Using Propagation Of Distrust
to find Untrustworthy Web Neighborhoods

Panagiotis Metaxas
Computer Science Department
Wellesley College
106 Central Street, Wellesley, MA 02481, USA
pmetaxas@wellesley.edu

Abstract

Web spamming, the practice of introducing artificial text and links into web pages to affect the results of searches, has been recognized as a major problem for search engines. But it is mainly a serious problem for web users because they tend to confuse trusting the search engine with trusting the results of a search.

In this paper, we propose "backwards propagation of distrust," as an approach to finding spamming untrustworthy sites. Our approach is inspired by the social behavior associated with distrust. In society, recognition of an untrustworthy entity (person, institution, idea, etc) is a reason for questioning the trustworthiness of those that recommended tis entity. People that are found to strongly support untrustworthy entities become untrustworthy themselves. So, in society distrust is propagated backwards.

Our algorithm simulates this social behavior on the web graph with considerable success. Moreover, by respecting the user's perception of trust through the web graph, our algorithm makes it possible to resolve the moral question of who should be making the decision of weeding out untrustworthy spammers in favor of the user, not the search engine or some higher authority. Our approach can lead to browser-level, or personalized server-side, web filters that work in synergy with the powerful search engines to deliver personalized, trusted web results.

1. Introduction

The web has changed the way we inform and get informed. Every organization has a web site and people are increasingly comfortable accessing it regarding any question they may have. The exploding size of the web necessitated the development of search engines and most people with online access use a search engine to get informed and make decisions that may have medical, financial, cultural, political, security or other important implications in their lives [16, 3, 9, 11]. However, 85% of the time, people do not look past the top 10 results returned by the search engine [14]. Given this, it is not surprising that anyone with a web presence struggles for a place in the top ten positions of relevant web search results. The importance of the top-10 placement has given birth to a new industry, the Search Engine Optimization (SEO) industry, which sells know-how for prominent placement in search results. It includes companies, publications, and even conferences. Some of the SEO's are willing to bend the truth in order to fool the search engines and their customers, by creating web spam [4].

Web spam is often defined as the practice of manipulating web pages in order to cause search engines to rank some web pages higher than they would without any manipulation. Spammers aim at search engines, but target the end users. Their motive is usually commercial, but can also be political, or religious.

Spammers attack search engines through text and link spam. **Text spam** includes repeating text excessively and/or adding irrelevant text on the page that will cause incorrect calculation of page relevance; adding misleading meta-keywords or irrelevant "anchor text" that will cause incorrect application of rank heuristics. **Link spam** aims to change the perceived structure of the webgraph in order to cause incorrect calculation of page reputation. Such examples are the so-called "link-farms," page "awards,"[1] domain flooding (plethora of domains that re-direct to a target site), etc.

Both kinds of spam aim to boost the ranking of spammed web pages. So as not to get caught, spammers conceal

[1] With this technique, the spammer pretends to run an organization that distributes awards for web site design or information. The awarded site gets to display the "award", an image linking back to awarding organization. The effect is that the awarded site increases the visibility of the spammer' site.

their actions through cloacking, content hiding and redirection. Cloaking, for example, aims to serve different pages to search engine robots and to web browsers (users). These pages could be created statically or dynamically. Static pages, for example, may employ hidden links and/or hidden text with colors or small font sizes noticeable by a crawler but not by a human. Dynamic pages might change content on the fly depending on the visitor, serving different content to web crawlers and to web browsers. For a extensive treatment of the known spamming techniques, see [5].

One of the reasons behind the users' difficulty to distinguish trustworthy from untrustworthy information comes from the success that both search engines and spammers have enjoyed in the last decade. Users have come to trust search engines as a means of finding information, and spammers have successfully managed to transfer that trust to the results of each search they are able to influence.

From their side, the search engines have put considerable effort in delivering spam-free query results and have developed sophisticated ranking strategies. Two such ranking strategies that have received major attention are the well-known PageRank [2] and HITS [10] algorithms. Achieving high PageRank has become a sort of obsession for many companies' IT departments, and the *raison d'être* of spamming companies. Some estimates indicate that at least 13.8% of all English-language pages indexed is spam [13] while experts consider web spamming the single most difficult challenge web searching is facing today[8]. Search engines typically see web spam as an interference to their operations and would like to restrict it, but there can be no algorithm that can recognize spamming sites based solely on graph isomorphism [1].

To address the problem, however, we need to understand *why* spamming works beyond the *how*, because spamming is a social problem first, then a technical one. [12] analyzes web spam's extensive relationship to social propaganda, and provides evidence of its influence on the evolution of search engines. In this paper we describe and evaluate an algorithmic way of discovering spamming networks automatically. In addition, we discuss a general framework for the long-term approach to web spam.

Web spamming has received a lot of attention in the last decade. Characteristics of spamming sites based on diversion from power laws are presented in [4]. An analysis of the popular PageRank method employed by most search engines today and ways to maximize it in a spamming network is described in [1]. TrustRank, a modification to the PageRank to take into account the evaluations of a few seed pages by human editors, employees of a search engine, is presented in [6]. Techniques for identifying link farms of spam pages were also presented in [18]. Recently, [13] have introduced heuristics to detect spam through statistical content analysis, while [17] have devised methods to deal with redirection spam.

It should be noted that in most of the anti-spamming efforts so far, the need for automatic methods of spam detection has focused on synthetic pages, that is, pages that are created massively using simple text concatenation extracted from a dictionary or online text [13]. In this paper we are looking at more "intelligent" spam that is designed to evade automatic detection based on statistical analysis. Specifically, we are looking to automatically uncover untrustworthy web neighborhoods that use link spamming techniques.

The rest of this paper is organized as follows. The next section describes the backward propagation of distrust algorithm and the following section presents some of our experimental results running this algorithm. The final section has the conclusions and discussion of future directions of this work.

2. An Anti-propagandistic Method

Since spammers employ propagandistic techniques [12], it makes sense to design anti-propagandistic methods for defending against them. These methods need to be user-initiated, that is, the user decides which web site not to trust and then seeks to distrust those supporting the untrustworthy web site. We are considering trustworthiness to be a personal decision, not an absolute quality of a site. One person's gospel is another's political propaganda, and our goal is to design methods that help individuals make more informed decisions about the quality of the information they find on the web.

Here is one way that people defend against propaganda in every day life:

In society, distrust is propagated backwards: When an untrustworthy recommendation is detected, it gives us a reason to reconsider the trustworthiness of the recommender. Recommenders who strongly support an untrustworthy recommendation become untrustworthy themselves.

This process is selectively repeated a few times, propagating the distrust backwards to those who strongly support the recommendation. The results of this process become part of our belief system and are used to filter future information. (Note that distrust is not propagated forward: An untrustworthy person's recommendations could be towards *any* entity, either trustworthy or untrustworthy.)

We set out to test whether a similar process might work on the web. Our algorithm takes as input s, a web site, which is represented by the URL of the server containing a page that the user determined to be untrustworthy. This page could have come to the user through web search results, an email spam, or via the suggestion of some trusted associate (e.g., a society that the user belongs to).

The obvious challenge in testing this hypothesis would be to retrieve a neighborhood of web sites linking to the starting site s in order to analyze it. Since we are interested in back links to sites, we can not just follow a few forward links (hyperlinks on web sites) to get this information. Otherwise we would need to possibly explore the whole web graph. Today, only search engines have this ability. Thankfully, search engines have provided APIs to help with our task.

Starting from s we build a breadth-first search (BFS) tree of the sites that link to s within a few "clicks" (Figure 1). We call the directed graph that is revealed by the backlinks, the "trust neighborhood" of s. We do not explore the web neighborhood directly in this step. Instead, we can use the Google API for retrieving the backlinks.

Referring to Figure 1, if one deems that starting site 1 is untrustworthy, and sites 2, 3, 4, 5 and 6 link directly to it, one has reasons to be suspicious of those sites too. We can take the argument further and examine the trustworthiness of those sites pointing to 2, ... 6. The question arises on whether we should distrust all of the sites in the trust neighborhood of starting site s or not. Is it reasonable to become suspicious of every site linking to s in a few steps? They are "voting in confidence" after all [2, 10]. Should they be penalized for that? Such a radical approach is not what we do in everyday life. Rather, we selectively propagate distrust backwards only to those that most strongly support an untrustworthy recommendation. Thus, we decided to take a conservative approach and examine only those sites that use link spamming techniques in supporting s. In particular, we focused on the biconnected component (BCC) that includes s (Figure 2).

A BCC is a graph that cannot be broken into disconnected pieces by deleting any single vertex. An important characteristic of the BCC is there are at least two independent paths from any of its vertices to s. Strictly speaking, the BCC is computed on the undirected graph of the trust neighborhood. But since the trust neighborhood is generated through the BFS, the cross edges (in BFS terminology) create cycles in the undirected graph (Figure 1). Each cycle found in the BCC must have at least one "ring leader", from which there are two directed paths to s, one leaving through the discovery edge and the other through the cross edge. We view the existence of multiple paths from ring leaders to s as evidence of strong support of s. The BCC reveals the members of this support group. The graph induced by the nodes not in the BCC is called "BFS periphery".

More formally, the algorithm is as follows:

```
Input:
  s = Untrustworthy starting site's URL
  D = Depth of search
  B = Number of backlinks to record

S = {s}
Using BFS for depth D do:
  Compute U={sites linking to sites in S}
    using the Google API
    (up to B backlinks / site)
  Ignore blogs, directories, edu's
  S = S + U
Compute the BCC of S that includes s

Output: The BCC
```

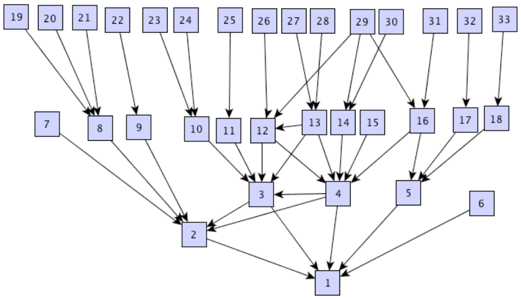

Figure 1. An example of a breadth-first search tree in the trust neighborhood of site 1. Note that some nodes (12, 13, 16 and 29) have multiple paths to site 1. We call these nodes "ring leaders" that show a concerted effort to support site 1.

2.1. Implementation Details

To be able to implement the above algorithm at the browser side, we restrict the following parameters: First, the BFS's depth D is set to 3. We are not interested in exploring a large chunk of the web, just a small neighborhood around s. Second, we limit the number B of backlink requests from the Google API to 30 per site. This helps reduce the running time of our algorithm since the most time-consumming step is the query to Google's backlink database. Finally, we introduced in advance a set of "stop sites" that are not to be explored further.

A *stop site* is one that should not be included in the trust neighborhood either because the trustworthiness of such a site is irrelevant, or because it cannot be defined. In the first category we placed URLs of educational institutions (domains ending in .edu). Academicians are not in the business of linking to commercial sites [13]. When they do, they do not often convey trust in the site. College libraries and academicians, for example, sometimes point to untrustworthy sites as examples to help students critically think about information on the web. In the latter category we placed a few well known Directories (URLs ending in yahoo.com, dmoz.org, etc.) and Blog sites (URLs containing the string 'blog' or 'forum'). While blogs may be set up

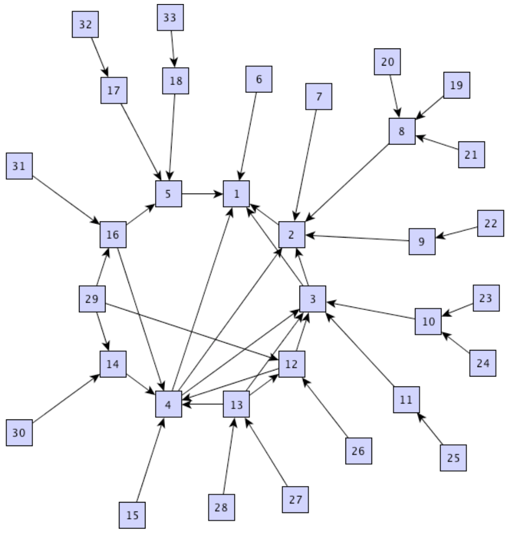

Figure 2. The BCC of the trust neighborhood of site 1 is drawn in a circular fashion for clarity. Note that the BCC contains the "ring leaders," that is, those nodes with multiple paths leading to s. The graph induced by the nodes not in the BCC is called "BFS periphery".

by well meaning people who are trying to increase the discourse on the web, blog pages are populated with opinions of many people and are not meant to represent the opinion of the owner. Anyone can put an entry into an unsupervised blog or directory, and following a hyperlink from a blog page should not convey the trustworthiness of the whole blog site. If the search engines were able to distinguish and ignore links inside the comments, blogs could be removed from the stop sites. No effort to create an exhaustive list of blogs or directories was made.

With these restrictions, our algorithm can be implemented on an average workstation and produce graphs with up to a few hundred nodes within minutes. As we mentioned, the most time demanding step is requesting and receiving the backlink lists from Google, since it requires initiating an online connection. No connections to the particular web sites was done during the creation of the trust neighborhood. Performing the BFS and computing the BCC of the graph assembled is done in time linear on the number of sites retrieved, so it is fast. The whole neighborhood can fit into the main memory of the workstation, so this does not require additional time.

3. Finding Untrustworthy Neighborhoods that use Link Spam

There are several ways one can run into an initial untrustworthy site to use it as a starting site s.. For example, search results for queries that happen to be controversial (e.g., "Armenian genocide", "morality of abortion" and "ADHD real disease") or happen to be the source of unreliable advertisement (e.g., "human growth hormone increase muscle mass"), contain plethora of responses that can be considered untrustworthy. In our experiments, we examined the trust neighborhoods of eight untrustworthy and two trustworthy sites. In Table 1 below these sites are labeled as U-1 to U-8 and T-1 to T-2, respectively.

We run the experiments between September 17 and November 5, 2004. At the time of the experiment, all sites happen to have comparable PageRank, as reported by the Google Toolbar. In fact, U-1 and T-1 both had PageRank 6 while the remaining sites had PageRank 5. We recorded the PageRank numbers as reported by the Google Toolbar because this is always one of the first questions people ask and because the spamming industry seems to use it as a measure of their success. In fact, one can find spam networks inviting the creation of "reciprocal links" for sites that have at lease a minumum of PageRank 5, in order to increase their overal PageRank. numbers.

To determine the trustworthiness of each site we had a human evaluator look at a sample of the sites of the BCC. The results of our experiments appear on Table 1. Due to the significant manual labor involved, only 20% of the total 1,396 BCC sites were sampled and evaluated. To select the sample sites, we employed stratified sampling with skip interval 5. The stratum used was similarity of the site to the starting site.

Each site in the sample was classified as either Trustworthy, Untrustworthy, or Non-determined. The last category includes a variety of sites for which the evaluator could not clearly classify.

We have two main results:

1. THE TRUSTWORTHINESS OF THE STARTING SITE IS A VERY GOOD PREDICTOR FOR THE TRUSTWORTHINESS OF THE BCC SITES.

In fact (see Table 1), there were very few trustworthy sites in the trust neighborhoods of sites U-1 to U-8. The reason is, we believe, that a trustworthy site is unlikely (though not impossible) to deliberately link to an untrustworthy site, or even to a site that associates itself with an untrustworthy one. In other words, *the "vote of confidence" link analogy holds true only for sites that are choosing their links responsibly.* The analogy is not as strong when starting from a trustworthy site, since untrustworthy sites are free to link to whomever they choose. After all, there is some value in portraying a site in good company: Non-critically thinking users may be tempted to conclude that, if a site points to "good" sites, it must be "good" itself.

2. THE BCC IS SIGNIFICANTLY MORE PREDICTIVE OF UNTRUSTWORTHY SITES THAN THE BFS PERIPHERY.

In particular (see Figure 3, top), in the BCC of an un-

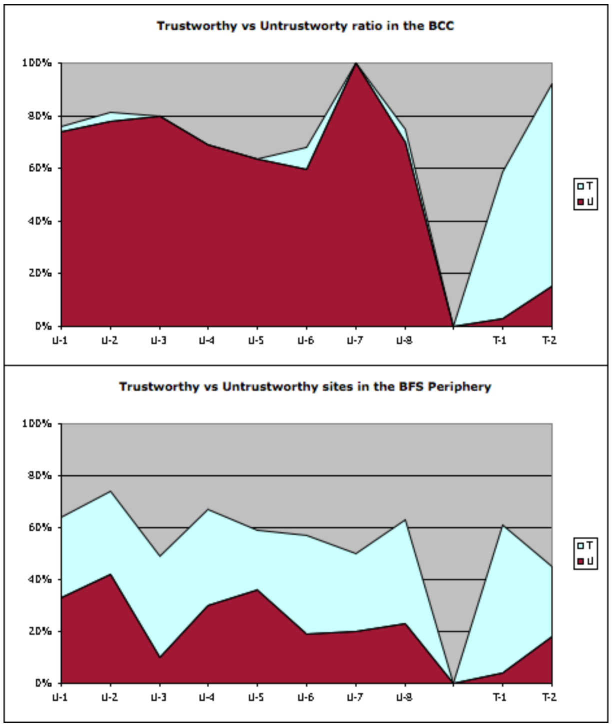

Figure 3. The trustworthy and untrusworthy percentages for trust neighborhoods of the BCC (top) and BFS peripheral (bottom) sites for the data shown in Table 1. Shown are 8 untrustworthy (left) and 2 trustworthy sites (right).

trustworthy starting site, we found that, on average, 74% of the sites were also untrustworthy, while only 9% were trustworthy. In the BFS periphery (see Figure 3, bottom), these average percentages change to 27% untrustworthy and 11% trustworthy, with the rest non-determined. This suggests that the trustworthiness of sites in the BFS periphery is essentially unrelated to the trustworthiness of the starting site.

4. Conclusions

In this paper we present a technique to identify spamming untrustworthy neighborhoods, developed by mimicking anti-propagandistic methods. In particular, we presented automatic ways of recognizing trust neighborhoods on the web based on the biconnected component around some starting site. Experimental results from a number of such instances show our algorithm's ability of recognizing parts of a spamming network.

One of the benefits of our method is that we do not need to explore the web graph explicitly in order to find these neighborhoods, which would be impossible for a client computer. Of course, it would be possible to support a user's trusted and untrusted sites through some personalization service provided by search engines. To be usable and efficient, this service would require the appropriate user interface. When a user encounters an untrustworthy site coming high up in the results of some search query, she would select the item and click on a "Distrust" button. The browser would add this site in the user's untrostworthy site collection and would run the algorithm that propagates distrust backwards. Next time the user runs a similar search query, the untrusted sites would be blocked or demoted. Recently, Google has introduced SearchWiki, a method of supporting personalized opinions about search results [15], which could be adjusted to support this operation.

The algorithm we described is a first step in supporting the trust network of a user. Ultimately, it would be used along with a set of trust certificates that contains the portable trust preferences of the user, a set of preferences that the user can accumulate over time. Organizations that the user joins and trusts may also add to this set. A combination of search engines capable of providing indexed content and structure [7], including identified neighborhoods, with personalized filtering those neighborhoods through the user's trust preferences, would provide a new level of reliability to the user's information gathering. Sharing ranking decisions with the end user will make it much harder for spammers to tune to a single metric – at least as hard as it is for propagandists to reach a large audience with a single trick.

In our experiments we also devised a simple method to evaluate the similarity of the contents of each site to the starting site s. After the trust neighborhood was explored, we fetched and concatenated a few pages from each site (randomly choosing from the links that appeared in the domain URL) into a document. Then, we tried to determine the similarity of each such document to the document of the starting site. Similarity was determined using the $tf.idf$ ranking on the universe of the sites explored. We are aware that having a limited universe of documents does not give the best similarity results, but we wanted to get a feeling of whether our method could further be used to distinguish between "link farms" (spamming sites controlled by a single entity) and "mutual admiration societies" (groups of independent spammers choosing to exchange links). The initial results are encouraging, showing a higher percentage of untrustworthy sites among those most similar to the starting site s.

Several possible extensions can be considered in this work. Generating graphs with more backlinks per site, studying the evolution of trust neighborhoods over time, examining the density of the BCCs, and finding a more reliable way to compute similarity are some of them. We also expect that the results would be strengthened if one consid-

S	$\|V_G\|$	$\|E_G\|$	$\|V_{BCC}\|$	$\|E_{BCC}\|$	**Trust**$_{BCC}$	**Untr**$_{BCC}$	**Trust**$_{BFS}$	**Untr**$_{BFS}$
U-1	1307	1544	228	465	2%	74%	31%	33%
U-2	1380	1716	266	593	4%	78%	32%	42%
U-3	875	985	97	189	0%	80%	39%	10%
U-4	457	509	63	115	0%	69%	37%	30%
U-5	716	807	105	189	0%	64%	23%	36%
U-6	312	850	228	763	9%	60%	38%	19%
U-7	81	191	32	143	0%	100%	30%	20%
U-8	1547	1849	200	430	5%	70%	40%	23%
T-1	1429	1566	164	273	56%	3%	57%	4%
T-2	241	247	13	17	77%	15%	27%	18%

Table 1. Sizes of the explored trust neighborhoods G and their BCC's for eight untrustworthy (U-1 to U-8) and two trustworthy (T-1 and T-2) starting sites. $|V_G|$ contains the number of vertices and $|E_G|$ the number of edges that our algorithm found in the trust neighborhood of starting site s (starting from site s and exploring in BFS mode their backlinks.) Columns $|V_{BCC}|$ and $|E_{BCC}|$ contains the numbers of vertices and edges of the largest biconnected component within G. The next four columns contains the estimated percentages of trustworthy and untrustworthy sites found in the BCCs and the BFS peripheries (respectively). 20% of each BCC and 10% of each BFS periphery were evaluated using stratified sampling.

ers tri- (or higher) connected components of the trust neighborhood. The Google API has been known to be filtering and restricting the number of the backlinks it is reporting but it was the only tool available at the time of this research. Using the Yahoo Search API will likely improve the results we are getting.

References

[1] M. Bianchini, M. Gori, and F. Scarselli. PageRank and web communities. In *Web Intelligence Conference 2003*, Oct. 2003.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[3] T. S. Corey. Catching on-line traders in a web of lies: The perils of internet stock fraud. Ford Marrin Esposito, Witmeyer & Glesser, LLP, May 2001. http://www.fmew.com/archive/lies/.

[4] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics. In *WebDB2004*, June 2004.

[5] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *Proceedings of the AIRWeb Workshop*, May 2005.

[6] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *VLDB 2004*, Aug. 2004.

[7] T. H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the eleventh international conference on World Wide Web*, pages 517–526. ACM Press, 2002.

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

[9] M. Hindman, K. Tsioutsiouliklis, and J. Johnson. Googlearchy: How a few heavily-linked sites dominate politics on the web. In *Annual Meeting of the Midwest Political Science Association*, April 3-6 2003.

[10] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[11] C. A. Lynch. When documents deceive: trust and provenance as new factors for information retrieval in a tangled web. *J. Am. Soc. Inf. Sci. Technol.*, 52(1):12–17, 2001.

[12] P. Metaxas. On the evolution of search engine rankings. In *Proceedings of the 5th WEBIST Conference*, Lisbon, Portugal, March 2009.

[13] A. Ntoulas, D. Fetterly, M. Manasse, and M. Najork. Detecting spam web pages through content analysis. In *WWW 2006*, May 2006.

[14] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[15] The official Google blog. Searchwiki: Make search your own. http://googleblog.blogspot.com/2008/11/searchwiki-make-search-your-own.html, Nov. 20 2008.

[16] The Pew Foundation. Pew internet and american life project. http://www.pewinternet.org, 2008.

[17] Y.-M. Wang, M. Ma, Y. Niu, and H. Chen. Spam double funnel: Connecting web spammers with advertisers. In *WWW 2007*, May 8–12 2007.

[18] B. Wu and B. Davison. Identifying link farm spam pages. In *WWW 2005*, May 2005.