulations.

Acknowledgment

This work has been supported by the German Science Foundation DFG under grant SCHE 540/10-1.

References

1. S. Adali, T. Liu, and M. Magdon-Ismail. Optimal link bombs are uncoordinated. In *Proc. of the Workshop on Adversarial IR on the Web*, 2005.
2. R. Baeza-Yates, C. Castillo, and V. López. Pagerank increase under different collusion topologies. In *Proc. of the Workshop on Adversarial IR on the Web*, 2005.
3. K. Bharat, B. Chang, M. Henzinger, and M. Ruhl. Who links to whom: Mining linkage between web sites. In *Proc. of the IEEE International Conference on Data Mining*, 2001.
4. A. Bifet, C. Castillo, P.-A. Chirita, and I. Weber. An analysis of factors used in search engine ranking. In *Proc. of the Workshop on Adversarial IR on the Web*, 2005.
5. A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. In *Proc. of the International WWW Conference*, 2000.
6. M. Cafarella and D. Cutting. Building Nutch: Open source search. *ACM Queue*, 2(2), 2004.
7. N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. Adversarial classification. In *Proc. of the ACM International Conference on Knowledge Discovery and Data Mining*, 2004.
8. B. Davison. Recognizing nepotistic links on the web, 2000. In Proceedings of the AAAI-2000 Workshop on Artificial Intelligence for Web Search.
9. Holger Ebel, Lutz-Ingo Mielsch, and Stefan Bornholdt. Scale free topology of e-mail networks. *Physical Review E*, 2002.
10. D. Fetterly, M. Manasse, and M. Najork. On the evolution of clusters of near-duplicate web pages. In *Proc. of the Latin American Web Congress*, 2003.
11. D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proc. of the International Workshop on the Web and Databases*, 2004.
12. D. Fetterly, M. Manasse, M. Najork, and J. Wiener. A large-scale study of the evolution of web pages. In *Proc. of the International WWW Conference*, 2003.
13. Zoltn Gyngyi and Hector Garcia. Web spam taxonomy. In *Proc. of the Workshop on Adversarial IR on the Web*, 2005.
14. Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *Proc. of the International Conf. on Very Large Data Bases*, 2004.
15. M. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. In *Proc. of the International Joint Conference on Artificial Intelligence*, 2003.
16. T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods – Support Vector Learning*. MIT Press, 1998.
17. R. Lempel, E. Amitay, D. Carmel, A. Darlow, and A. Soffer. The connectivity sonar: Detecting site functionality by structural patterns. *Journal of Digital Information*, 4(3), 2003.
18. L. Page and S. Brin. The anatomy of a large-scale hypertextual web search engine. In *Proc. of the Seventh International World-Wide Web Conference*, 1998.
19. Baoning Wu and Brian D. Davison. Identifying link farm spam pages. In *Proc. of the 14th International WWW Conference*, 2005.