Foundations and Trends® in
Information Retrieval
Vol. 4, No. 5 (2010) 377–486

DOI: 10.1561/1500000021

Adversarial Web Search

By Carlos Castillo and Brian D. Davison

Contents

Foundations and Trends® in
Information Retrieval
Vol. 4, No. 5 (2010) 377–486
© 2011 C. Castillo and B. D. Davison
DOI: 10.1561/1500000021

Adversarial Web Search

Carlos Castillo[1] and Brian D. Davison[2]

[1] *Yahoo! Research, Diagonal 177, 8th Floor, Barcelona 08018, Catalunya-Spain, chato@yahoo-inc.com*
[2] *Lehigh University, 19 Memorial Drive West, Bethlehem, PA 18015, USA, davison@cse.lehigh.edu*

Abstract

Web search engines have become indispensable tools for finding content. As the popularity of the Web has increased, the efforts to exploit the Web for commercial, social, or political advantage have grown, making it harder for search engines to discriminate between truthful signals of content quality and deceptive attempts to game search engines' rankings. This problem is further complicated by the open nature of the Web, which allows anyone to write and publish anything, and by the fact that search engines must analyze ever-growing numbers of Web pages. Moreover, increasing expectations of users, who over time rely on Web search for information needs related to more aspects of their lives, further deepen the need for search engines to develop effective counter-measures against deception.

In this monograph, we consider the effects of the adversarial relationship between search systems and those who wish to manipulate them, a field known as "Adversarial Information Retrieval". We show that search engine spammers create false content and misleading links to lure unsuspecting visitors to pages filled with advertisements or malware. We also examine work over the past decade or so that aims to

discover such spamming activities to get spam pages removed or their effect on the quality of the results reduced.

Research in Adversarial Information Retrieval has been evolving over time, and currently continues both in traditional areas (e.g., link spam) and newer areas, such as click fraud and spam in social media, demonstrating that this conflict is far from over.

1

Introduction

Information Retrieval (IR) is a branch of computer science that deals with tasks such as gathering, indexing, filtering, retrieving, and ranking content from a large collection of information-bearing items. It is a field of study that is over 40 years old, and started with the goal of helping users locate information items in carefully curated collections, such as the ones available in libraries. In the mid-1990s, the emergence of the World Wide Web created new research opportunities and challenges for information retrieval. The Web as a whole is larger, less coherent, more distributed and more rapidly changing than the previous document collections in which IR methods were developed [9].

From the perspective of an information retrieval system such as a search engine, the Web is a mixture of two types of content: the "closed Web" and the "open Web" [37]. The closed Web comprises a small number of reputable, high-quality, carefully maintained collections which a search engine can fully trust. The "open Web", on the other hand, includes the vast majority of Web pages, and in which document quality cannot be taken for granted. The openness of the Web has been the key to its rapid growth and success, but the same openness is the most challenging aspect when designing effective Web-scale information retrieval systems.

Adversarial Information Retrieval addresses the same tasks as Information Retrieval: gathering, indexing, filtering, retrieving, and ranking information, with the difference that it performs these tasks in collections wherein a subset has been manipulated maliciously [73]. On the Web, the predominant form of such manipulation is "search engine spamming" (also known as *spamdexing* or *Web spam*). Search engine spamming is the malicious attempt to influence the outcome of ranking algorithms, usually aimed at getting an undeservedly high ranking for one or more Web pages [92].

Among the specific topics related to Adversarial Information Retrieval on the Web, we find the following. First, there are several forms of general Web spam including link spam, content spam, cloaking, etc. Second, there are specialized forms of Web spam for particular subsets of the Web, including for instance blog spam (*splogs*), opinion spam, comment spam, referrer spam, etc. Third, there are ways in which a content publisher may attempt to deceive a Web advertiser or advertiser broker/intermediary, including search spam and click spam. Fourth, there are other areas in which the interests of the designers of different Web systems collide, such as in the reverse engineering of ranking methods, the design of content filters for ads or for Web pages, or the development of undetectable automatic crawlers, to name a few.

1.1 Search Engine Spam

The Adversarial IR topic that has received the most attention has been search engine spam, described by Fetterly et al. as "Web pages that hold no actual informational value, but are created to lure Web searchers to sites that they would otherwise not visit" [74].

Search engines have become indispensable tools for most users [17]. Web spammers try to deceive search engines into showing a lower-quality result with a high ranking. They exploit, and as a result, weaken, the trust relationship between users and search engines [92], and may damage the search engines' reputation. They also make the search engine incur extra costs when dealing with documents that have little or no relevance for its users; these include network costs for downloading them, disk costs for storing them, and processing costs for indexing them. Thus, the costs of Web spam are felt both by end-users and those providing a service to them.

Ntoulas et al. [182] measured Web spam across top-level domains (TLDs) by randomly sampling pages from each TLD in a large-scale Web search engine, and then labeling those pages manually. In their samples, 70% of the pages in the `.biz` domain, 35% of the pages in `.us` and 20% of the pages in `.com` were spam. These are uniform random samples, while the top results in search engines are much more likely to be spam as they are the first target of spammers. In a separate study, Eiron et al. [69] ranked 100 million pages using PageRank and found that 11 out of the top 20 achieved such high ranking through link manipulation.

Ignoring Web spam is not an option for search engines. According to Henzinger et al. [98], "Spamming has become so prevalent that every commercial search engine has had to take measures to identify and remove spam. Without such measures, the quality of the rankings suffers severely." In other words, on the "open Web", a naïve application of ranking methods is no longer an option.

1.2 Activists, Marketers, Optimizers, and Spammers

The existence of Web spam pages can be seen as a natural consequence of the dominant role of search engines as mediators in information seeking processes [85]. User studies show that search engine users only scan and click the top few results for any given search [87], which means that Web page exposure and visitor traffic are directly correlated with search engine placement. Those who seek visibility need to have pages in the top positions in search engine results pages, and thus have an incentive to try to distort the ranking method.

There are many reasons for seeking visibility on the Web. Some people (activists) spam search engines to further a political message or to help a non-profit achieve its end. This is the case of most link bombs (perhaps better known as *Google bombs*) that spam a particular term or phrase to link it to a particular Web page. A memorable example of this manipulation is the one that affected the query "miserable failure", which during the 2004 presidential election, returned the home page of

George W. Bush as the first result in several Web search engines. This was the result of a coordinated effort by bloggers and Web page authors around the world. We discuss link bombing further in Section 4.

Most search engine spam, however, is created for financial gain. There is a strong economic incentive to find ways to drive traffic to Web sites, as more traffic often translates to more revenue [231]. Singhal [212] estimated the amount of money that typical spammers expected to receive in 2005: a few US dollars per sale for affiliate programs on Amazon or E-Bay, around 6 USD per sale of Viagra, and around 20–40 USD per new member of pornographic sites. Given the small per-sale commissions and the low response rates, a spammer needs to collect millions of page views to remain profitable. Further, some spam pages exist to promote or even install malware [68, 192, 193].

The incentive to drive traffic to Web sites, both for legitimate and illegitimate purposes, has created a whole industry around search engines. The objective of *Search Engine Marketing* (SEM) is to assist marketers in making their Web content visible to users via a search engine.[1] SEM activities are divided by the two principal kinds of information displayed on a search results page: the editorial content and the advertising (or "sponsored search").

Advertising on search engines today is also a ranking process, involving bidding for keywords to match to user queries, the design of the ads themselves, and the design of the "landing pages" to which users are taken after clicking on the ads. An advertiser's goal in sponsored search is to attract more *paid* traffic that "converts" (i.e., buys a product or service, or performs some other action desired by the advertiser), within a given advertising budget.

Sponsored search efforts are fairly self-regulated. First, marketers have to pay the search engine for each click on the ads. Second, the marketer does not simply want to attract traffic to his Web site, but to attract traffic that leads to conversions. Thus, it is in his best interest to bid for keywords that represent the actual contents of his Web site.

[1] Some practitioners define SEM more narrowly, focusing on the sponsored search side, but from a business perspective, all of these efforts fall under marketing.

Also ad market designers are careful to design systems that provide incentives for advertisers to bid truthfully.

The objective of *Search Engine Optimization* (SEO), on the other hand, is to make the pages of a certain Web site rank higher in the editorial side of search engines, in order to attract more *unpaid* or *organic* traffic to the target Web site.

The efforts of a search engine optimizer, in contrast, are not self-regulating, and in some cases can significantly disrupt search engines, if counter-measures are not taken. For this reason, search engines threaten SEOs that have become spammers with penalties, which may include the demotion or removal from the index of pages that use deceptive practices. The penalties that search engines apply are well known by the SEO community. Boundaries are, of course, fuzzy, as all search engines seem to allow some degree of search engine optimization.

Moran and Hunt [169] advise Web site owners on how to tell search engine spammers from SEOs. A search engine spammer tends to (i) offer a guarantee of top rankings, which no reputable firm can do as there are many variables outside their control; (ii) propose minimal changes to the pages, which indicate that they are likely to create a *link farm* (described in Section 4.3) instead of actually modifying the way the content is presented to users and search engines; and (iii) suggest to use server-level *cloaking* (described in Section 3.5) or other modifications whose typical purpose is to spam.

1.3 The Battleground for Search Engine Rankings

In general, search engine results are ranked using a combination of two factors: the *relevance* of the pages to the query, and the *authoritativeness* of the pages themselves, irrespective of the query. These two aspects are sometimes named respectively *dynamic ranking* and *static ranking*, and both have been the subject of extensive studies from the IR community (and discussed in IR textbooks [13, 58, 154]).

Some search engine spammers may be assumed to be knowledgeable about Web information retrieval methods used for ranking pages. Nevertheless, when spammers try to manipulate the rankings of a search engine, they do not know the details about the ranking methods used by the search engine; for instance they do not know which are the specific features used for computing the ranking. Under those conditions, their best strategy is simply to try to game *any* signal believed to be used for ranking.

In the early days of the Web, search engine spammers manipulated mainly the contents and URLs of the pages, automatically generating millions of pages, including incorporating repetitions or variants of certain keywords in which the spammer was interested. Next, as search engines began to use link-based signals [33, 34, 122, 183], spammers started to create pages interlinked deceptively to generate misleading link-based ranking signals.

As the search engines adapted to the presence of Web spam by using more sophisticated methods, including the usage of machine-learning-based ranking for Web pages [201], more elements of the pages were taken into consideration which pushed spammers to become more sophisticated. Next, the possibility of adding comments to forums and the existence of other world-writable pages such as *wikis* presented new opportunities for spammers as they allowed the insertion of arbitrary links into legitimate pages.

Recently search engines have devised other ways of exploiting the "wisdom of crowds", e.g., through usage data to rank pages, but search engine spammers can also pose as members of the crowds and disrupt rankings as long as they are not detected. Web spam has been evolving over the years, and will continue to evolve to reflect changes in ranking methods used by popular services.

Thus, there are a variety of useful signals for ranking and each of them represents an opportunity for spammers, and in Sections 3–7 we will highlight how spammers have taken advantage of these opportunities to manipulate valuable ranking signals and what work has been done to detect such manipulation.

1.4 Previous Surveys and Taxonomies

In 2001, Perkins [189] published one of the earliest taxonomies of Web spam. This taxonomy included content spam, link spam, and cloaking. It also suggested a test for telling spam from non-spam: Spam is "any attempt to deceive a search engine's relevancy algorithm", non-spam is "anything that would still be done if search engines did not exist, or anything that a search engine has given written permission to do."

In 2005, Gyöngyi and Garcia-Molina [93] proposed a different taxonomy. This taxonomy stressed the difference between boosting techniques and hiding techniques. Boosting techniques are directly aimed at promoting a page or a set of pages by manipulating their contents or links. Hiding techniques, instead, are used by spammers to "cover their tracks", thus preventing the discovery of their boosting techniques.

In 2007, a brief overview of Adversarial IR by Fetterly [73] appeared in *ACM Computing Reviews*. It included a general description of the field, and references to key articles, data sources, and books related to the subject. In the same year Heymann et al. [101] published a survey focused on social media sites, stating that in the case of social media sites, a preventive approach was possible, in addition to detection- and demotion-based approaches. Prevention is possible because in social media sites there is more control over what users can do; for example, CAPTCHAs can be incorporated to prevent automated actions, the rate at which users post content can be limited, and disruptive users can be detected and banned.

Additionally, several Ph.D. and M.Sc. theses have included elements related to Web spam. A partial list of them includes theses in the areas of link spam [95, 149, 160, 208], splogs and spam in blogs [124, 166], content spam [180], Web spam systems in general [45, 232, 236, 251], and search engine optimization [123].

We have left out the closely related subject of e-mail spam. While some methods overlap, particularly in the case of content-based Web-spam detection (which we discuss in Section 3.6), there are substantial differences between the two areas. For a survey on e-mail spam, see, e.g., Cormack [55].

1.5 This Survey

In this survey we have tried to be relatively inclusive; this is reflected in citations to about 250 publications, which we consider large for a survey on a young sub-field of study. We also intended to appeal to a wide audience including developers and practitioners. For this reason, we have chosen to present general descriptions of Web spam techniques and counter-measures, and to be selective with the details.

The rest of this monograph is organized as follows:

> **Section 2** describes general systems for detecting search engine spam, including the choice of a machine learning method, the feature design, the creation of a training set, and evaluation methodologies.
> **Section 3** describes content-based spam techniques and how to detect them, as well as malicious mirroring, which is a form of plagiarism for spam purposes.
> **Section 4** describes link-based spam techniques and how to detect them, and covers topics such as link alliances and nepotistic linking.
> **Section 5** describes methods for propagating trust and distrust on the Web, which can be used for demoting spam pages.
> **Section 6** describes click fraud and other ways of distorting Web usage data, including Web search logs; it also deals with the subject of using search logs as part of Web spam detection systems.
> **Section 7** describes ways of spamming social media sites and user-generated content in general.

Finally, the discussion in **Section 8** includes future research directions and links to research resources.

2
Overview of Search Engine Spam Detection

Adversarial Web IR problems can be attacked from many different perspectives, including Information Retrieval, Machine Learning, and Game Theory. Machine learning methods have been shown to be effective for many document classification tasks and Web spam is not an exception. In this section we briefly outline how an automatic Web spam classifier is usually built; for surveys of approaches for text classification in general and Web page classification in particular, see Sebastiani [207] and Qi and Davison [195], respectively.

We discuss first how to create a training corpus in Section 2.1, then how to represent documents through features in Section 2.2. Next, we discuss the choice of a learning mechanism in Section 2.3 and the evaluation of a system in Section 2.4.

2.1 Editorial Assessment of Spam

Current Web spam classification systems used by search engines require some degree of supervision, given that spam techniques may vary extensively. Moreover, the difference between spam and non-spam pages can be the result of very small changes.

The design of a sound labeling procedure for the training instances includes the development of clear guidelines for the editors. This first necessitates the operationalization of a definition of spam. For example, given that the Web is comprised of pages, perhaps the pages containing inappropriate material should be marked; or perhaps it is the pages that benefit from the inappropriate material (e.g., the page that is the target of an inappropriate link) that should be marked; or perhaps it is the material itself (link, content, redirection, etc.) that is marked, and not the pages containing or benefiting at all. Moreover, there is a decision as to the granularity (domain, host, page, or page element) at which the label should be applied. For instance, tagging at domain level versus tagging at site or page level may influence the evaluation of the effectiveness of spam detection methods, particularly if some hosts of a domain are in the training set and some hosts in the testing set [217].

In their help pages, search engines have various definitions of what constitutes Web spam, emphasizing different aspects. These public guidelines show only a moderate level of overlap among different search engines, and they also tend to be terse. The guidelines that are actually used by the editors that work for search engines are not known in detail.

One approach to designing guidelines for the editorial assessment of spam (used in the creation of a few public datasets[1]) is to enumerate different spamming characteristics that pages may have, and describe them through examples.[2] This means framing the task as "finding pages with spamming aspects", and trying to the best possible extent to decouple the problem of finding spam from the problem of assessing the quality of Web pages.

It is important that the editors realize that opposite of spam is not high-quality; in theory spam and quality are independent axes and the opposite of spam is simply "non-spam".[3] In practice spam and quality tend to be correlated, but apart from the expected low-quality + spam and high-quality + non-spam examples, there is also

[1] http://barcelona.research.yahoo.net/webspam/datasets.

[2] See http://barcelona.research.yahoo.net/webspam/datasets/uk2007/guidelines/ for one such set of guidelines and examples.

[3] Also referred in the e-mail spam literature as *ham*.

low-quality + non-spam and high-quality + spam. For instance, in 2006 Google temporarily removed[4] both `bmw.de` and `ricoh.de` from its index after detecting that these sites, which hold high-quality and legitimate content, used deceptive JavaScript-based redirects.

2.1.1 Subjectivity in Assessment

Labeling Web spam is a task involving a great deal of subjectivity. There are cases in which spam is obvious to a human, and cases where spam is hard to see. There are many borderline cases, including pages that seem to provide utility for users by themselves, but also use reciprocal linking to distantly related sources that are suspicious of being spam. There are also cases of unsophisticated spam, such as intentional reciprocal linking among family or friends (and general link exchanges) [62].

In practice the degree of agreement in spam assessment tasks has been reported as either poor ($\kappa = 0.45$ in [21]) or moderate ($\kappa = 0.56$ in [43]). To alleviate this problem, the task must be specified very carefully to the assessors, making sure that they understand the definition correctly. An alternative to compensate for the low agreement is to collect many pairs of judgments so that we are likely to find more pairs in which the editors agree (in which case we would perhaps throw away the rest).

Some have proposed that more assessments can be collected by using a form of a two-player game [84] similar to the ESP game for labeling images [227]. In any case, given a small budget of assessments, *active learning* can be used. In active learning, a classifier or a set of classifiers is available at the time the assessments are collected. The classifiers run over the whole collection finding items that they cannot classify with high confidence, or in which they disagree on the predicted label. Those examples are the ones presented to the editors. This reduces the time required of editors, and it has been studied in the context of splogs by Katayama et al. [121].

[4] http://www.mattcutts.com/blog/ramping-up-on-international-webspam/.

2.1.2 Hybrid Sites with Both Spam and Non-spam Pages

If the classification is done at the host level, it is important to consider that some hosts may supply a mixture of non-spam and spam contents. Some hosts include publicly writable pages (see Section 7.3) such as *Wikis* or pages containing comment forms that can be abused by spammers to insert links to the sites they want to promote.

Other sites are simply compromised by spammers. Data from the closely related area of *phishing* (a widespread type of e-mail fraud) indicates that the landing pages of the “phishing” messages are hosted in compromised Web sites in 76% of the cases and on free hosting sites on 14% of the cases [168]. The same reference indicates that Web sites that are vulnerable to compromise can be easily located through simple Web searches, e.g., by searching for names of popular scripts known to be vulnerable; so easy to locate indeed that 19% of the compromised phishing sites are re-compromised in the next six months.

This means that in general it is not safe to assume that hosts are either entirely spam or entirely non-spam, and that Web spam classification should occur at a finer granularity than that of entire hosts.

2.2 Feature Extraction

To be effective, an automatic Web spam classifier needs a rich document representation that takes many aspects of a page into account before taking a decision. This representation is obtained by collecting typically hundreds of features for each page. These features, while typically specific to a page, can be divided by their sources.

There are basically three points in time at which features can be computed: while pages are being crawled, while pages are being indexed, and while pages are being ranked in response to a user’s query. Next we provide an overview of them, and defer the details to the sections dealing with specific types of spam.

2.2.1 Index-Time Features

Out of the three sources of features, the most important role is played by *index-time features*, which include any features that can be calculated at index-generation time (that is, without knowledge of a particular query).

Content-based features are those features that are simply a function of the content of the Web object (e.g., the text on the page).

Content-based spam detection methods are discussed in detail in Section 3, particularly in Section 3.3. One of the most comprehensive studies of content-based features is due to Ntoulas et al. [182]. Among their findings, they note that many spam pages: (i) have an abnormally high number of words in the title, (ii) are either longer or shorter than non-spam pages, (iii) use words that are longer than average, (iv) contain less HTML markup and more text, and, (v) have more redundant content as measured by applying a text compression algorithm and observing the compression ratio. Spam pages also tend to contain words that appear in popular queries, among other features that can be exploited by a classifier.

Link-based features are those features that reflect the existence of hyperlinks between Web pages. These may be locally calculated values (e.g., number of outgoing links) or a global value such as the importance of the page compared to all pages in the graph.

Link-based spam detection methods are discussed in detail in Sections 4 and 5. Link-based metrics can be used as features for the pages and hosts being analyzed. Section 4.4.2 describes features that can be extracted to detect anomalous linking patterns, such as degree correlations, neighborhood sizes, etc. Gyöngyi et al.'s TrustRank [94] is an example of a global value which has been used as a feature for spam classification (described in Section 5.3).

Links can also be incorporated directly during the learning process (instead of using them simply to compute features), as explained in Sections 2.3.2 and 2.3.3.

Usage-based features are characteristics extracted from records of human interactions with the pages or sites, and include measures such as the number of times a particular site was visited.

Usage-based spam detection methods are discussed in detail in Section 6. Usage data can be used to detect spam; for instance, *browsing* *trails* can be used to find sites in which users spend too little time, or to which users never arrive by following links. These trails can be collected, e.g., by a toolbar or add-on supplied by the search engine that, with explicit permission from the user, submits anonymous information about the user's activities. This is discussed in Section 6.3.1.

Query sessions collected in search engine logs can be used to identify popular queries (which can help determine if a page is made almost exclusively of popular query terms), or to identify pages that attract visits from too many unrelated queries. This is discussed in Section 6.3.2

Temporal features are features that incorporate time or change over time, such as the number of incoming links a page has acquired this year.

A recent development has been the consideration of how the Web changes over time and the effect on Web spam. Chung et al. [53] report on how link farms evolve over time, finding that large link farms were created quickly, and that they did not grow. Others [60, 71] asked the question of whether historical information could be of value in spam detection. Dai et al. [60] obtained archival copies of Web pages from the Internet Archive, and based on features derived from how the pages changed in the past, obtained a substantial improvement in classifier accuracy.

2.2.2 Crawl-Time Features

If a Web crawler can discard pages that are almost certainly spam, a search engine can avoid the costs of storing and indexing them, providing substantial savings. This can be achieved by avoiding crawler traps, by prioritizing high-quality sites, or by running an automatic spam classifier at the crawler.

Heydon and Najork [100], when documenting their Web crawling system Mercator, describe the presence of *crawler traps* as early as 1999. Crawler traps are programs on Web sites that automatically generate an infinite Web of documents, and many sites that contain an abnormally high number of documents are indeed cases of crawler traps. One option is to establish a maximum number of pages to download per host; Lee et al. [140] propose to allocate these maxima proportionally to the number of in-links from different domains received by each host in the crawler's queue.

URLs sometimes provide enough information to guide the crawling process. Bar-Yossef et al. [16] present a classification scheme to avoid downloading near-duplicates by identifying different URLs leading to the same text. Ma et al. [152] propose a spam-specific classifier that uses URL features, in particular information from the hostname portion (e.g., IP address or geolocation).

Crawlers can also exploit features gathered from the HTTP response to the HTTP request used to fetch a page. Webb et al. [234] observe that when comparing spammers with non-spammers in a large corpora from the Web, the distribution of Web sites into IP addresses is much more skewed for spammers than for non-spammers. This means that spammers are often concentrated into a few physical hosts. If spammers also use the same software in all their hosts, this can be used by search engines to increase the effectiveness of features obtained from the HTTP response headers while crawling. For instance, these headers may provide information about the specific server version and module versions being used, thus helping identify a group of servers belonging to a single entity.

In general, a scheduling policy for crawling that emphasizes page quality can help stop spam at crawling time; a survey of such policies appears in [40].

2.2.3 Rank-Time Features

Many pages optimized for search engines actually succeed in fooling search engine ranking methods. The amount of spam pages is massive, and for queries that have moderate (neither very high nor very low) frequencies, it may be the case that there is a spam page that is heavily optimized for that query. For instance, a page containing all the query terms in the title and the URL will have a big boost in ranking, even if textual similarity is just one of many factors of the search engine ranking.

Svore et al. [217] suggest to use query-dependent features that are computed after a (preliminary) set of result pages is selected and ranked according to some ranking algorithm. These rank-time features can be used to build a last "line of defense" against spam, and may include for instance the number of query terms in the title and other sections of a document, counts of different occurrences of the query terms across documents, and word n-gram overlaps between the query terms and the document. The purpose of these features is to find anomalies and remove or demote results that were ranked high by the search engine but whose statistical properties deviate significantly from the other search engine results of the same query.

2.3 Learning Schemes

In this section we present the types of learning schemes used to train Web spam detection systems. First, *local* learning methods consider each node as a separate entity, and independently infer its label (spam or non-spam). Second, *neighborhood-based* learning methods introduce additional features for each node, based on the nodes they are linked to; the process still infers the label of each node independently. Third, *graph-wide* learning methods compute simultaneously the labels for all of the nodes in a graph, using the links between nodes as dependencies that have to be taken into account during the learning process.

2.3.1 Local Methods

Link-based features and content-based features can be used together to classify each page in isolation based on all the different signals available. Indeed, this was the approach taken in one of the earliest works in this subject by Davison [62], in which multiple signals were incorporated into a single classifier (albeit to recognize spam links, rather than spam pages). The classifier used was a C4.5 [197] decision tree. Wang et al. [229] also describe a classifier based on Ranking Support Vector Machines (Ranking SVMs) [99] that ranks pages into three classes (good, normal, and spam) using features from the contents and links of the pages.

A common concern for these methods is how to perform *feature aggregation* in the cases in which features are extracted at a different granularity from the one at which the classification needs to be performed. For instance, one could need to classify entire sites based on features that include the sizes or number of links of individual pages. In this case, a possible approach is to compute multiple aggregates from the pages of a site, representing each host by features including, e.g., the average page size, the maximum page size, the average number of links, and the maximum number of links.

2.3.2 Neighborhood-Based Methods

Non-graphical features can be used in conjunction with the link structure of the Web to create graph-regularized classifiers that exploit "guilt by association"-like rules. Figure 2.1 illustrates that hosts connected by links tend to belong to the same class (either both are non-spam or both

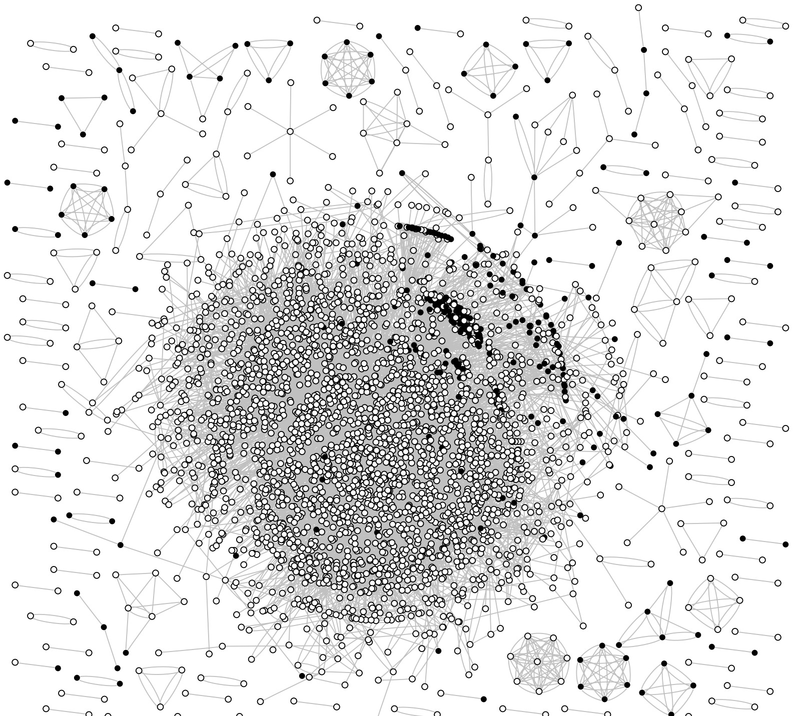

Fig. 2.1 Host graph from [44] including several thousand nodes from the .uk labeled by editors. Black nodes are spam sites and white nodes are non-spam sites.

are spam). This correlation can be exploited when learning to detect spam nodes.

One might simply look at the result of classifying neighboring nodes when deciding the class of the current node. Qi and Davison [194] demonstrate this method for topical classification of Web pages.

In stacked graphical learning, this approach can be applied more than once (allowing for propagation beyond immediate neighbors). A standard classifier is first used to obtain a base prediction for a node. Let k be the number of features used by this classifier. Next, an aggregate (such as the average) of the base predictions of the neighbors of a node is used to compute an extra feature. This extra feature is added to the features of the original node, and then the base classifier is trained again using $k + 1$ features. For the task of Web spam detection, the resulting classifier has been shown to be more accurate [44]. This method can be applied recursively for a few iterations, stopping when the accuracy in the test set no longer improves. Computationally, this is fast given that in practice the base classifier needs only to be invoked a few times, and the extra feature can be computed quickly.

2.3.3 Graph-Wide Methods

Another approach is to consider the graph structure directly when stating the objective function for the learning process. The methods that take into account this type of dependencies are known as graphical learning methods or more generally as collective inference methods.

At a high-level, these methods operate in a *transductive* setting, a learning paradigm in which all the test instances (all of the Web pages indexed by the search engine) are known at training time. This includes the pages for which we do not have human-provided labels. By knowing the testing set in advance, the algorithm can produce a classifier that generates "smooth" predictions, that is, predictions that are similar for neighboring nodes. As an example, in Section 4.6, we discuss Abernethy and Chapelle [2] and Abernethy et al.'s [4] study which demonstrates the potential for graph regularization in Web spam detection.

Finally, not only edges, but also nodes, can be given different weights. Most pages on the Web are seldom visited and never show up among top results in search engines, and there is a small fraction of pages that are very important for users. This motivates incorporating measures of page and link importance during the learning process. Zhou et al. [254] describe a method in which the PageRank score for the nodes in the graph is computed, and then PageRank scores are used to weight nodes and PageRank flows are used to weight edges. These weights can be used during the learning process; for instance classification errors on the top pages can be given more weight than classification errors on pages that are not so important.

2.4 Evaluation

Given a ground truth consisting of a set of labels for elements known to be spam or known to be non-spam, for evaluation purposes the set is divided into a *training set*, used to create the automatic classifier, and a *testing set*, used to evaluate it. In this section, we outline some of the commonly used methods for evaluating Web spam detection systems.

2.4.1 Evaluation of Spam Classification Methods

There are a number of methods for the evaluation of automatic classifiers, see, e.g., [235]. In the context of Web spam detection, given a classification method and a testing set, we can examine first its confusion matrix:

		Prediction	
		Non-spam	Spam
True Label	Non-spam	a	b
	Spam	c	d

where a represents the number of non-spam examples that were correctly classified, b represents the number of non-spam examples that were falsely classified as spam, c represents the spam examples that were falsely classified as non-spam, and d represents the number of spam examples that were correctly classified.

For evaluating a classification algorithm, two important metrics are the *true positive rate* (or recall) and the *false positive rate*. In a Web spam detection system, the true positive rate R is the amount of spam that is detected (and thus may be deleted or demoted). The false positive rate is the fraction of non-spam objects that are mistakenly considered to be spam by the automatic classifier. The true positive rate R is defined as $\frac{d}{c+d}$. The false positive rate is defined as: $\frac{b}{b+a}$.

The *F-measure* F (also called F_1 score) is a standard way of summarizing both aspects in a single number. The F-measure is defined as $F = 2\frac{PR}{P+R}$, where P is the precision $P = \frac{d}{b+d}$.

Most classification schemes generate a binary prediction (non-spam or spam) based on an estimation of the probability that an object is spam (a "spamicity" score) which is then thresholded to produce the final output. The drawback of the F-measure is that it requires a fixed choice of classification threshold, and the resulting performance can be quite sensitive to that choice.

As a result, it is better to ignore the choice of a threshold, and evaluate instead the ordering of the pages induced by the "spamicity" estimates of an algorithm. The *Area Under the ROC curve* (AUC) metric provides a natural measure of the accuracy of a predicted ranking, and requires only that the algorithm outputs an ordering of the test set. A good classification algorithm should give higher spam scores to spam pages than to non-spam pages; the threshold and the weight given to the spam score in the final ranking are left as choices for the search engine designer who uses a spam classifier.

2.4.2 Evaluation of Spam Demotion Methods

Some methods for fighting Web spam are not based on classification. Instead, they try to modify the way a certain authority estimation method is computed in order to produce a different ranking. For instance, they might alter the way PageRank counts different links to lessen the effect of link manipulation. In this case, typical evaluation compares the original ranking ordering with the modified one.

This type of evaluation is applied in several papers introducing spam-aware ranking methods (e.g., [20, 21, 94, 178, 242, 243]). The elements (pages or hosts) are divided into a set of b buckets. The elements in each bucket are assigned in descending rank order on the basis of the authority score given to each element by the ranking function.

The assignment is such that each bucket contains elements whose scores add up to $1/b$ of the score.

Thus, the last bucket contains the set of smallest valued elements. The next to last contains the next smallest scores, and so on. The first bucket contains the highest-scoring pages. If the scores are distributed according to a skewed distribution such as a power-law (which is the case for PageRank), then the first bucket contains very few elements compared to the last bucket.

At this point, one can then ask about the distribution of subsets of those pages (e.g., where spam pages are located). Proposed spam-aware ranking methods would then use buckets of the same sizes and again distribute pages according to their score. A successful method would be one that tends to push less-desirable pages toward the bottom of the ranking (to buckets with low score) and potentially desirable pages toward the top.

Since this approach implicitly considers the importance of a page (e.g., there is a high cost for permitting a spam page to rank highly), it is arguably an improvement over the simpler methods for spam classification that do not consider ranking positions. However, in most cases we do not know whether the ordering generated (that also demotes spam) is valuable as an estimate of authority for result ranking. Moreover, no single metric for demotion has gained enough acceptance in the research community to be used for comparison.

In summary, methods that are evaluated based on spam classification or spam demotion cannot be assumed to reduce directly the impact of spam on retrieval by users.

2.4.3 Evaluation in Retrieval Context

One of the principal reasons to detect search engine spam is to improve the results seen by a searcher. Thus, much of the research we describe is intended to ameliorate the effect of search engine spam. Jones et al. [117] refer to this as *nullification.* They also distinguish between detection and nullification; the effect of removing all spam pages might miss the spam links between "good" pages, or might punish sites that permit user submitted content (which we will discuss in Section 7).

Unfortunately, relatively few researchers have adopted this approach. Notable exceptions include Davison et al. [176, 177, 240, 241], Jones et al. [116], and very recent work by Cormack et al. [56]. This is likely the result of several factors. First, depending on the data set and queries chosen, there may be few instances of spam pages ranking highly and thus having an impact on quality metrics. Second, and more significantly, it requires more resources than spam detection: at minimum a full search engine, indexed data set, and queries, but typically also require (expensive) relevance judgments of the kind used in the long-running Text Retrieval Competitions (TREC) organized by the U.S. National Institute of Standards and Technology. Thus, most of the research that we will describe focus on spam detection.

2.5 Conclusions

This section described how to create a Web spam page classification system. We started by examining the problem of acquiring labeled data and the extraction of features for use in training a classifier. Typical classifiers were shown to use the content on a page, the links between pages, and even the content or classes of neighboring pages. Finally, in Section 2.4, we saw that most evaluation focused on the quality of the classifier, while some work also considered the effects on ranking. Most of these ideas will be present in the remaining sections, which go into additional detail on specific methods for dealing with search engine spam.

3

Dealing with Content Spam and Plagiarized Content

Early search engines such as *Webcrawler*, *Altavista*, and *Lycos* relied mainly on the content of pages and keywords in URLs to rank search results. Modern search engines incorporate many other factors but content-based methods continue to be an important part of the computation of relevance. However, page contents and URL keywords can be easily manipulated by spammers to improve their ranking. The methods that spammers use typically involve repetition or copying of words or passages of text that contain keywords that are queried frequently by search engine users.

This section is about content-based Web spam, and methods to detect Web spam that are content-based. We start with some background on features for static and dynamic ranking. Then we describe particular forms of content-based Web spam including malicious mirroring and cloaking. While not being the central topic of this monograph, in the last section we provide a couple of pointers about e-mail spam detection.

3.1 Background

Modern search engines use a variety of factors to compute the importance of a document for a query. Richardson et al. [201] indicate that these factors include similarity of the document to the query, ranking scores obtained from hyperlink analysis, page popularity obtained from query click-through logs, and other features about the page, host, or domain itself.

3.1.1 Static and Dynamic Ranking

Query-independent features are also referred to as *static ranking* features, and they can be pre-computed at indexing time, thus saving time when answering a user's query. Among these features are aspects of the pages such as total document length, frequency of the most frequent word, number of images embedded in the page, ratio of text to markup, and many others. From a search engine's perspective, the effectiveness of a large number of features can be tested using a feature selection method to filter out the irrelevant ones, so in practice the number of static features that are computed can be a few hundreds.

Most of the features for static ranking are under the control of the document author because they depend on aspects of the page itself. For search engines, this means that the specific list of features that have a high weight in the ranking function cannot be disclosed or Web sites will optimize their pages according to those features, rendering them ineffective for ranking.

Without knowing exactly why content is considered high quality, one way of leveraging the quality is, for instance, to copy well-written articles from other sources and modify slightly the copies to include spam terms or links. The purpose of such a copy is to create a spam page having static feature values resembling those of a high-quality page, with the hope of making it rank highly in search engine results or hoping that the high-quality text will make a spam link more believable.

Dynamic ranking features, on the other hand, are those that can only be calculated at query-time, and most obviously include estimates of the relevance of the document to the query but can also include user-specific features such as the searcher's country, query history, time-of-day, etc. Given our focus on content in this section, we next describe in more detail how the content can be modeled and how the relevance of a query to a document can be estimated.

Some researchers have performed controlled experiments to determine what factors (dynamic and static) are important to the ranking process used by search engines [27, 68, 216]. The methodology used to perform some of these experiments is public and can be reproduced by some spammers.

3.1.2 Document Models

Bag-of-words The dominant paradigm for determining the relevance of a document to a query is the vector-space model, described by Salton et al. [204]. The vector-space model is an instance of a "bag-of-words" model in which only the number of occurrences of words in a document is taken into account, but not the ordering in which they appear.

We sketch here a basic version of the vector-space model; for details see [13]. Both the query $\mathbf{q}$ and each document $\mathbf{d_i}$, $i = 1, 2, \ldots, N$ in a document collection D of size $|D| = N$ are represented as vectors in $\mathbb{R}^T$, where T is the total number of terms in the collection.

The value of the j-th coordinate of a document vector $\mathbf{d_i}$ indicates roughly the strength with which document $\mathbf{d_i}$ is associated to term j and how rare is term j in the collection. This strength is often computed as a product of a *term frequency* $\mathrm{tf_{i,j}}$ and an inverse document frequency $\mathrm{idf_j}$:

$$(\mathbf{d_i})_j \triangleq \mathrm{tf_{i,j}} \times \mathrm{idf_j}.$$

The *term frequency* gives more importance to words appearing multiple times in a document with respect to words appearing less often. The frequency of term j in document $\mathbf{d_i}$ can be defined as:

$$\mathrm{tf_{i,j}} \triangleq \frac{n_{i,j}}{\sum_{k=1}^{T} n_{i,k}},$$

where $n_{i,j}$ indicates the number of occurrences of term j in document $\mathbf{d_i}$.

The *inverse document frequency* gives more importance to rare words that do not appear in many documents as opposed to words like "the" which appear everywhere in an English language document collection. For a term j, its inverse document frequency can be defined as:

$$\mathrm{idf_j} \triangleq \log \frac{N}{|\{\mathbf{d_i} \in D : n_{i,j} > 0\}|},$$

that is, the logarithm of the reciprocal of the fraction of documents of the collection in which the term j appears.

Using this representation, the relevance of document $\mathbf{d_i}$ for a query $\mathbf{q}$ is defined as the cosine of the respective vectors. There are more elaborate choices for representing the documents and the query; a popular choice is Okapi BM25 [203] which follows basically the same principles but differs in the specific computation of the coordinate values.

From an adversarial perspective, the document $\mathbf{d_i}$ is under the control of its author, who has complete control over the number of occurrences $n_{i,j}$ of any term j in the document and thus has an opportunity to manipulate the ranking function as we show in Section 3.2.

n-Grams and term proximity In the context of information retrieval, a language model for a document collection is usually understood as its distribution in terms of words or sequences of words. A popular class of language models are *n-gram* models. A *word n-gram* is a sequence of n-words in order. When there is no possible confusion with *character n-grams*, which are sequences of n characters, word n-grams are simply called *n-grams*. n-Grams are useful from an information retrieval perspective as they preserve the ordering of words in a document, as opposed to "bag-of-words" models, while keeping the computational requirements low due to the use of a fixed length for the sequences.

Rasolofo and Savoy [199] introduce the idea of using *term proximity* during the ranking process. Documents in which query terms appear close to each other are given a higher ranking than documents in which the query terms are spread across different passages of the document. This method can be implemented efficiently in practice. The technique is improved and evaluated experimentally by Büttcher et al. [39].

Again, from an adversarial perspective, the frequencies of n-grams and the term proximities can be manipulated by the document author, thus opening the possibility of gaming the information retrieval system.

At the same time, a richer document representation yields more ways of detecting spam pages.

3.2 Types of Content Spamming

Gyöngyi and Garcia-Molina [93] introduce a comprehensive taxonomy of content spam. Content spam can be created, according to their nomenclature, by:

- *repetition* of terms to boost their TF values in TF.IDF weighting;
- *dumping* unrelated terms or phrases into the page to make the page at least partially "relevant" for multiple topics;
- *weaving* spam phrases into non-spam content copied from other sources; or
- *stitching* together non-spam content to create new artificial content that might be attractive for search engines.

A different dimension of content spam classification is the location of the spammed content. If the spam content is on the page itself, it can be either body spam, title spam, or meta-tags spam depending on the part of the HTML document where the spam is located. The spam content can also be located outside the page, for instance in the URL, by creating long URLs or host names with many terms, or in the anchor text of a link farm created to boost the popularity of the target page. All of these may be sources of signals used by a search engine for query relevance assessment.

3.3 Content Spam Detection Methods

3.3.1 Document Classification Methods

Ntoulas et al. [182] describe several content-based features, some of which were already mentioned in Section 2.2.1.

Besides differences, e.g., in length, number of words in the title, and other characteristics, they found that spam pages have an abnormal language model, including the fact that they contain more *popular* terms than non-spam pages. This is exploited, e.g., by Chellapilla and

Chickering [50] to detect cloaking pages. Ntoulas *et al.* also built a 3-gram language model for the whole collection and compared it with the subset of spam and non-spam documents. They found that spam pages have abnormally low or abnormally high likelihood given the collection, basically, because their distribution of n-grams is often substantially different from the background distribution. The likelihood of a document in this setting is the probability of generating that document by a process of independently drawing n-grams using the language model of the whole collection, until reaching the document length.

A richer representation of the textual content of the documents can be used to improve the accuracy in this classification task. For instance, Piskorski et al. [190] experimented with annotating the documents with part-of-speech (POS) tags that indicate the morphological class of each word (e.g., adjective, noun, verb, etc.). This leads to features such as POS n-grams that can indicate, for instance, that a sequence such as $\langle noun, verb, noun\rangle$ is more likely than $\langle verb, verb, verb\rangle$ in non-spam pages written in English.

Textual features can also include term distance features, as proposed by Attenberg and Suel [10]. The proposed method computes the frequency of pairs of words at a certain distance (lying in a particular distance bucket); and use this as a feature for classifying documents as spam or non-spam.

Rather than a static content analysis, Zhou et al. [253] propose to calculate the maximum query-specific score that a page with n keywords and l occurrences can achieve, and pages with scores close to that maximum are considered more likely to be exploiting term spam.

Content-based features obtained at the page level can be aggregated to obtain features for classifying at the level of entire hosts. Fetterly et al. [74] report three such useful aggregates. The first two are the variance of word counts of all pages served by a single host and the distribution of the sizes of clusters of near-duplicate documents. Both help detect the use of templates in the spam page generation process. The third is the average change in page content from week to week, which would find sites with content that changes almost completely each week (a signal of spamming activity).

A different approach is to use a generative model for documents, such as Latent Dirichlet Allocation (LDA), a paradigm introduced by Blei et al. [30]. In LDA, when writing a document given a language model, the author first picks a topic according to a distribution over topics, and then picks a word according to a topic-dependent distribution over words. Bíró et al. [29] use a *multi-corpus* LDA to find, roughly speaking, whether a document is more likely to have been generated from a non-spam model or from a spam model; both models are inferred by training on a set of labeled examples. More recently, Bíró et al. [28] improved on that performance by 3–8% by using a linked LDA model in which topics are propagated along links.

3.3.2 Classifying Pairs of Documents

A number of researchers have focused on the task of detecting nepotistic links in a document collection using some or all of the content of the source and target pages [20, 62, 159, 196]. The assumption in most of these approaches is that in a non-spam link, the content of the source document and the target document should be similar.

Benczúr et al. [20] and Martinez-Romo and Araujo [159] measure the Kullback–Liebler divergence of the unigram language model of both documents and consider a link as nepotistic if it exhibits a very high divergence. Benczúr *et al.* noticed that comparing all pairs of documents connected by a link may be computationally prohibitive, so they suggested comparing only the anchor text (or a few words around it) in the source document with the target document. Martinez-Romo and Araujo additionally explored other subsets of content, including internal links (pointing to pages on the same host) versus external links (pointing to pages on a different host), URL terms, surrounding anchor text, titles, etc.

In the context of comments in a blog, Mishne et al. [167] describe how to detect spam comments in blogs by analyzing the disagreement between the language model of a comment and the language model of the blog posting to which the comment is directed. This is described in more detail in Section 7.3.2.

3.3.3 Detection Using Coding-Style Similarity

A weakness of Web spam that can be exploited by Web spam detection systems is that most of it is automatically generated. Urvoy et al. [225, 224] preprocess Web pages by considering the page as an XHTML document and removing all element names, attribute names and values, and all printable character data. For instance, a Web page such as:

```
<p class="myclass">This is an <b>example</b></p>
```

is transformed into:

```
< =""><></></>
```

Next, similarity between the *coding style* of two Web pages can be computed directly by means of character n-gram comparison, or indirectly by using a faster technique such as hash sketches [36] (described below in Section 3.4). If two pages are similar in this regard, they are likely to have been generated using (variants of) the same template, which can be used as a proxy for authorship. This allows propagation of information about known spammers to other pages sharing the same coding style.

3.4 Malicious Mirroring and Near-Duplicates

Fetterly et al. [75] observed a large number of pages containing text that was automatically generated by the "stitching" of random "phrases" copied from other pages. In many cases these were not even phrases in the linguistic sense, but instead were just word n-grams. To be able to find phrase duplication on the Web, they resorted to an algorithmic technique based on sampling that can be used for computing a fast estimation of the size of the intersection of two sets [36].

The phrase-level duplicate detection works as follows. First, each sequence of n-words in a document d (such sequences are referred to as "*shingles*" [36]) is hashed using a fixed hash function f. This gives $n_d - n + 1$ hashes per document $f_1(d), f_2(d), \ldots, f_{n_d-n+1}(d)$ where n_d is the total number of words in the document. Next, a set of m different hash functions $h_1, h_2, \ldots, h_m$ is applied in turn to each of the hashes; we retain the minimum value obtained for each of them, which is called a fingerprint, *hash sketch* or simply *sketch*. Then, a document is represented by a set of sketches: $s_1, s_2, \ldots, s_m$ where

$$s_i(d) \triangleq \min_{j=1,2,\ldots,n_d-n+1} h_i(f_j(d)).$$

The sketches are useful because they can be used to quickly compute the similarity between two documents. More specifically, let $J(A,B)$ denote the *Jaccard coefficient* between two sets A and B, defined as $J(A,B) \triangleq |A \cap B|/|A \cup B|$. Given two documents u and v having sketches of sizes n_u and n_v respectively, we have that [36]:

$$J(\{f_i(u)\}_{i=1}^{n_u}, \{f_i(v)\}_{i=1}^{n_v}) \approx J(\{h_i(u)\}_{i=1}^{m}, \{h_i(v)\}_{i=1}^{m}),$$

where the right-hand side is much faster to compute given that $m \ll n_u, n_v$. In practice, Fetterly et al. [75] use $m = 84$ and compute on sequences of $n = 5$ words. They found that even after discarding duplicates and near-duplicates, about a third of the pages on the Web have more phrases in common with other pages than phrases that are unique to that page. Also, they found that pages with an abnormally high fraction of phrases shared in common with other documents in the collection are more likely to be spam than to be non-spam.

Wu and Davison [240] also considered the issue of near-duplicates, but focus on duplicate "complete links" in which both anchor text and target URL were copied. When a sufficient number of complete links are found to have been copied, the weights of those links are reduced accordingly, thus ameliorating much of the effect of link farms and replicated pages.

3.5 Cloaking and Redirection

Cloaking is a technique by which a Web server provides to the crawler of a search engine a page that is different from the one shown to regular users. It can be used legitimately to provide a better-suited page for the index of a search engine, for instance by providing content without ads, navigational aids, and other user interface elements. It can also be exploited to show users content that is unrelated to the content indexed by the engine. While they use different mechanisms, redirection and visual cloaking have a similar effect — the content that a user sees is different from that seen and indexed by the search engine.

3.5.1 Semantic Cloaking

Cloaking can be used as a spamming technique to deceive the search engine, when the page sent to the search engine is semantically different from the page shown to users. Thus, malicious cloaking is sometimes referred to as *semantic cloaking* [238, 239].

Detecting if a page is using cloaking is not easy. It may require the search engine to pose as a regular user (e.g., by changing the `user-agent` header sent to the Web server), which is against broadly accepted rules of behavior for Web crawlers [130]. Moreover, search engine optimization folklore suggests that spammers keep and exchange lists of IP addresses associated with search engine crawlers. Search engines must vary the IP addresses they use when testing semantic cloaking using a crawler.

It is also not enough to compare two copies (the regular crawler and the browser's-perspective crawler) as there are many dynamic pages that can yield false positives. A possible method for detecting crawling is the following, described by Wu and Davison [239]. First, download two copies from each page, one from a browser's perspective and one from a crawler's perspective. If the two copies are identical, there is no cloaking. If they are sufficiently different, download one more copy from each perspective to verify that the difference is not due to normal changes to the page. If the browser-perspective and crawler-perspective copies are equal in the same perspective but different across different perspectives, flag the page as cloaking.

Once a set of pages employing cloaking has been identified, an automatic classifier can be built to identify which of them are cases of semantic cloaking. Such a classifier is described in [239] and relevant features include whether the crawler-perspective has more meta-tags, words or links than the browser-perspective. Recently, Lin [146] proposes methods that leverage HTML tag multisets, particularly for dynamic pages, as the tag structure for a Web page is likely to persist over time even if page contents change.

The above method can be refined by adding intermediate steps to avoid false positives, as suggested by Chellapilla and Chickering [50], and by using external knowledge about the Web. This may include characteristics that indicate semantic cloaking, such as the presence of popular or highly monetizable queries.

In practice, for a search engine using any of these methods, generating multiple requests for every page in a crawl is simply infeasible. In such cases, either a sample of suspicious pages, or multiple copies of the same file obtained in different visits to the same page for refreshing the search engine's copies, can be used.

A completely different method for obtaining the browser-perspective is suggested by Najork [173], by using a fingerprint obtained by a toolbar installed in some user's browsers, which is transmitted to the search engine and compared with a similar fingerprint obtained from the crawlers' perspective.

3.5.2 The 302 Attack

A particular type of cloaking involving HTTP headers was known to the SEO community as the "302 attack" [206] referring to the HTTP code for a "temporary redirection".

This attack works as follows. A spammer creates a page U, and submits it to a search engine to be crawled (or includes a link to it in one of the pages the spammer already controls). This page U, when visited by a Web crawler, simply returns a 302 code redirecting the crawler to a reputable page V.

Now, in this situation, search engines until *circa* 2005 would consider U and V to be two identical mirrors of the same page, and more importantly, would pick arbitrarily one of the two URLs (U or V) as the canonical URL — the one to show to the user when showing that page as a result to a query. Using this technique, an attacker was able to lure to his Web site users searching for the content of the reputable page V, and upon receiving the visit from a normal user instead of a crawler, show spam content instead of a redirection.

3.5.3 Redirection Spam

Redirection spam is closely related to semantic cloaking. Instead of providing a different copy to the search engine at indexing time, a redirection is performed once a user arrives to a page. This redirection leads the user to a semantically different page.

This is usually accomplished by using a scripting language such as JavaScript to redirect the user to a spam site. Most search engines do not interpret all the scripts due to the high computational cost of doing so for every page.

Instead of trying to completely interpret the scripts, a search engine's crawlers may try to do some shallow parsing of the scripts to try to reduce the effect of Web spam.[1] Unfortunately, there are many code obscuring techniques that can be used in JavaScript and that hide malicious redirections [51]. Besides cataloging different code obfuscation techniques used by spammers, the authors advocate the use of redirection detectors based on lightweight JavaScript parsers operating in a controlled environment (a "sandbox") with an execution timeout.

In practice, the presence of obfuscated JavaScript code is often by itself a strong signal that a page is involved in spam [231]. Given the way spammers operate, through a few networks that "funnel" traffic to some sites by serving ads that lead to redirections, Wang et al. [231] argue that detecting those aggregators and funnels that do the redirection for large sets of pages is an effective way of eliminating massive amounts of Web spam with less effort than blacklisting individual spam sites.

3.5.4 Visual Cloaking

Finally, spammers may exploit the fact that Web crawlers typically do not render a Web page like a browser does (e.g., with JavaScript,[2] fixed resolution screen, etc.) to display contents that are different from the ones indexed by search engines.

An old, and generally useless approach today to visual cloaking is to make the spamming text of a page appear in the same color as the

[1] http://thenoisychannel.com/2009/07/29/sigir-2009-day-3-industry-track-matt-cutts/.

[2] In recent years, some search engines have been able to scan within JavaScript and can execute some JavaScript [70].

background or so small as to be unreadable. Such an approach, if successful, would have the effect of making the search engine index all of that content but only show the non-hidden content to the user. Similarly, one might use CSS to render some of the text in an area that was beyond one of the borders of the browser. A more modern approach would generate content for the user using JavaScript, or perhaps using an `iframe` showing the content of a different page to obscure the original indexed content underneath.[3]

Many of these attacks can be and are recognized by well-designed crawlers and spam classifiers. When the page's content is more complex, sometimes it is necessary to incorporate more aspects of a regular Web browser into the crawler's logic (e.g., [51, 170, 230]).

3.6 E-mail Spam Detection

Content-based Web spam detection techniques overlap with the methods used for e-mail spam detection, to the extent that both Web spam and e-mail spam detection can be described as text classification problems. Over the years, the e-mail spam detection community has developed several classification methods. In Section 7.3.2 we will see an example of an e-mail spam classifier being used to detect comment spam. The reader interested in the state of the art in e-mail spam detection, can read, e.g., a recent survey by Cormack [55], or start by looking at the entries in the Spam Track at TREC[4] and at research articles presented in recent editions of the Conference on E-mail and Anti-Spam.[5]

3.7 Conclusions

Since there are no restrictions on who can publish Web pages, a Web spammer can easily create a Web page and put whatever content is desired into it. This can include the repetition of terms or phrases to make the page rank highly for such queries, or arbitrary content so that

[3] See for example, Chellapilla's AIRWeb 2006 presentation available at http://airweb.cse.lehigh.edu/2006/.

[4] `http://plg.uwaterloo.ca/~gvcormac/spam/`.

[5] `http://www.ceas.cc/`.

the page might be considered relevant for many queries. Content might be copied from high-quality pages, or generated artificially, or might simply replicate a spam page that has already been copied thousands of times. Such content might be visible to the user, or hidden in some fashion (by placing the text off of the visible portion of the page, by obscuring it with other content, by cloaking depending on which client is requesting the page, or by redirecting the user to a new page).

In this section we have discussed such possibilities, and how researchers have worked to identify instances of such Web spam. Content spam is one of the oldest forms of search engine spam, and immunity to human tampering via content spam was one of the early claimed features of Google[6] given the introduction of the PageRank algorithm. However, once Google became popular, spammers started figuring out how to manipulate PageRank and other link-based methods, as we discuss in the next section.

[6] See for example, the Integrity paragraph at the bottom of a copy of Google's technology page from 2002: http://web.archive.org/web/20021203021211/http://www.google.com/technology/index.html.

4

Curbing Nepotistic Linking

Citation analysis is one of the key tools used in bibliometrics to assess the impact of an author, or of a document. The basic assumption is that citations in texts are not random, but that they indicate that documents are somehow related, and confer authority to the document being cited. Of course, several caveats can be mentioned: citations can be used to criticize as well as to praise, self-citations are frequent, some documents — e.g., methodological papers or surveys — attract a disproportionately large number of citations, citation patterns vary across disciplines, many citations on a text are irrelevant, and so on. Last but not least, authors who obtain a benefit from having high citation scores have an incentive to try to "game" bibliometrics, for instance, by citing each other frequently in a nepotistic way (independent from merit), forming a "mutual admiration society" through their citations.

The Web is much more open than traditional publication, and several forms of citation analysis are used extensively to rank documents. Hyperlinks can be created on the Web essentially for free, and all the standard link-based ranking algorithms such as counting in-links [145], computing PageRank [183] or running HITS [122] or SALSA [141] are trivial or easy to game, unless counter-measures are taken.

Because of this, three research problems have attracted a considerable share of the research effort of the Adversarial IR community: (i) developing methods to detect spamming aimed at link-based methods, (ii) determining to what extent that spamming was able to successfully boost ranking, and (iii) studying how to make the link-based ranking methods robust to manipulation. This section deals with these topics. Other graph-related topics such as trust and distrust are addressed in Section 5.

4.1 Link-Based Ranking

This section briefly describes the fundamental link-based ranking methods used on the Web.

4.1.1 PageRank

PageRank [183] is an estimate of the importance (or equally, authority or reputation) of a Web page. It is arguably the most successful link-based ranking method, as demonstrated by the Google search engine. Currently, most search engines probably use some form of PageRank-style computation for ranking, but in practice its contribution to the final ordering of pages is believed to be in general small compared to other factors [201]. Nevertheless, PageRank is well known by the search engine optimization and marketing communities and by spammers.

The computation of PageRank is relatively straightforward. We start with a graph $G = (V,E)$ representing Web pages $V = \{1,2,\ldots,|V|\}$ and hyperlinks $E \subseteq V \times V$. Then, this graph is represented in a matrix $\mathbf{M}_{|V|\times|V|}$ with

$$m_{ij} = \begin{cases} 0 & \text{if } (i,j) \in E \\ 1/\text{out-degree}(i) & \text{otherwise.} \end{cases} \tag{4.1}$$

Next, a new matrix $\mathbf{P}$ is derived from $\mathbf{M}$ by adding links with a small weight from every node to all other nodes in the graph. This means that $\mathbf{P}$ represents a graph that is similar to $\mathbf{M}$, but has the property of being *strongly connected*: there is always a directed path between any pair of nodes. The matrix $\mathbf{P}$ is irreducible, and it is given by:

$$\mathbf{P} = \frac{\epsilon}{|V|}\mathbf{1}_{|V|\times|V|} + (1-\epsilon)\mathbf{M}, \tag{4.2}$$

where $\mathbf{1}$ is a matrix containing only ones, and ϵ is a small value, typically 0.1 or 0.15 as proposed by the authors of the PageRank paper [183]. Next, the PageRank of a page i is the i-th component of the eigenvector of $\mathbf{P}$ associated with its largest eigenvalue.

PageRank has been studied extensively due to its simple formulation. For surveys on many aspects of PageRank, see Langville and Meyer [138, 139]. For a survey specifically on how to compute PageRank values efficiently, see Berkhin [23].

4.1.2 HITS and SALSA

The HITS (*Hyperlink-Induced Topic Search*) algorithm proposed by Kleinberg [122] is another method for ranking Web pages. It starts by building a set of pages related to a topic by querying a search engine, and then expands this set by identifying and retrieving incoming and outgoing links. Next, two scores for each page are computed: a hub score and an authority score. Intuitively, a page has a high authority score if it is pointed to by pages with a high hub score, and a page has a high hub score if it points to many authoritative resources.

The IMP (*imp*roved) algorithm proposed by Bharat and Henzinger [25] is an extension to HITS that attempts to eliminate the effect of mutually reinforcing relationships in HITS by considering only external links (removing the links between pages in the same site) and by re-weighting the edges in a manner slightly more complicated than that of PageRank. In particular, it adjusts the link weights such that the weights of edges of multiple source pages on a single host that point to a single target page on a different host sum to one when calculating authority, and that the weights of links from a single document that point to a set of pages on the same target host to also sum to one when calculating hubs.

SALSA [141] (*Stochastic Approach for Link-Structure Analysis*) builds an expanded set as in HITS, re-weights the edges on the basis of the in and out-degrees of the source and target pages, and then does an alternating random walk on this set of pages, by following in alternating order a link forward and a link backward in the sub-graphs induced by the selected pages. The ranking induced by this algorithm is equivalent to in- and out-degree when there are no weights and the expanded set is connected [31].

4.2 Link Bombs

A link bomb is the cooperative attempt to place a Web page in the result list for a (typically obscure) search query. The specific target engine, Google, leads to the much more common (and original) name for this technique: *Google bombing.*[1] While there can be different motivations, including humor, ego, or ideology [15], the approach exploits the use that search engines make of in-links and anchor text for ranking. The link bomb organizer will typically attempt to convince many Web page authors to use particular anchor text in a link to a particular target, so that the target ranks highly for a query corresponding to the anchor text. In many cases, link bombing is considered to be a form of online protest, and is effectively "creating alternate constructions of reality through collective action online" [220].

This particular type of spam has earned publicity a number of times in the popular press (e.g., [143, 163, 175]). An early publicized attempt, and the source of the term Google bombing was Adam Mathes' 2001 Blog post encouraging the creation of a Google bomb for a friend's blog using the phrase "talentless hack" [162]. While for many years, most Google bombs were created for humor, some, such as the competition for the query "jew" [14] were more worrisome. By 2006, the technique was being used for political advantage [118]. There have been many instances of this type of search engine *hacktivism* since.[2] As a result, Google finally addressed the issue with a revision to their ranking system [171].

Finally, link bombing is often the subject of questionable SEO contests, such as the one sponsored by the firm Dark Blue in 2004. The

[1] The phrase Google bombing has even been included in the second edition of the New Oxford American Dictionary [191].

[2] http://blogoscoped.com/googlebomb/.

goal of this contest was to rank highly for the phrase “nigritude ultramarine” [198], a pair of nonsense word derived from the name of the sponsor. There have been many such contests over the years, often demonstrating the lengths to which some SEO practitioners will go.

4.3 Link Farms

A *link farm*[3] is a set of pages linked together with the objective of boosting the search engine ranking of a subset of those pages. The pages for which the spammer wants to increase the ranking are *boosted* pages, while the pages used to that end are *boosting* or *hijacked* pages, depending on whether they are legitimately or illegitimately under the control of the spammer [91]. The latter can be the case of a publicly writable Web page (such as a blog page which accepts comments), in which a spammer may post a comment containing an out-link to participate in a link farm. (There are specific methods to reduce the effect of spam in publicly writable pages described in Section 5.2.)

From the perspective of those manipulating the search engine ranking, we may consider two types of link manipulation that are different in principle: Sibyl attacks and collusion attacks [52].

In a *Sibyl*[4] attack, there is a single attacker (an individual or a company) that creates multiple identities, in this case multiple Web pages or sites, not easily identifiable as belonging to the same individual [64]. The purpose of these pages or sites is to boost the ranking of a subset of those pages belonging to the attacker or to a third party that hires the attacker for this purpose.

In a *collusion* attack, a group of individuals or companies agree to mutually link their Web pages in a manner that is independent of the quality or relevance of each other’s resources (e.g., as is the case in many link exchange sites). This has been termed a mutual admiration society [164], and its purpose is to boost the ranking of at least one page per participant.

The difference is that in the case of the sybil attack, a single page could be boosted even at the detriment of the ranking of all the other

[3] Confusingly, a link farm is also sometimes referred to as a “link bomb”.

[4] Named after the split personality case of Sybil Dorsett.

pages that have been manipulated. In the case of a collusion attack, each participant must obtain some benefit out of participating. In other words, in the sybil attack the benefits can be either concentrated or spread across several sites, while in a collusion attack the benefits cannot be concentrated on a single site (or the other participants would not have incentives to participate). Thus link farms created by collusion are a more restricted form of link farms than those created by a sybil attack.

In the case where the objective of the spammer is to boost the PageRank of a single page, the best strategy is to have all boosting and hijacked pages link to that target [91]. Depending on the presence of other constraints, variations of this linking pattern can be employed [5, 66], particularly if the spammer wants to avoid being detected.

If the rest of the Web does not link to any of the attacker's boosting pages, then there is no point in creating a complex structure in terms of increasing the PageRank [54]; in order to substantially change the PageRank of a target, out-links from pages linked to by the rest of the Web must be created.

If these out-links cannot be obtained, then many new pages have to be created to exploit the "random jump" factor of PageRank. However, this imposes a cost in an attack of a domain-level PageRank, as Web pages can be created easily, but domain names need to be purchased.

Different link farm structures are studied empirically in [12, 247] comparing the PageRank gain of forming a clique or quasi-clique versus other structures such as a star or a ring. The clique yields a much higher PageRank gain but it might be much easier to detect than other structures; it might also be impractical from the Web site design perspective if the number of participants is too large.

In the specific case of collusion or semi-collusion [35] (autonomous agents cooperating in some aspects, but competing in others), by introducing some constraints the process of link farm construction can be modeled formally as a game [106]. One such constraint can be, for instance, that each page has a limited amount of space to place prominent out-links (those with some chance of being clicked by users). In this game, the objective of the players is to maximize the expected revenue, that is, per-link revenue times number of visits, for the subset of pages under their control. For the players, actions in the game correspond to placement of links in the subset of pages under their control. Several variants of this game are further discussed by Immorlica et al. [106]. More generally, the creation of links in the Web or any competitive network can be modeled as a game [219]. Using game theory, Hopcroft and Sheldon [103] specifically consider how a reputation system (e.g., PageRank) can affect the dynamics of link formation and thus the structure of the network. They find that even when the participant models are simple (and selfish), different reputation measures can lead to dramatically different outcomes.

4.4 Link Farm Detection

The methods for link farm detection often search for anomalous patterns within the interconnection graph of the Web. The huge size of this graph reduces the class of feasible methods. A popular class of methods which are considered practical in large-scale applications is that of semi-streaming graph algorithms: methods that require $O(|V|)$ bits of main memory and work by doing a small number, typically $O(\log(|V|))$, of sequential scans over the edges of the graph [72].

4.4.1 Detecting Dense Sub-graphs

Link farm detection methods are usually aimed at finding dense sub-graphs, a problem for which no efficient exact solutions are known, but for which there are several approximate algorithms.

One such algorithm is the method for dense sub-graphs described by Gibson et al. [79], which is based on hash sketches [36]. The general method is similar to the one described in Section 3.4 for finding near-duplicate contents; the basic idea is to compute hash sketches over subsets of elements on a large set, and then use those sketches to quickly estimate the sizes of set intersections. Instead of using sequences of n words as we did for contents, in the case of graphs we use sequences of n links.

The algorithm in [79] works as follows: first, the graph is represented as an adjacency list, and groups of n nodes from each adjacency list are converted into a hash sketch. Next, an inverted index of sketches is created; this is a list of ordered sketches in which each sketch s is associated with a list of nodes in the graph, in whose out-links the sequence of nodes represented by s can be found. Finally, to find dense sub-graphs the posting lists of each sketch are scanned and groups of n nodes are sketched again. The sets of nodes associated with values present in frequent second-generation sketches are good candidates for members of dense sub-graphs.

Another algorithm that can be used for detecting dense sub-graphs is the Approximate Neighborhood Function (ANF) [184], an estimation of the neighborhood size of a set of nodes in a graph, obtained using probabilistic counting. An ANF-plot is the plot of the neighborhood size of a node for different distances, and a dense sub-graph should appear as a growing number of in-neighbors at short distances that later "stalls" when reaching the boundary of the link farm [19].

Zhang et al. [247] define the *amplification factor* of a set of nodes as the ratio between the current sum of their PageRank scores, and the sum of their PageRank scores they would obtain if the links between their members were removed. They show that the amplification factor is bounded by $1/\epsilon$ in which ϵ is the random jump factor, typically 0.15; a tighter bound is shown in [12]. Interestingly, the amplification factor can be used as a link farm detection strategy: the PageRank is computed with different ϵ parameters, and then the nodes that have a high PageRank and whose PageRank is strongly anti-correlated with ϵ are suspicious of belonging to a link farm.

Wu and Davison [241] discover link farms by first finding a candidate set of pages whose in-links and out-links have a sufficient number of domains in common (a kind of dense sub-graph). This list of candidates is then expanded by finding pages with sufficient links to confirmed cases of spam. In [240], the authors heuristically look for dense sub-graphs (specifically those pages with many examples of anchor texts and targets in common), and consider those found to be examples of plagiarism.

A completely different algorithmic scheme can be found in Yu et al. [245] which uses *random routes*. In a random walk, every time the random walker arrives at a node he chooses at random which out-link to follow. In a random route, there is a random, but fixed, permutation applied to the links of all nodes, which uniquely determines the outgoing edge given the incoming edge. This also means that a random route can be traversed backward without actually storing the route. In the SybilGuard method proposed in [245], a node v accepts a node u as legitimate if, after a few trials, a random route starting at u intersects a random route from the verifier node v.

4.4.2 Detecting Anomalous Sub-graphs

Another approach is to look for anomalies in general, instead of specifically dense sub-graphs. In addition to other signals, Fetterly et al. [74] examined the distribution of in-degrees and out-degrees, finding values well beyond the expected Zipfian distribution that corresponded significantly to spam. In Becchetti et al. [18] a number of link-based features are extracted from a set of nodes, including degree, average degree of neighbors, edge reciprocity, etc. Using these features, an automatic classifier of spam sites is learned using a large set of training examples. Among the features used by these classifiers, a particularly useful set are neighborhood sizes at short link-distances [19, 184], different variants of PageRank [183], and TrustRank [94] which is described in Section 5.3.

Benczur et al. [21] study the distribution of PageRank scores in the neighborhood of a page. Their method includes identification of suspicious candidates, candidate set expansion and penalization of the contributing nodes. Suspicious nodes are identified by looking at regularities in the distribution of the PageRank of the in-neighbors of a node. The more uniform this distribution is, the more likely the links are placed automatically and are part of a link farm.

Da Costa-Carvalho et al. [59] use several site-level (e.g., host-level) link-based heuristics to detect anomalous linking patterns. Among the patterns they detect are: mutually reinforcing sites (having high link-reciprocity at the page level), sites that are responsible for a large fraction of the in-links of another site, sites whose in-neighbors have an abnormally high fraction of links between them, and so on.

Caverlee and Liu [46] measure the “credibility” of a page by looking at the quality of its out-neighborhood; they do this by simulating short random walks starting from the page to be studied. Each page is then assigned a credibility score reflecting how close and how dominant spam pages are in the pages reachable from the current page.

Zhou et al. [253] consider the calculation of “spamicity” in an interactive browser-based setting. They consider the set of pages that support (have a path to) the page in question, and consider pages whose supporting page farms are sufficiently effective to be spam.

The methods that perform structural analysis of the link graph can often also be used to find spam in social networks [174] (see Section 7 for more).

4.4.3 Detecting Abnormal Link Change Rates

According to Ntoulas et al. [181], the rate of change of links on the Web is very fast, even faster than the rate of change of contents. Shen et al. [209] study temporal link-based features. These include the rate of growth and death of new in-links and out-links from the perspective of entire sites. They show that in practice for spammers these change rates are abnormally high while the link farm is being created, and this observation can be used to improve the accuracy of a link-based spam classifier.

4.5 Beyond Detection

The previous section describes methods to detect link farms. This section describes methods that reduce the effect of link farms without trying to determine their boundaries explicitly. This includes finding specific instances of nepotistic links (Section 4.5.1), or reducing the effect of nepotistic linking on ranking by considering groups of pages as a single entity (Section 4.5.2), or by reducing the effect of short cycles (Section 4.5.3).

4.5.1 Removing or Down-Weighting Links

One might consider the detection of nepotistic links as a form of data cleansing. After detecting suspect link structures, a fairly common approach is to remove anomalous links before estimating authority on the nodes of the remaining graph [20, 59, 196, 241].

Removing or down-weighting nepotistic links can also be used as a measure that is not as drastic as removing or demoting nodes, and that can be used in the cases in which the classifier has not enough confidence on a spam prediction. In these borderline cases, which are many, a natural way of dealing with link farms is to down-weight the links, enabling the content to still be visible but (hopefully) reducing the effect of spam (e.g., as in [240]).

4.5.2 Lumping (Merging) Nodes

On the Web, a new page can be created almost for free, which implies that the "one-page, one vote" paradigm [145] may not be truly appropriate. A natural way of making link-based ranking methods robust against the creation of multiple pages managed by the same entity is by considering all those pages as a single node in the graph. In Markov-chain theory this is usually referred to as the process of "lumping" some states in a Markov chain.

As a concrete example, we can use the fact that a domain name has a monetary cost (even if it is only a few dollars per domain) to design a link-based ranking method that considers all the pages in a domain as a single node. In this way, we effectively ignore the links among pages in the same domain, as well as multiple links from one domain to another.

In the case of PageRank, Caverlee et al. [49] study methods for lumping together nodes in a graph into sets of nodes for reducing spam. These sets can be arbitrary, for instance based on domain names. If domain names are used to group nodes, a link from a page u in domain d_u to a page v in domain d_v, is turned into a hyperlink connecting d_u to d_v, and is weighted proportionally to the number of inter-domain links starting from d_u. This effectively reduces the effect of page-level link farms and generates a domain-level ranking in this case.

Berlt et al. [24] study a different method motivated by the same observation as the one above. They turn the Web graph into a hypergraph in which links do not connect a node to another node, but a set of nodes to another node or to another set of nodes. If for instance domain names are used to group nodes, a link from a page u in a domain d_u to a page v, is turned into a hyperlink connecting the set of all pages in d_u to the page v. This counts all the links between pages in a domain to a page in another domain as a single link, and still can provide a page-level ranking.

A related alternative is to split the influence of any given group of pages; in short, having k out-links may mean that each out-link gets $1/k$ of the score mass. Both of these concepts are introduced in the improved [25] version of HITS. Roberts and Rosenthal [202] further improve HITS by first clustering the set of candidates using a link-based clustering method. Next, only the links between nodes in different clusters are considered in the computation. If a randomized clustering method is used, this process can be repeated a few times using different clusterings, and then the scores can be averaged across all the runs.

4.5.3 Reducing the Effect of Short Cycles

Hopcroft and Sheldon [102] propose a link-based method that is based on the same random walk as PageRank, but ranks nodes according to the expected *hitting time* from the restart distribution instead of by their probabilities in the stationary state. The authors show that this is more resistant to manipulation than PageRank given that a node u cannot influence its own ranking (e.g., by placing out-links to other nodes that have short paths linking back to u).

4.6 Combining Links and Text

Most comprehensive spam detection approaches combine both the analysis of links with content-based analysis from Section 3.

The simplest way to incorporate both content and link information is to provide them both as features directly to the classifier. Simple features of this type might include the number of out-links to the same site and the number to different sites.

A slightly more involved method examines such features at the neighbors of a node, and provides those features (in aggregated form) to the classifier. For instance, among other features Drost and Scheffer [65] computed the feature “number of tokens in title” of a page p. This means they also compute as features for p: the average number of tokens in the titles across all the in-links of p, and the average number of tokens in the titles across all the out-links of p. They used both average and sum as aggregate functions. After a feature selection procedure, they show that many of these features, computed from the neighborhood of a page, rank among the most important ones in the classifier.

Instead of propagating features, one could instead propagate classifications. Castillo et al. [44] include link-based and content-based features in a classifier to produce a base prediction, then describe multiple methods for post-processing this prediction considering the graph structure. The first method is to propagate by averaging the base prediction across the neighbors of a node. A second method is to cluster the graph (for instance, using METIS [120]), and then give to all the nodes in each cluster the same label (spam or non-spam) using majority voting. A third method, which outperforms the others, is to use stacked graphical learning [131], which is a fast way of incorporating information from neighboring nodes in a classifier at a small computational cost.

Gan and Suel [78] explore a related method in which a base classifier is learned, and then the predicted class for a node is refined by doing a weighted majority voting of the predicted classes of the nodes linked from it (but without re-training as in stacked graphical learning). In the case of a host graph, the weights can be for instance the number of page-level links between two hosts. The weighted majority voting can use only the in-links or only the out-links of the page. A related method that yields a larger improvement is to use a second classifier that uses as features the prediction from the base classifier (but not its features as in stacked graphical learning) and statistics about the predictions in the neighborhood of a page.

Finally, one might optimize the classification of all nodes across a graph simultaneously in which neighbors are *expected* to have similar classes. Abernethy et al. introduce WITCH (Web Spam Identification Through Content and Hyperlinks) [2, 3, 4]. The classification is performed using an SVM that includes graph regularization and slack variables. The SVM receives as input a set of labeled examples $(\mathbf{x}_i, y_i)$

for $i = 1 \ldots \ell$; where vector $\mathbf{x}_i$ contains the feature values for page i, and y_i is a label provided by a human editor. We define this label to be $+1$ for spam pages, and -1 for non-spam pages. The goal is to learn a linear classifier whose prediction is given by $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$. In a standard SVM, vector $\mathbf{w}$ contains the parameters of the SVM, which are learned by minimizing the following function:

$$\Omega(w) = \frac{1}{\ell} \sum_{i=1}^{\ell} R(\mathbf{w} \cdot \mathbf{x}_i, y_i) + \lambda \mathbf{w} \cdot \mathbf{w},$$

where R is a *loss function* that penalizes the difference between the prediction and the actual label, for the ℓ examples for which labels are available. For instance R can be the difference in absolute values, but other loss functions can be used. The second term is a *regularization term*, controlled by parameter λ that prevents the coefficients of $\mathbf{w}$ from getting too large, which would produce overfitting.

Graph regularization is included by an extra term that accounts for the graph structure:

$$\Omega(w) = \frac{1}{\ell} \sum_{i=1}^{\ell} R(\mathbf{w} \cdot \mathbf{x}_i, y_i) + \lambda \mathbf{w} \cdot \mathbf{w} + \gamma \sum_{(i,j) \in E} a_{i,j} \Phi(\mathbf{w} \cdot \mathbf{x}_i, \mathbf{w} \cdot \mathbf{x}_j),$$

where γ controls the aggressiveness of the graph regularization, whose cost is computed over all the links $(i, j) \in E$. The coefficients $a_{i,j}$ are weights for each link (e.g., the count of page-level links between two hosts when we are operating in the host graph), and $\Phi(\mathbf{w} \cdot \mathbf{x}_i, \mathbf{w} \cdot \mathbf{x}_j)$ is a cost incurred by the classifier when it predicts a different label for nodes i and j connected by a link. A natural choice for $\Phi(\cdot, \cdot)$ is

$$\Phi(f_i, f_j) = (f_i - f_j)^2,$$

which measures the square of the difference between the two predicted labels. However, a better choice is to use

$$\Phi(f_i, f_j) = max(0, f_j - f_i)^2,$$

which penalizes the case of a non-spam page linking to a spam page, but not the converse. Actually the best penalization in [4] involves giving a large penalization every time it is predicted that a non-spam page links to a spam page, and giving a small penalization every time it is predicted that a spam page links to a non-spam page.

Building a graph-regularized classifier involves many design choices: how much importance to give to the regularization term (parameter γ above), how to describe the cost of predicting different labels for nodes connected by an edge (function Φ above), how to weight different edges (parameters a_{ij}), etc.

4.7 Conclusions

To have a significant effect on a link-based ranking method that estimates authority, a spammer needs to coordinate many links in the form of a link farm. Most of the anti-spam methods described in this section aim at making this process more difficult or ineffective, by re-weighting or removing suspicious links, or by changing the unit of influence from the page to host, among other techniques.

In this section we have introduced the fundamental link analysis algorithms. Variations of these approaches are assumed to be utilized by the major engines to estimate the importance of pages and sites, and then used as one of many factors for result ranking. As a result, link-based measures continue to be a significant target of Web spam, and motivate many of the attacks described later in Section 7.

5

Propagating Trust and Distrust

An aspect of link analysis on electronically mediated communications that have attracted a considerable amount of research is the study and inference of trust relationships. These methods are related to the better-known authority propagation methods discussed in Section 4, but are different because they are given a set of confirmed trustworthy and untrustworthy agents as inputs.

Trust propagation methods employ the labeled agents in a way that tends to match the heuristics that we apply in our social lives. For instance, in the case of untrustworthy agents, we try to apply the "guilt-by-association" heuristic; while in the case of trustworthy agents, we try to apply the "a friend of a friend is my friend" heuristic.

On the Web, the input sets can be thousands of hand-labeled pages or sites, and the propagation can occur through forward or reverse hyperlinks. Trust-aware methods such as the ones discussed on this section have been shown to be successful at countering ranking manipulation on the Web.

5.1 Trust as a Directed Graph

The concept of a "Web of Trust" was first introduced in large-scale systems during the design of key-management protocols for PGP

(*Pretty Good Privacy*) [256]. A Web of Trust is a directed graph where nodes are entities, and arcs indicate a trust (or distrust) relationship between two entities.

The Web of Trust in a large community tends to be very sparse. Any given agent interacts only with a small fraction of the members of the community, and thus can only assess the trustworthiness of a handful of other agents. A natural way of alleviating this sparsity problem is to aggregate the ratings given by several people, usually through the use of some sort of propagation mechanism.

According to the taxonomy presented by Ziegler and Lausen [255], there are two basic types of trust computation: local and global. In a *local trust computation*, trust inferences are done from the perspective of a single node, and thus each node in the network can have multiple trust values. In a *global trust computation*, trust inferences are computed from the perspective of the whole network, and thus each node has a single trust value. In both local and global trust scenarios, the computation can be either centralized or distributed among a number of peers.

In the specific case of trust for Web search, under current technologies the most relevant case is global trust propagation computed in a centralized manner. Guha et al. [89] study global methods for propagation of trust and distrust in a systematic manner. Let $G = (U,T)$ represent the explicit trust ratings, with a set of users U, and let T represent a trust relationship, where $T \subseteq U \times U$ with $(u,v) \in T \iff u$ trusts v. Let $S \subseteq U \times U$ represent the *implicit* trust between users that is inferred by the system, so that $(u,v) \in S$ implies that given the available evidence, u should trust v.

To build the relationship S, we start obviously by considering $(u,v) \in T \Rightarrow (u,v) \in S$. Next, Guha et al. note four different propagation types:

- Direct (transitive) propagation: $(u,v) \in T \wedge (v,w) \in T \Rightarrow (u,w) \in S$
- Co-citation: $(u,v) \in T \wedge (u,w) \in T \wedge (s,v) \in T \Rightarrow (s,w) \in S$
- Transpose propagation: $(u,v) \in T \wedge (w,v) \in T \Rightarrow (w,u) \in S$
- Trust coupling: $(u,v) \in T \wedge (w,v) \in T \wedge (s,u) \in T \Rightarrow (s,w) \in S$.

In most of the research we describe next, trust propagation is not binary, but is real-valued. Direct (transitive) propagation occurs, but most approaches will degrade it by some amount. Although Guha et al. found that all propagation types were useful in a combined trust propagation system, most of the work described here focus on the use of direct propagation on the Web graph, the reverse Web graph, or both.

5.2 Positive and Negative Trust

In many communities the base assessments from which trust is computed include both positive (u trusts v) and negative (u distrusts v) assessments. However, most of the research focuses on the propagation of trust, and much less on how to deal with distrust. The reasons are threefold.

First, the semantics of trust propagation ("the friend of a friend is my friend") are clear and effective in practice, while the semantics of distrust propagation ("the enemy of my enemy is my friend") have been shown to be less effective in practice. For instance, according to the results of Guha et al. [89], a good method for global trust computation uses an iterative (multi-step) direct propagation of trust, but only a single-step direct propagation of distrust.

Second, in many communities positive assessments are dominant, as people are much more cautious when providing negative judgments for fear of retaliatory negative feedback, or simply to avoid further unpleasant interactions [200]. This means that in some cases the absence of a trust rating after an interaction cannot be considered automatically as a neutral rating.

Third, in the case of the Web in particular, there are no labels on the edges that allow the separation of hyperlinks indicating trust from those that might not reflect trust. To alleviate this, there are two proposals to annotate hyperlinks, both involving adding an attribute to the hyperlink XHTML tag <a>. VoteLinks [157] suggests to annotate hyperlinks with `rev="vote-for"`, `rev="vote-against"`, and `rev="vote-abstain"` to indicate respectively positive, negative, and neutral opinions. The `nofollow` proposal [156] which presently is more used in practice suggests that hyperlinks indicate trust except when they are annotated with `rel="nofollow"` where they should be considered neutral in the sense of conferring trust. See Section 7 for more discussion of nofollow.

Given that the `nofollow` tag is used but not widespread, learning to recognize the polarity (positive, negative, or neutral) of a link is key. In experiments by Massa and Hayes [161], over the Epinion community, in which each opinion can be seen as a link, they observed a substantial disagreement between the scores obtained by computing PageRank considering both positive and negative links, compared to considering only the positive links.

5.3 Propagating Trust: TrustRank and Variants

TrustRank is a well-known trust propagation mechanism for Web pages proposed by Gyöngyi et al. [94]. The TrustRank method uses a small seed set of non-spam (trustworthy) pages that are carefully selected by human editors. Next, a random walk with restart to the seed set is executed for a small, fixed number of iterations. In [94], the restart probability is the same (0.15) as in the original PageRank paper [183], and the number of iterations is 20. TrustRank has been shown to be very effective in demoting spam pages in the original paper as well as in later studies.

A closely related concept is the *relative spam mass* [96] of a node. It is defined as the fraction of its PageRank contributed by spam nodes. Given that assuming *a priori* knowledge of which are the spam nodes is unrealistic, the relative spam mass of nodes has to be estimated. A method for estimating the relative spam mass of nodes is to compute a *(good)-core-based PageRank*, which is basically a TrustRank score computed over an order of magnitude larger seed set. The seed set for the spam mass estimation should include not only the highest quality nodes, but many diverse non-spam nodes. The relative spam mass is estimated as the PageRank of a node minus its score as obtained using this procedure.

Several variants of the original TrustRank can further improve its efficiency. TrustRank scores tend to be biased toward large communities representing popular topics on the Web. Topical TrustRank [243] tries to alleviate this problem by computing several independent topic-dependent TrustRanks for each page, starting with a topic-specific seed set in each run. This is inspired by the way in which Haveliwala [97] computes topic-dependent PageRanks.

Another improvement is to step out of the random-walk paradigm and look at the TrustRank computation as an iterative scoring function and not as a Markov process. In this sense, Wu et al. [242] and Nie et al. [178] propose alternative ways of “splitting” the trust mass of a node among its out-neighbors, and of aggregating the trust mass received by the in-neighbors of a node. While the original formulations of TrustRank and PageRank divide this score by the out-degree of a node, there are alternatives. For instance, the score can be divided by the logarithm of the out-degree, or not divided at all. For the aggregation of the trust mass received, the nodes can use a summation, as in the original formulation, or take the maximum trust received from an in-neighbor, or take the sum capped to be at most the maximum trust received from an in-neighbor. Their results show that these alternatives improve over the original formulation in terms of demoting spam.

Finally, seed selection is another important aspect to take into consideration when using TrustRank. Under certain conditions, an automatically selected large seed set (which may contain a few errors) is preferable to a manually selected cleaner, but smaller, seed set [110]. Zhao et al. [250] go further, detailing a semi-automatic mechanism to find both good and bad seeds for use in detecting spam.

5.4 Propagating Distrust: BadRank and Variants

The opposite of TrustRank is known in the SEO community as “BadRank” [214]. The intuition behind it is that while the in-links of a page are not under the control of its author, the out-links of a page can be edited freely by the author and thus creating a link to a spam page means participating in the spamming activity. There is strong evidence that indeed, non-spammers do not link to spammers in general [44]. BadRank, also known as “anti-TrustRank” can be implemented as a random walk that follows links backward, and restarts to a known set of spam nodes; the “badness” of a page is its probability in the stationary state of this random walk. It has been shown experimentally to be effective in detecting spam pages in [135].

In general, once a group of Web pages or hosts has been confirmed to be spam, it makes sense to attempt to automatically find the other pages, hosts, or domains participating in the spamming activity. Also, in practice in a large search engine a group of assessors (editors) may help in the labeling of spam sites in a semi-automatic setting. For this, a system must present a set of candidates to a human operator. In both cases, we need ways of expanding a set of confirmed spammers into a set that is very likely to be spam (that we can automatically label to be spam with high confidence) and a set of very suspicious sites (that can be presented to a group of editors for confirmation).

For link farms, given a suspicious node, the nodes contributing a large share of their PageRanks can be detected using a greedy method [252], and the properties of this group can be analyzed to classify it as a link farm or not. Similarly, Andersen et al. [8] present an efficient approximate algorithm for computing the δ-contributing set of a node v, which is defined as the set of nodes that contribute at least a δ fraction of v's PageRank. Their algorithm examines a small subset of nodes, $O(1/\delta)$.

Another automatic expansion method is to use SimRank for spam detection as suggested by Benczúr et al. [1]. SimRank is a generalization of the co-citation and can be used as a feature for a spam classifier, as a page that has a high link similarity (as measured by SimRank) to a spam page is likely to also be a spam page.

Random walks can also be used to expand a set of known spam pages. Wu and Chellapilla [237] start with a given set of confirmed spam nodes and then walk randomly to find other nodes that might be involved in the same spam activity.

Metaxas and DeStefano [164] suggest a graph-based method in which the in-links of a set of confirmed spammers are followed recursively for a few levels, and then all the nodes in the strongly connected component containing the confirmed spammers are labeled as spam. Other expansion procedures, including a triangle-walk method that expands a suspicious set of attackers while they form triangles, are described in [210]. Similarly, Wu and Davison [241] perform a discretized propagation — that is, new pages are only added to the set of spamming pages if there is sufficient evidence, but once added, may push other pages above the threshold.

5.5 Considering In-Links as well as Out-Links

A number of researchers have proposed methods in which both in-links and out-links are taken into account, propagating both trust and distrust [178, 242, 248]. Zhang et al. [248] describe two interrelated propagation process, which depend on each other, and that propagate scores through in-links and out-links.

Specifically, two values are computed for each page p: one value $Q_c(p)$ depends on in-links, and is computed iteratively using the following update rule:

$$Q_c(p) = \sum_{p' \to p} \left(\alpha \frac{Q_c(p')}{o(p')} + (1-\alpha) \frac{Q_\ell(p')}{o(p')} \right),$$

where $o(p')$ is the number of outlinks of page p'. The other value $Q_\ell(p)$ depends on the out-links, and is also computed iteratively as:

$$Q_\ell(p) = \sum_{p \to p'} \left(\alpha \frac{Q_c(p')}{i(p')} + (1-\alpha) \frac{Q_\ell(p')}{i(p')} \right),$$

where $i(p')$ is the number of in-links of page p'. The values $Q_c(p)$ and $Q_\ell(p)$ are initialized using a list of known spammers, which are assigned a value close to -1, and a list of known non-spammers, which are assigned a value close to $+1$.

5.6 Considering Authorship as well as Contents

A subtle but interesting aspect of trust on the Web is that most models deal with assessing the trustworthiness of an entity that produces content (the author of a Web site for instance), while the actual goal is to determine how trustworthy a piece of content is [80]. For instance, even in the absence of trust assessments, a piece of information that is repeated by several independent sources can be considered trustworthy; by the same reasoning, a piece of information posted by a reputable source may be considered untrustworthy if it is contradicted by a large group of independent sources of lower trustworthiness.

This type of concern is particularly relevant when examining documents having multiple authors. For instance, Wikipedia articles are authored by many volunteer editors and a reader might be interested in knowing how trustworthy a particular passage of an article is. Mc Guinness et al. [90] annotate sentences using the reputation of their authors as source information; the reputation of authors is obtained by looking at the citations of the articles in which they participate. Adler and de Alfaro [6] and Hu et al. [104] also look at the trustworthiness of passages of Wikipedia considering that a passage written by a user u that remains unchanged after an edit of another user v, might be considered as "approved" by user v and thus can be considered more trustworthy.

With rare exceptions [7], authorship of general Web pages cannot be established easily at this time, but if widely accepted mechanisms for indicating authorship develop over the years, the issue of computing content-level trust from entity-level indicators will become more relevant in practice.

5.7 Propagating Trust in Other Settings

There are other environments in which trust propagation has been studied; they include online social networks, e-mail networks, and peer-to-peer (P2P) networks. Several methods of trust propagation in online social networks are described in Section 7.4.1. Methods for trust propagation in e-mail and P2P networks are related but not central to the topic of this survey, so we provide only a few pointers in this section.

Trust propagation in e-mail networks is studied, among other authors, by Boykin and Roychowdhury [32] where the sub-graph induced by legitimate and spam e-mail messages are shown to be clearly different. A related study is due to Gomes et al. [83].

Trust propagation in a P2P network requires decentralized trust computations to establish the quality of the files offered by each peer for download. A taxonomy of P2P reputation systems is introduced by Marti and Garcia-Molina [158]; this taxonomy considers factors such as how the information is gathered and aggregated and what the actions taken by the system are with respect to inauthentic peers. A well-known example of a mechanism for trust computation is EigenTrust [119] which is shown in simulations to reduce the ratio of inauthentic downloads in P2P networks with malicious peers. Other approaches include PeerTrust [244], Credence [228], and JXP [186].

5.8 Utilizing Trust

As discussed in Section 2.4.3, often detecting spam or providing an estimate of trust is only a means to an end. The output of these methods needs to be utilized to achieve a more complex goal, such as improving ranked retrieval in Web search.

Nie et al. [176, 177] show one way of utilizing trust estimates in retrieval systems. After a page-level trust metric has been computed, they integrate it into calculations of authority (e.g., PageRank or SALSA). They use the trust value to affect the probability of following outgoing links (e.g., emphasizing the "votes" of trusted nodes) and to affect which next node to select (i.e., a non-uniform selection of out-links, or for random jumps in PageRank). The authors demonstrate improved retrieval performance with their approach when utilizing trust estimates computed from multiple algorithms, including TrustRank.

5.9 Conclusions

To some extent, *truly* authoritative pages are also trustworthy. However, as we have seen in Section 4, estimates of authority can be significantly affected by nepotistic links. Thus, in this section we have seen that it is beneficial to explicitly consider trustworthiness, and how graph locality permits estimates of trust to be calculated based on the trustworthiness of known peers. Such trust information can be used to identify or to demote spam pages, or integrated into ranking algorithms.

6

Detecting Spam in Usage Data

Usage information has attracted considerable attention in the research community in recent years, and it is one of the current frontiers in Web search. Data from search logs, browsing logs, or ad-click logs obtained from different sources are used extensively by modern search engines that use the "wisdom of the crowds" contained in them to rank documents.

As we mentioned in the introduction, the best strategy for spammers is to game *any* signal they believe is used for search engine spam. By issuing automated queries and clicks, spammers try to fool search engines into believing that certain documents are more relevant than others. By issuing automated clicks, spammers try to inflate the number of clicks received by a given ad to defraud those who advertise on the Web. Both types of manipulation are studied in Section 6.2.

Fortunately, usage analysis can also be employed against spammers, as machine-generated data tend to be statistically different from human-generated data. Section 6.3 outlines how to use signals from usage analysis against spammers as a way of improving Web spam detection systems.

6.1 Usage Analysis for Ranking

Ranking Web pages is a difficult problem, and large-scale search engines are able to produce relevant results by considering a combination of many different factors [201]. The activities of users are an important source of information that has been started to be used extensively over the last few years.

Usage information consists of triples $\langle u, t, e \rangle$ where u is a possibly anonymized user identifier (e.g., a unique user-id, a unique browser cookie, or an IP address), t is a timestamp, and e is an event. Usage data sets for large populations or extended periods of time are huge, and can be very noisy and sparse (e.g., for a particular page or query we may have very little information). From the perspective of commercial search engines, usage information is found in three main forms:

(1) *Search logs* (query logs) which include the keywords searched by the users and the pages on which the users clicked. Sequences of actions are usually referred to as *query sessions*.
(2) *Browse logs* obtained from users that opt-in to a system for tracking their activities, e.g., by specifying this in their preferences when installing toolbar software. Sequences of actions are usually referred to as *browsing trails*.
(3) *Ad-click logs* in the case of search engines that also operate, or have agreements with, ad networks of pay-per-click ads.

Query logs can be used as a source of information for search engine ranking by boosting the pages that are more clicked by users for a given query, being careful to account for the *positional bias* [114, 57]. Eye-tracking studies and query click-through logs have shown that users strongly favor search results shown near the top of the search engine results page. The fact that certain areas of Web pages tend to be clicked more often independently of their relevance to the user task has been observed for several types of Web pages (not only search result pages), and it affects the interpretation of Web clicks in general [115].

Browse logs can also be used to improve search engine rankings; for instance BrowseRank [151] builds a continuous-time Markov chain from browsing trails, and then considers that the pages with the highest probability in the stationary distribution should be ranked higher, much as in PageRank but considering users' browsing activities as transitions and not hyperlinks.

Ad-click logs are used extensively to improve the click-through rate of ads shown to the users, as these clicks represent a large share of the income of search engines. In the case of ads that are displayed along search engine results for a query (known as *sponsored search*), the methods are based on estimating the expected revenue of a click, which is a function of the advertiser bids and the expected click-through-rate (CTR). This estimate has to be computed for a particular ad, in a particular slot (position), for a particular user, issuing a particular query. Given that often previous information about a specific combination of user, query, ad, and slot may be scarce or simply not available, the CTR is estimated by a prediction that aggregates information about, e.g., similar ads, close-by slots, similar queries, or similar users. This means that a malicious user issuing queries and then clicking (or not clicking) in an untruthful way may affect the CTR estimations for other users and thus manipulate indirectly the frequency with which certain ads are shown. For instance, a spammer operating on behalf of a company may issue many searches for the name of a product and then skip the ad of a particular competitor, clicking in other parts of the results page, in an attempt to reduce the chances of that ad being shown.

6.2 Spamming Usage Signals

In this section, we discuss three notable types of spamming behavior in which spammers attempt to corrupt the usage information that a search engine uses. Click fraud (Section 6.2.1) interferes with the analysis of clicks that would otherwise be the result of an unbiased human browsing the Web. Search spam (Section 6.2.2) interferes with the analysis of queries received by a search engine as they now include automated and intentional queries that do not reflect real human usage. Referrer spam (Section 6.2.3) interferes with the analysis of Web site visitation logs as they may include recorded sources of visits that in fact do not contain links to the Web site in question.

6.2.1 Click Fraud

Click fraud is the practice of skewing pay-per-click advertising data by generating illegitimate events [187], an activity that is prevalent and potentially very harmful for the sponsored search business model [109].

Many cases of click fraud are as follows: a content publisher has an agreement with an advertising network that will select ads to place in the pages of the publisher; then, the advertising network will pay the publisher for each click on the ads. Under these conditions, the publisher has an incentive to generate as many clicks as possible on the ads. These clicks can be generated by automated scripts, or by hiring people in low-income countries to browse the Web and click on ads [226]. In both cases, the spammer needs to conceal the fact that the clicks are automatic or all originating in the same geographical area, for instance by hiding behind a proxy or several layers of proxies (a technique known as "onion routing"). Another option is the use of large botnets, which, through their size, can conceal the behavior by spreading it out over thousands of infected machines [61].

Other cases of click fraud are more subtle: if two companies advertise similar products on the Web, there is an incentive for one company to click on the other's ads and thus deplete some of the advertising funds of its competitor, in some reported cases up to 30–40% of them [153].

In general, there are several possible responses to click fraud [109]. One part of the solution is monitoring (either by the search engine or by a third party) and filtering the click streams to discard fraudulent clicks. Another component of the solution may be to move toward pay-per-action, in which the payment is made not whenever a click occurs but when the user actually buys the product or service being advertised. However, there are barriers to the widespread adoption of pay-per-action, including the fact that it might require companies to share potentially sensitive data about business transactions with the advertising networks.

Estimates of expected revenues from clicks (based on click-through estimations) are in general susceptible to spamming activities. Immorlica et al. [107] describe estimation methods based on observing the previous impressions and clicks for an ad. They show that for a broad class of methods, a spammer that manipulates some of the impressions and clicks on them, can increase the average payment of an advertiser.

Metwally et al. [165] study how to detect fraudulent click coalitions. The underlying hypothesis is that an automatic clicking system is composed of several agents (automated programs, or humans hired for this task) that attempt to generate deceptive clicks for more than one "customer", for efficiency reasons. Thus, a detection method can be based on recognizing groups of pages or ads that share a number of visitors much larger than what would be expected by chance.

6.2.2 Search Spam

In the case of sponsored search (ads shown along search engine results), a malicious user may affect CTR estimations for ads both by searching and then clicking in some ads, as well as by searching and then *not clicking* in some ads. This is one of the reasons why not only automated clicks, but also automated searches, should be detected by search engines and labeled as such in a query log. Other reasons might include violations of the terms-of-service of a search engine by attempting to download and copy a subset of the retrieved result pages. On top of that, often these result pages are used as base content to be mixed with spam content, during the creation of content-based spam pages using the methods described in Section 3.2.

In an analysis of a 15 million entry search engine query log from 2006, Zhang and Moffat [249] recognized that the log included queries that originated from external sources such as Web APIs, toolbars, and other third-party programs. When they examined the hundred largest sessions, they found that about 90% of them were machine-driven, though they did not attempt to figure out if they were generated erroneously, intentionally, or maliciously.

Buehrer et al. [38] studied automated search traffic on a large-scale Web search engine. They found that given a sequence of queries labeled with the client IP address and associated with HTTP cookies, it was possible to classify the sessions into normal and automatic traffic with over 90% accuracy. Salient features in this classifier include the number of queries, the entropy of the keywords in the queries, the number of queries issued in a short (ten seconds or less) period of time, and the click-through rate. Duskin and Feitelson [67] also consider this issue, and suggest that the interaction between query submittal rate and minimum inter-query dwell time would be a useful feature.

6.2.3 Referrer Spam

A persistent, but relatively low-impact spamming technique is to employ a Web crawler that selectively retrieves Web pages but instead of including a referrer field in the HTTP request to show the source of the link being followed (as a browser would) or leaving the referrer field empty (as other Web crawlers do), the spamming Web crawler would include a target URL in the referrer field. The goal, presumably, is to attract links to the target URL from automatically generated (and published) pages containing Web logs excerpts (or perhaps human traffic from attentive webmasters who might be curious why their page is getting traffic from unexpected places[1]).

While referrer spam has not been extensively investigated in the scientific literature (an exception is Yusuke et al. [246]), it has been significant enough to merit its own entry in Wikipedia[2] and has a category in the Open Directory Project.[3] Fortunately, the crawlers that produce the referrer spam can often be detected as non-human [185, 215, 218], and thus their activities, in theory, can be filtered from usage logs.

6.3 Usage Analysis to Detect Spam

The previous section showed how spammers can interfere with usage signals. In this section, we briefly describe how usage signals can be used to improve Web spam detection systems. The usage data that can be exploited by search engines are browsing logs (e.g., captured

[1] For instance, http://www.completepills.com/ and http://store.liftmaster-remote.com/ were both listed as referrer more than a dozen times in the span of two months for references to pages on the airweb.cse.lehigh.edu host.

[2] http://en.wikipedia.org/wiki/Referer_spam.

[3] http://www.dmoz.org/Computers/Internet/Abuse/Referrer_Spam/.

using a toolbar), which are discussed in Section 6.3.1; and search logs, discussed in Section 6.3.2.

6.3.1 Traffic Analysis

Information from browsing logs can be used to help detect Web spam pages.

Bacarella et al. [11] analyze the *traffic graph*, a graphical representation of the browsing trails of users, in which nodes can be subsets from the Web, for instance pages or Web sites, and edges between two nodes u, v, contain information about users visiting u and v in sequence. An interesting measure in the traffic graph is the *relative traffic*, defined for a site v as the average fraction of the traffic of the in-neighbors of a v in the traffic graph, which is converted into traffic for the site v. For instance, if site v is the next site visited by 50% of the visitors of u and 70% of the visitors of w, then its relative traffic is 60%. Sites having very high relative traffic (over 90%) were empirically found to be mostly spammers, attracting visitors by deceptive means including pop-ups, pop-unders, redirects, etc.

Liu et al. [150] study both a query log and a browsing log to discover anomalies in some Web sites. They showed that Web pages that do not attract in-links or visits from in-links, but whose traffic relies almost completely on search engine-originated visits, are much more likely to be spam than non-spam. Other features that they show to be useful for detecting spam pages are the probability of clicking an out-link after arriving to a page (low probability for spam pages, meaning the click-through rates of spam pages are low), and the number of pages on a site viewed by users visiting the site (low number of pages for spam sites).

6.3.2 Query Search Logs

The query logs of search engines contain valuable information about popular queries, which are an attractive target for spammers.

Ntoulas et al. [182] and Castillo et al. [44] use this observation to create features for content-based Web spam detection. For instance, a list of the top popular queries submitted to a search engine can be assembled, and then a page can be considered suspicious if it contains an abnormally high fraction of those queries.

The ad-click logs of sponsored search can also be used to build lists of *monetizable* queries, those that attract high bids or many clicks from users. Monetizable queries are the queries that generate more revenue for the search engine. Chellapilla and Chickering [50] show that both popular queries and highly monetizable queries, particularly the latter, attract much more cloaking spam that other queries.

Finally, when automatically generating content for creating content spam (any of the types of mentioned in Section 3.2), spammers may generate pages that are shown in search engine results for many unrelated query terms. If some relationship among terms can be inferred from a query log (e.g., by looking at queries that generate clicks on the same documents, or a more general co-click relationship of this type), then a feature for Web spam detection can be built. This is studied in [41, 42] where Web pages that attract traffic for many unrelated queries are considered more likely to be spam than other pages.

6.4 Conclusions

In this section we recognize that activities on the Web are recorded, and that these records can be automatically aggregated to generate a useful signal, e.g., for ranking of both editorial results and advertising, as well as for charging advertisers. We have seen that adversaries may want to manipulate “the system”, not only to influence the ranking of their pages, but also to affect the advertising payments made by a competitor or the payments to a publisher site. In the following section we consider how spammers are not only involved in records of what we have done, but in the content that we generate online.

7

Fighting Spam in User-Generated Content

User-generated content in so-called social media, as opposed to professionally generated content from traditional media, has been a strong force behind the growth of the Web since the early 2000s, and a key aspect of its unique character as a communications medium. Over time, more and more users participate in content creation, rather than just consumption. Using widely available digital tools, people are becoming producers and consumers: "*prosumers*", a term coined by Alvin Toffler in 1980 [222]. Approximately one-third of users contribute content to the Web, as measured by studies performed in the U.S.A. [77] and China [144].

Popular user-generated content domains include blogs and Web forums, social bookmarking sites, photo and video sharing communities, as well as social networking platforms such as Facebook and MySpace. These opportunities are embraced by the majority in a constructive way, but abused by a minority that disrupts these platforms, or use them for deceptive or fraudulent purposes. Sections 7.2 and 7.3 describe how platforms for user-generated content are exploited by spammers to manipulate search rankings. Section 7.4 describes disruptive and deceptive activity in the social media platforms themselves.

7.1 User-Generated Content Platforms

There are basically three kinds of platforms for user-generated content that are attacked by spammers.

Free hosting sites. Most blogs are hosted on sites that offer free blog hosting and creation tools, usually in exchange for ads in the blog pages. Spammers use these hosting sites to create *splogs* ("spam blogs"), fake sites that present themselves as user-generated content while being machine-generated for the purpose of spamming.

Publicly-writable pages. Part of the user-generated content is gathered and aggregated through open systems in which anyone can write. These include opinion forums and comment forms, user review sites, and collaborative editing tools known as *wikis*.

Social media sites. User-generated content is also shared through sites in which users can upload content (images, videos, answers, etc.), and interact with the content through annotations, tags, votes, etc. Furthermore these sites usually allow people to interact with each other through social networking features.

As noted by Heymann et al. [101], apart from approaches based on detection of the spam items or the demotion of them in search results, a *preventive* approach is possible in the context of user-generated content. Social media sites can use CAPTCHAs or similar mechanisms to slow down automatic registration and automatic posting of content, or limit the number of users that can be affected by a single action. For instance, social media sites impose limits in contact lists or in the number of users that can receive any single message. Additionally, in social media sites users can, and usually do, help with policing bad behavior by reporting abuse and spam.

However, preventive approaches have disadvantages that have to be balanced with the obtained benefits. In the case of comments, methods which require user effort such as registration may reduce the number of spontaneous responses [167] that may be valuable for the authors seeking comments from their readers. Something similar happens in the case of Wikipedia, where spontaneous additions and corrections by casual readers are encouraged and must be balanced carefully against the prospects of vandalism.

In general, open and generative systems as described by Zittrain [257] are by their very nature susceptible to abuse. But as a general design principle the counter-measures to keep abuse under control should be postponed as much as possible, to give time for the system to mature and for its actors to develop social norms for dealing with abuse and spam. This is particularly important in large-scale systems in which borderline cases are frequent.

7.2 Splogs

Spam blogs are a particular and prevalent type of spam page. Some splogs are simply spam sites hosted in free hosting sites, and as usual their main aim is to deceive the algorithms of search engines to boost the ranking of some set of pages. Other splogs sites also try to deceive users either to click on ads, or by (falsely) presenting themselves as independent opinion sources about a product or service.

Another communication channel that is abused by spammers are blog pings. "Pings" are a lightweight mechanism by means of which blog platforms signal that new content is available to blog indexing or aggregator sites. The system that receives a ping adds the blog that sent the ping to a crawling queue. Splogs tend to generate an abnormally high number of pings; in that context, spam pings are sometimes referred to as *spings*.

Kolari et al. [127] present a characterization of the *splogosphere* based on a blog collection and on a collection of pings. The blog collection was provided by BlogPulse[1] and corresponds to 1.3 million blogs during a 21-day period in July 2005. The ping collection was provided by Weblogs.com[2] and corresponds to 15 million pings during a 20-day period in November 2006. With respect to classification of splogs, they show that it is possible to build an automatic classifier using content- and link-based features as for other spam pages [128].

[1] http://blogpulse.com/.

[2] http://weblogs.com/.

In contrast to authentic blogs, spam blogs generate more pings (≈75% of the pings in [127]). Also, the periodicity of pings is abnormal in the case of spam blogs. Legitimate users are more likely to post blog entries during the day than late at night. This is actually observed in the data if blogs that are likely to be concentrated in a single time zone are selected; in the case of [127], the authors examine blogs written in Italian and observe a clearly periodic behavior, with daily periodicity and a peak frequency more than 10 times larger than the valley frequency. In the case of blogs written in English, given that these blogs are distributed across several time zones, the periodicity of pings is not as sharp as in the case of Italian, with peak frequency (at U.S. working hours) about twice as large as valley frequency.

The classification of a splog, however, typically occurs after crawling and indexing of the blog has taken place. Given the high ratio of pings coming from splogs, a lighter-weight alternative is the detection of splogs through their pings. Kolari et al. [126] propose a meta-ping server that would receive notifications from ping servers but also utilize reader feedback, blacklists, whitelists in order to provide a filtered ping service which can then be fed to an indexing system.

Lin et al. [147, 148] look specifically at temporal patterns in blog postings in order to separate authentic blog from spam blogs. Three main observations are derived from their study. First, normal bloggers post at a regular but not precise time, while splogs show machine-like regularity (e.g., posting a new item exactly every three hours). Second, in splogs the distribution of content into topics varies either very rapidly or not at all, signaling either content plagiarized at random from the Web or a single source of content that is reproduced over and over. Third, splogs exhibit a smaller variability in their links over time than authentic blogs. These observations can be used to build features that capture regularity and self-similarity of temporal patterns, and these features can yield substantial improvements in the accuracy of a splog detection system.

Sato et al. [205] classify keywords appearing in Japanese splogs according to two dimensions. One is the informational content of the keywords, basically its inverse document frequency as described in Section 3.1.2. The other dimension is whether the keyword is long-lived or short-lived (e.g., a burst). The analysis of the keywords may lead to insights that help build better splog detection systems. Also, they found that a few professional spammers were responsible for the majority of the splogs in Japan.

7.3 Publicly-Writable Pages

Getting feedback and collecting ideas and opinions from the users is important for Web site authors and developers, but comment and opinion spam are annoyances for both Web site authors as well as for the users.

7.3.1 Forum Spam

Discussion boards and forums are among the oldest kinds of user-generated content with roots in bulletin board systems in the decades before the Web. Today, however, such wide-open sites are a visible target to spammers, and as a result forum spam is widespread, and is typically used to increase link-based authority [179]. All kinds of popular forums suffer from such spam.

Fortunately, many of the methods to detect or ameliorate comment spam, discussed next, can be applied to forum spam.

7.3.2 Comment Spam

Currently, two large commercial providers of comment spam protection (Akismet[3] and Mollom[4]) indicate that they receive more spam comments than legitimate comments in the Web site of their customers. As of April 2010, Mollom reports 90% of the messages they process are spam, while Akismet reports that 83% of the comments they process are spam. The services have different user bases and protect somewhat different services which may explain partially the difference; in any case it is clear that this is a prevalent problem.

State-of-the-art e-mail spam filters have been applied successfully to filter comment spam. Thomason [221] analyzes a collection of over

[3] http://akismet.com/.

[4] http://mollom.com/.

6,000 comments (of which 78% are spam) and shows that DSPAM[5] can reach a false positive rate of about 1% and a false negative rate of less than 0.5%.

Mishne et al. [167] proposed a content-based approach to filter comment spam. They observe that, while in the case of e-mail spam each message should be analyzed independently in principle, in the case of comments there is a context which is the page and site where the comment is posted. Their method starts by computing two language models: one for the original page in which the comment is posted, and one for each comment. Next, a measure of distance between the language models of the page and each comment is computed. This distance is based on the difference between their language models, measured using the Kullback–Leibler divergence. Finally, a threshold in this distance is used to discriminate between spam comments (larger distance) and legitimate comments (smaller distance).

Given that both the page and the comment may be very short, the model of each one can be enriched by means of linked pages. Thus, the language model for a page can be computed by taking into consideration the text of the pages being linked, and the language model for a comment can be computed including the text of the pages linked from the comment. This is useful given that most spam comments currently include out-links as they are aimed at influencing link-based ranking. A link to an unrelated page to the one being commented on increases the distance between the language models and makes the comment more likely to be spam.

The "nofollow" attribute Because of the negative impact that comment spam was having on the blogosphere, in 2005 the major blogging services and software vendors, along with Google, Yahoo!, and MSN Search proposed a new link attribute called `nofollow` [156]. Any link on a Web page having this form:

```
<a href="..." rel="nofollow">...</a>
```

indicates that the destination of that hyperlink should not be afforded any additional weight or ranking by search engines doing ranking of

[5] http://dspam.nuclearelephant.com/.

pages based on link analysis. This means that the link may be *followed* by a Web crawler, but discarded when computing, for instance, PageRank or HITS. The usage scenarios for this type of link are publicly-writable Web pages like Wikis or Blogs where users can post links; several applications for maintaining these types of sites by default add the `nofollow` tag to links posted by untrusted users. The intention is to discourage spammers from posting links in such pages. By mid-2006, about 1% of all Web links had the nofollow attribute applied [63]. By 2009 that fraction had grown to 2.7% [76], but almost three-quarters of those were links to other pages on the same site, suggesting that most use of nofollow was to control explicitly how authority flows, rather than simply to make user-generated content not affect authority calculations. To the best of our knowledge, no study has been released publicly measuring the effectiveness of the `nofollow` tag, either as a deterrent (comment spam is still ubiquitous!) or as a useful signal for ranking.

7.3.3 Opinion and Review Spam

There are broadly two types of spam reviews [112]. First, there are false reviews that deliberately try to mislead readers or automatic systems by giving undeserving positive or negative opinions about a product. Second, there are non-reviews that contain spam and are basically cases of spam in publicly-writable pages. A more refined classification of review spam [111] includes the following classes:

- False reviews: containing misleading judgments portrayed as truthful. Detecting these in general requires a considerable amount of domain knowledge.
  - Positive false reviews: undeservedly positive opinions.
  - Negative false reviews: undeservedly negative opinions.
- Non-reviews: do not contain misleading judgments.
  - Advertisement: these are similar to spam in publicly-writable pages.

    - Other: questions, meta-comments, vandalism, etc. These can be on purpose (as in the case of vandalism) or by mistake (as in the case of, e.g., a misplaced question).
- Brand reviews: contain only statements about a brand, but not about a product. Again, this can be done by users simply by mistake.

The detection mechanism presented by Jindal and Liu [111] uses textual features from the review as well as context information. The context information includes the feedback from other users which can vote a review as helpful or unhelpful, user information such as statistics about the ratings she has provided in the past, and also information about the product being reviewed.

In a data set of 470 manually labeled reviews from Amazon product reviews, they report a very high accuracy (AUC ≈ 0.98) in separating non-reviews and brand reviews from legitimate reviews. Finding false reviews is harder, even for humans. To detect a subset of the possibly false reviews, they use near-duplicate detection to gather a set of suspicious cases. They include, for instance, the same user-id posting repeatedly the same review to different products, or different user-ids posting exactly the same review text; these are often false reviews. In separating this class of false reviews from legitimate reviews, they achieve AUC ≈ 0.78.

The study of false reviews is further deepened in [113] where also the ratings are taken into account. User ratings are numeric votes for quality of the product that go from 1 to 5, usually represented as a number of "stars" in the user interface. An interesting finding is that ratings that are substantially lower (more negative) than the average rating of a product are more likely to be spam reviews, than ratings that are more positive than the average.

Methods of trust propagation such as the ones discussed in Section 5 can be employed to score opinions according to how trusted are the users producing them. One method of this type is the *TrustWalker* method by Jamali and Ester [108] in which reviews in a collaborative recommendation platform are scored by trust.

7.4 Social Networks and Social Media Sites

The growth of online social networks like Facebook, Twitter, and MySpace has dramatically increased the ability for users to find and communicate directly with more people. This power can also be used for illegitimate aims, such as phishing attacks, malware dissemination, and as expected, spam.

7.4.1 Social Network Trust

A first step toward fighting spam in social networks is to develop methods for computing the reputation of members. This is particularly needed because participants of online social networks often share personal details of their lives, making it important that the people with whom they connect be trustworthy.

The SocialTrust model by Caverlee et al. [47, 48] is a global trust function computed in a centralized manner (as defined in Section 5.1). First, the *core trust score* for a user u is computed; it is a function of the core trust of the "friends" of u, of the inferred quality of the connections of u with them, and of the explicit feedback ratings received by u. Second, the SocialTrust score for a user u at time t is computed; it is a linear combination of the core trust of u at time t, of the derivative of u's core trust with respect to time, and of the average of the SocialTrust score of u in the past. The purpose is to mitigate the effect of users who accumulate a good reputation over time, and then take advantage of that reputation to behave maliciously.

There are also models based on local trust computations, in which the trust of a user is computed from the perspective of another user, and not on the entire network. They include among others the reputation system implemented in the *Advogato* community, based on maximum flows [142], a model based on weighted paths due to Mui et al. [172] and *Appleseed*, a system based on spreading activation [255].

Social network trust can also be used across different services. The FaceTrust method by Sirivianos et al. [213] provides a general mechanism for verifying the credentials of a user. The idea is that if a user u needs to prove to a third party that the user has a certain property (e.g., that u is above 18 years old), the user may direct this third party to a social network, which in turns can ask u's connections to validate this assertion. This can be done in a privacy-preserving way, in which data about u that is not necessary for the transaction with the third party to take place, and does not need to be disclosed outside the social network. Naturally, this can be extended to create an environment in which users can assure new online services that they are "good netizens" by providing credentials from their previous activities in other social networks.

7.4.2 Social Media Spam

Social media sites such as Delicious, YouTube, and Flickr, which allow posting of items or shared bookmarks, are prone to various types of spam. Users can post or bookmark spammy content, or send it to other users when a direct user-to-user communication mechanism is available. The mechanisms for user voting and reporting of abuse and spam can help reduce the impact of such spam, unless users collude by agreeing to promote or demote certain items through tags or votes. In any case, automatic detection of abusive of spammy items may help discourage spammers and improve the user experience for the whole community.

Tagging spam Koutrika et al. [132, 133] introduce a model for malicious (spammy) and normal behavior of users in tagging systems. According to their model, each object (photo, page, etc.) d posted to the social media site can be described by some tags $S(d)$, while some malicious users pick tags in $S(d)^C$ as descriptors for d. Given that determining $S(d)$ requires domain knowledge, a spam-resistant tagging system considers the matches of the tags of one user with the rest of the users in the system as a measure of reliability for that user. Specifically, every time two users assign the same tag to the same object, the reliability of both users increases; and this reliability is used when computing the strength of the assignment of a task to an object. Their paper includes experiments in a simulated tagging environment in which malicious users operate either individually or as a group.

Abnormal tagging patterns can be detected, e.g., by graphical methods. Neubauer et al. [174] consider a graph in which users are connected if the number of items they have both tagged exceeds a certain threshold. Coalitions of spammers may appear as large connected components (that are separated from the largest connected component) in this graph. In other words, groups of users tagging documents in which the majority of legitimate users are not interested, are more likely to be spammers.

Krause et al. [134] study social bookmarking systems, in particular the case of Bibsonomy.[6] Their paper focuses on the characterization of *users* that contribute spam content to the system. These users are represented by vectors containing features extracted from their user profile (e.g., number of digits in their names or e-mail addresses), from their network location (e.g., number of users sharing the same IP address or domain), and from the tags they use (e.g., checking the intersection of their tags with a blacklist of known tags used by spammers). Putting these features together, they are able to build a classifier which achieves a high accuracy (AUC ≈ 0.93) on a test set of about 2,700 unseen cases.

The problem of finding these *anti-social* users in social media sites motivated the ECML/PKDD 2008 discovery challenge.[7] In that competition, a data set from Bibsonomy was provided including human-annotations for 22,000 spam users and 2,000 non-spam users. The winning entry by Gkanogiannis and Kalamboukis [82] represented each user by a document that was the concatenation of all their postings in the system; then they used their text classification algorithm [81] obtaining an AUC of 0.98 in a test set of unseen cases. The runner-up entry by Gramme and Chevalier [86] used their RANK modeling tool over a richer feature set computed from the text, tags, resources posted by users; they obtained an AUC of 0.97 over the same test set.

Markines et al. [155] also consider the problem of spam within social bookmarking systems and consider characteristics of all three aspects of a social bookmark: the tag, the user, and the target. They offer six features for this task: probability of a tag being used by a spammer, dissimilarity of tags used in a post, the likelihood of a target page being generated automatically, the number of ads on the target page, the

[6] http://www.bibsonomy.org/.

[7] http://www.kde.cs.uni-kassel.de/ws/rsdc08/.

likelihood of the content of the target page being plagiarized, and the fraction of a user's posts that still refer to valid resources. Each of these features is demonstrated to be useful, and the combination along with a classifier like AdaBoost provides quite competitive performance.

An application exists that uses simple heuristic to filter a stream of posts from the microblogging platform Twitter. The *Clean Tweets* extension for Firefox hides posts from accounts that are less than one day old or that contain too many tags.[8]

Voting spam Bian et al. [26] study the effect of voting spam in the Yahoo! Answers[9] question-answering portal. In this Web site, users post questions and answers and vote on the answers of others either with positive ("thumbs up") or negative ("thumbs down") votes. The authors introduce two synthetic attack models, one in which a set of users picks a set of answers and vote positively on them, and another in which a set of users pick a set of answers and vote positively on them and negatively on the other answers to the same questions of the answers being promoted. In both cases, the rankings are affected by the introduction of spam votes; a ranking system can be "hardened" against spam by introducing synthetic votes following a certain attack model in the training set. In practice, this "teaches" the ranking system to reduce the weights of the features that are affected by spam (e.g., number of positive or negative votes) and thus reduces the impact of spam at run time.

Tran et al. [223] describe a voting method that analyzes the social network of users, and gives less weight to votes from users that are not well connected to other users. The authors show results from preliminary experiments in Digg[10] indicating that this method is effective in demoting highly ranked spammy content.

Video spam Spam also plagues social video sites such as YouTube, in which users can post responses to existing videos. Benevenuto et al. [22] target video spammers in their work by examining attributes of objects,

[8] https://addons.mozilla.org/en-US/firefox/addon/12384.

[9] http://answers.yahoo.com/.

[10] http://www.digg.com/

users, and the social network connecting them. They showed that spam users and objects had distributions that differed from legitimate users and videos for characteristics such as number of friends, number of responses received, number of favorites, etc.

7.5 Conclusions

While since the beginning of the Web, user-generated content has played a central role, over time the Web has become even more interactive. Content is generated not only by dedicated authors, but even casual Web surfers are now participants in discussion boards, social networks, photo and bookmark sharing sites, and more. We are communicating and collaborating across many social environments, each of which has the potential to be manipulated by malicious participants of varying sophistication. A number of studies have examined detection and amelioration approaches under simulated adversaries (e.g., [47, 132, 133]) while a few have used labeled data from real services (e.g., [22, 82, 155]). We also highlighted difficulties of the vociferous subset of the Web called the blogosphere, as the size, popularity and ease of creation of blog content has made such participatory content both valuable and easily affected by adversaries wishing to manipulate the system.

Successful spam detection approaches for user-generated content have significant parallels to those methods described in earlier sections. By modeling the entities involved, it is often possible to find features that have different characteristics for spam versus non-spam content or creators. The explicit modeling of author reputation can be helpful in estimating entity reputation, and finally, training data to train and exploit automated classifiers is as always crucial.

8

Discussion

Given our coverage of the various spamming mechanisms and the methods to detect them, in this final section we discuss our views on the status of research in adversarial Web search. The struggle between the search engines and the spammers continues, with searchers and content providers often suffering from collateral damage. The continued difficulties, however, lend themselves as ongoing research problems, and a number of resources are available to those interested in pursuing such topics.

8.1 The (Ongoing) Struggle Between Search Engines and Spammers

8.1.1 Search Engine Perspective

Over time, search engines have developed sophisticated algorithms to fight Web spam. Indeed, one of the original motivations behind PageRank [183] was trying to counter content spam, by adding features to the ranking function of the search engine that were not under the control of the author of the Web page itself. However, much like any other feature, PageRank is manipulated by Web page authors trying to deceive search engines. This is the reason why most search engines, while not publishing the exact details on their ranking functions, are believed to use hundreds or even thousands of different signals for ranking pages. Also, search engines use a variety of “penalties” once they detect a spammer. These penalties can range from demotion of the page to its removal from the search index. Again, the specific conditions under which these penalties are applied are not disclosed.

A part from efforts from individual search engines, the search engine industry has been able to achieve consensus in key areas in the past — such as sitemaps,[1] and robot exclusion [130] — indicating that it is possible to agree on industry-wide efforts for dealing with Web spam. One example is the `nofollow` tag described in Section 7. Another example related to search engine optimization is the recent introduction of the canonical URL tag [137] to eliminate self-created duplicate content in search engine indexes.

In the end, search engine providers want to provide the most useful and valuable service possible, which typically means they want to provide an objective ranking of relevant results. Generally speaking, this means they would like Web site owners to optimize for users, not for search engines.

8.1.2 Spammer’s Perspective

Search engines may be able to change the economics of spamming by making it more expensive and less profitable for spammers to spam. However, even if the chances of suceeding become very small, there will always be some people and organizations that will try to increase their audience in the short term by spamming.

As in e-mail spam, there is a general trend toward increasing sophistication of Web spam, but there is also a mixture of sophisticated and naïve Web spam, possibly due to some new spammers trying old tactics that are well known by search engines and that can do little harm to their ranking methods.

Also as in e-mail spam, sometimes the spam companies may sue the companies or groups that blacklist them. A high-profile case was

[1] http://www.sitemaps.org/.

the lawsuit by Traffic-Power.com [188], which was as unsuccessful as previous attempts by e-mail spammers to obtain legal protection for their activity. Google has maintained the position in US courts that their rankings of Web sites are opinions about those sites and as such are protected by the First Amendment.[2]

8.1.3 SEO Perspective

Unsuspecting Web site owners are often hurt by the adversarial situation in Web IR. If owners create high-quality original content, that content may be copied and re-used by spammers. When a spammer's Web page rises and legitimate sites subsequently fall in rankings, content providers may be sorely tempted to go beyond SEO and utilize spamming techniques to be competitive.

However, SEO experts such as Moran and Hunt [169] advise Web site administrators to refrain from doing spam: "Many unethical search marketing techniques (known as spam) try to fool search engines to find your pages when they really should not match, and every search engine takes measures to avoid being fooled."

The main reason for SEOs to avoid spamming is the risk of being detected, either immediately or in the future:

> "The search engines are aware of the many sneaky ways that site owners try to achieve undeserved ranks. . . If they discover that you're trying to do this, your site may be penalized: Your rank may be downgraded, or your page — or even your whole site — could be banned. Even if your site is never caught and punished, it's very likely, we dare say inevitable, that your tricky technique will eventually stop working." [88].

> "[I]f search engines do not flag [some spam] pages today, some day they will. They get smarter every year. Beyond search engine smarts, over-optimized pages also leave you vulnerable to being reported by your competitors to search engines for spamming — causing a

[2] http://www.internetlibrary.com/cases/lib_case337.cfm.

> human editor to check the page and possibly ban your site.” [169].

While a high-profile site that gets cleaned up might be able to be removed from a blacklist fairly quickly (e.g., BMW [105]), most sites may find recovering from blacklisting actions to be difficult and time-consuming.

8.2 Outlook

Spam is likely to continue to be an issue for search engines as both spammers and search engines develop more powerful techniques. The ability to communicate at a low cost without any approval from a third party has been key to the success of the Web and is unlikely to change in the future.

According to Metaxas and Destefano [164] Web spam is mostly a social problem, not a technical one; and as a social problem, the solution is on the hands of people who are the objective of the spammer’s attempt. Accordingly, Gyöngyi and Garcia-Molina state that:

> “[I]n the long run, the best solution to the ongoing battle is to make spamming ineffective — not only in its attempt to subvert search engine algorithms but also — and more important — in its attempt to coerce users. If people are more conscious about spamming and avoid being lured into its traps, the economic or social incentive for spamming will decrease.” [92].

From a technical perspective, the arms race between search engine spammers and search engines does not need to continue indefinitely:

> “Victory does not require perfection, just a rate of detection that alters the economic balance for a would-be spammer.” [182].

Note that in both cases, Web spammers need to understand that the situation has changed, or they will still generate spam, even when it is ineffective (as in the case of comment spam when nofollow is applied).

Thus, it is the spammer's perception of the relative benefits and drawbacks of spam that needs to change in order for Web spam to truly end.

Web spam may, in the long term, become a smaller piece of the efforts related to Adversarial IR on the Web, as other problems may become more important over time. Bloggers and other enthusiasts are generating vast amounts of content and competing with each other to get the attention of users, and in this competition some of them resort to deceptive practices. Social media sites allowing a more direct one-to-one communication are popular and susceptible to be spammed.

8.3 Research Resources

8.3.1 Data Sets

Over the years, a few annotated corpora have been made available to the research community for research on Web spam.

The *Webb Spam Corpus*[3] [233] is a collection of 350,000 Web spam pages. It was built semi-automatically starting from a large corpus of e-mail spam messages, and scanning for URLs mentioned in those messages. The collection includes the contents of all the Web spam pages, and it can be freely downloaded.

The *Splog Blog Dataset*[4] [125] is a set of 3,000 blog home pages, out of which 700 have been labeled manually as splogs and 700 as legitimate blogs. The collection includes the contents of all the blog home pages and the labels, and it can be freely downloaded.

The *WEBSPAM-UK2006* and *WEBSPAM-UK2007* data sets[5] [43] are a set of hosts from a crawl of the `.uk` domain labeled by an international team of volunteer researchers. In the newest collection, there are 114,529 hosts out of which 6,479 have been labeled. The collection includes the contents of up to 400 pages per host (a larger version is available), the links between the hosts, and the labels. The collection is freely available for download, except for the page contents that are

[3] http://www.cc.gatech.edu/projects/doi/WebbSpamCorpus.html.

[4] http://ebiquity.umbc.edu/resource/html/id/212/Splog-Blog-Dataset.

[5] http://barcelona.research.yahoo.net/webspam/datasets/.

available for download upon signing a research-only agreement. It was used in the Web Spam Challenge 2007 and 2008.[6]

The *2008 ECML PKDD Discovery Challenge Dataset*[7] is a collection of bookmarks in a social bookmarking service (*Bibsonomy*). It contains a few hundred thousand bookmarks (URLs or BibTeX files). The operators of the bookmarking service have identified about 25,000 accounts as spammers by manually inspecting bookmarks in the site. The collection includes users, bookmarks, and labels, and is freely available for download.

The Clue Web09 Dataset[8] contains more than a billion Web pages and has been used in TREC since 2009. Gordon Cormack and collaborators at the University of Waterloo has built a classifier from honeypot queries, labels from the WEBSPAM-UK2006 and WEBSPAM-UK2007 collections, and a small set of hand-labeled data [56] and evaluated it in terms of the effect on retrieval. The labels produced have been made publicly available.[9]

The *2010 ECML PKDD Discovery Challenge*[10] includes tasks related to Web host quality prediction for Internet Archives.

8.3.2 Query Logs

With respect to usage data, in general query logs are used by search engine companies but are difficult to obtain and are not easily available for the academic community, mostly because of privacy issues as they are hard to sufficiently anonymize them without degrading them substantially, although research continues in that direction, e.g., [136, 129].

AOL released in 2006 a query log in what became a highly publicized incident. User privacy was compromised because of insufficient anonymization of the query log, causing a major uproar that resulted in the employees involved in the sharing of the query log being fired. This incident and its implications are described in [211]. The query

[6] http://webspam.lip6.fr/.

[7] http://www.kde.cs.uni-kassel.de/ws/rsdc08/dataset.html.

[8] http://boston.lti.cs.cmu.edu/Data/clueweb09/.

[9] http://durum0.uwaterloo.ca/clueweb09spam/.

[10] http://www.ecmlpkdd2010.org/.

log was officially withdrawn by its authors, but it continues to be used for research on query log mining for several reasons. First, while it is no longer available for download from the AOL site, copies of it are available elsewhere[11]; second, it is assumed that research on this query log may help avoid future privacy issues for users; and third, because being a widely available data source, it allows researchers to reproduce and compare their approaches.

One main issue with the AOL query log was that it was freely available without any formal agreement from people receiving the query log. During the WSCD 2009 workshop, Microsoft made available a sample of their query log[12] under a research-only agreement in which researchers had to assure, among other things, that they would not attempt to use the query log to uncover private information about any user, nor reveal any of the contents of the query log to third parties.

Finally, we note that an academic effort led by faculty from Carnegie Mellon University and the University of Massachusetts, Amherst has recently begun to collect query and usage logs from volunteers.[13]

8.3.3 Venues

Adversarial Information Retrieval on the Web[14] is a series of yearly workshops started in 2005. This workshop continued as a joint AIR-Web/WICOW workshop on Web Quality in 2011.[15] The topics of these workshops are closely related to those in this monograph. Other research workshops related to the topics covered in this monograph are the Conference on E-mail and Anti-Spam CEAS[16] and broader venues such as the World Wide Web Conference.[17]

In addition to the aforementioned resources, there is a low-volume, announcements only mailing list with respect to Adversarial IR on the Web.[18]

[11] See http://gregsadetsky.com/aol-data/ for a list of mirrors.
[12] http://research.microsoft.com/en-us/um/people/nickcr/WSCD09/.
[13] http://www.lemurproject.org/querylogtoolbar/.
[14] http://airweb.cse.lehigh.edu/.
[15] http://www.dl.kuis.kyoto-u.ac.jp/webquality2011/.
[16] http://www.ceas.cc/
[17] http://www.iw3c2.org/conferences/.
[18] http://groups.yahoo.com/group/webspam-announces/.

8.4 Conclusions

By now it should be apparent that there is no panacea for either side in adversarial information retrieval, and that new opportunities for spam continue to appear as the Web continues to evolve into a more participatory form. While decidedly less than ideal for searchers and content owners caught in the crossfire, this scenario bodes well for those employed on both sides of the battle.

Acknowledgments

Carlos Castillo was partially supported by the Spanish Centre for the Development of Industrial Technology under the CENIT program, project CEN-20101037, “Social Media” (http://www.cenitsocialmedia.es/).

Brian Davison was partially supported by the U.S. National Science Foundation under Grant Number IIS-0545875.

References

[1] B. A, K. Csalogány, and T. Sarlós, "Link-based similarity search to fight Web spam," in *Proceedings of the Second International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2006.

[2] J. Abernethy and O. Chapelle, "Semi-supervised classification with hyperlinks," in *Proceedings of the ECML/PKDD Graph Labeling Workshop*, September 2007.

[3] J. Abernethy, O. Chapelle, and C. Castillo, "Webspam identification through content and hyperlinks," in *Proceedings of the Fourth International Workshop on Adversarial Information Retrieval on the Web (AIRWEB)*, pp. 41–44, ICPS: ACM Press, April 2008.

[4] J. Abernethy, O. Chapelle, and C. Castillo, "Graph regularization methods for web spam detection," *Machine Learning Journal*, vol. 81, no. 2, pp. 207–225, 2010.

[5] S. Adali, T. Liu, and M. Magdon-Ismail, "Optimal link bombs are uncoordinated," in *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, May 2005.

[6] B. T. Adler and L. de Alfaro, "A content-driven reputation system for the Wikipedia," in *Proceedings of the 16th International Conference on World Wide Web (WWW)*, pp. 261–270, New York, NY, USA: ACM, 2007.

[7] E. Amitay, S. Yogev, and E. Yom-Tov, "Serial sharers: Detecting split identities of Web authors," in *Workshop on Plagiarism Analysis, Authorship Identification, And Near-Duplicate Detection*, July 2007.

[8] R. Andersen, C. Borgs, J. Chayes, J. Hopcraft, V. Mirrokni, and S.-H. Teng, "Local computation of PageRank contributions," in *Algorithms and Models*

for the Web-Graph, vol. 4863 *of Lecture Notes in Computer Science*, pp. 150–165, Springer, 2007.

[9] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke, and S. Raghavan, "Searching the Web," *ACM Transactions on the Internet Technology (TOIT) 1*, vol. 1, pp. 2–43, August 2001.

[10] J. Attenberg and T. Suel, "Cleaning search results using term distance features," in *Proceedings of the Fourth International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 21–24, New York, NY, USA: ACM, 2008.

[11] V. Bacarella, F. Giannotti, M. Nanni, and D. Pedreschi, "Discovery of ads Web hosts through traffic data analysis," in *Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*, pp. 76–81, New York, NY, USA: ACM, 2004.

[12] R. Baeza-Yates, C. Castillo, and V. López, "PageRank increase under different collusion topologies," in *First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 17–24, May 2005.

[13] R. Baeza-Yates and B. Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley, May 1999.

[14] J. Bar-Ilan, "Web links and search engine ranking: The case of Google and the query "jew"," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 12, pp. 1581–1589, 2006.

[15] J. Bar-Ilan, "Google bombing from a time perspective," *Journal of Computer-Mediated Communication*, vol. 12, no. 3, 2007.

[16] Z. Bar-Yossef, I. Keidar, and U. Schonfeld, "Do not crawl in the DUST: Different URLs with similar text," *ACM Transactions on the Web*, vol. 3, no. 1, pp. 1–31, 2009.

[17] J. Battelle, *The Search: How Google and Its Rivals Rewrote the Rules of Business and Transformed Our Culture.* New York: Portfolio, 2005.

[18] L. Becchetti, C. Castillo, D. Donato, R. Baeza-Yates, and S. Leonardi, "Link analysis for Web spam detection," *ACM Transactions on the Web*, vol. 2, no. 1, pp. 1–42, February 2008.

[19] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates, "Using rank propagation and probabilistic counting for link-based spam detection," in *Proceedings of the Workshop on Web Mining and Web Usage Analysis (WebKDD)*, ACM Press, August 2006.

[20] A. A. Benczúr, I. Bíró, K. Csalogány, and M. Uher, "Detecting nepotistic links by language model disagreement," in *Proceedings of the 15th International Conference on World Wide Web (WWW)*, pp. 939–940, ACM Press, 2006.

[21] A. A. Benczúr, K. Csalogány, T. Sarlós, and M. Uher, "SpamRank: Fully automatic link spam detection," in *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, May 2005.

[22] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, C. Zhang, and K. Ross, "Identifying video spammers in online social networks," in *Proceedings of the Fourth International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 45–52, New York, NY, USA: ACM, 2008.

[23] P. Berkhin, "A survey on PageRank computing," *Internet Mathematics*, vol. 2, no. 2, pp. 73–120, 2005.

[24] K. Berlt, E. S. de Moura, C. M. André, N. Ziviani, and T. Couto, "A hypergraph model for computing page reputation on Web collections," in *Proceedings of the Simpósio Brasileiro de Banco de Dados (SBBD)*, pp. 35–49, October 2007.

[25] K. Bharat and M. R. Henzinger, "Improved algorithms for topic distillation in hyperlinked environments," in *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 104–111, August 1998.

[26] J. Bian, Y. Liu, E. Agichtein, and H. Zha, "A few bad votes too many?: Towards robust ranking in social media," in *Proceedings of the Fourth International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 53–60, New York, NY, USA: ACM, 2008.

[27] A. Bifet, C. Castillo, P.-A. Chirita, and I. Weber, "An analysis of factors used in search engine ranking," in *Proceedings of the First International Workshop on Adversarial Information Retrieval (AIRWeb)*, May 2005.

[28] I. Bíró, D. Siklósi, J. Szabó, and A. Benczúr, "Linked latent dirichlet allocation in Web spam filtering," in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 37–40, ACM Press, 2009.

[29] I. Bíró, J. Szabó, and A. A. Benczúr, "Latent dirichlet allocation in Web spam filtering," in *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 29–32, New York, NY, USA: ACM, 2008.

[30] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[31] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas, "Finding authorities and hubs from link structures on the World Wide Web," in *Proceedings of the 10th International Conference on World Wide Web (WWW)*, pp. 415–429, 2001.

[32] O. P. Boykin and V. Roychowdhury, "Personal Email networks: an effective anti-spam tool," Condensed Matter cond-mat/0402143, 2004.

[33] S. Brin, R. Motwani, L. Page, and T. Winograd, "What can you do with a Web in your pocket?," *Data Engineering Bulletin*, vol. 21, no. 2, pp. 37–47, 1998.

[34] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," in *Proceedings of the 7th International Conference on the World Wide Web*, pp. 107–117, April 1998.

[35] A. Brod and R. Shivakumar, "Advantageous semi-collusion," *The Journal of Industrial Economics*, vol. 47, no. 2, pp. 221–230, 1999.

[36] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, "Syntactic clustering of the Web," *Computer Networks and ISDN Systems*, vol. 29, no. 8–13, pp. 1157–1166, September 1997.

[37] T. A. Brooks, "Web search: How the Web has changed information retrieval," *Information Research*, vol. 8, no. 3, April 2003.

[38] G. Buehrer, J. W. Stokes, and K. Chellapilla, "A large-scale study of automated Web search traffic," in *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 1–8, New York, NY, USA: ACM, 2008.

[39] S. Büttcher, C. L. A. Clarke, and B. Lushman, "Term proximity scoring for ad-hoc retrieval on very large text collections," in *Proceedings of the 29th ACM Annual SIGIR Conference on Research and Development in Information Retrieval*, pp. 621–622, New York, NY, USA: ACM Press, 2006.

[40] C. Castillo, "Effective Web Crawling," PhD thesis, University of Chile, 2004.

[41] C. Castillo, C. Corsi, D. Donato, P. Ferragina, and A. Gionis, "Query log mining for detecting polysemy and spam," in *Proceedings of the KDD Workshop on Web Mining and Web Usage Analysis (WEBKDD)*, Springer: LNCS, August 2008.

[42] C. Castillo, C. Corsi, D. Donato, P. Ferragina, and A. Gionis, "Query-log mining for detecting spam," in *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, ICPS: ACM Press, April 2008.

[43] C. Castillo, D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini, and S. Vigna, "A reference collection for Web spam," *SIGIR Forum*, vol. 40, no. 2, pp. 11–24, December 2006.

[44] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri, "Know your neighbors: Web spam detection using the web topology," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, July 2007.

[45] J. Caverlee, "Tamper-Resilient Methods for Web-Based Open Systems," PhD thesis, College of Computing, Georgia Institute of Technology, August 2007.

[46] J. Caverlee and L. Liu, "Countering Web spam with credibility-based link analysis," in *Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 157–166, New York, NY, USA: ACM, 2007.

[47] J. Caverlee, L. Liu, and S. Webb, "Socialtrust: Tamper-resilient trust establishment in online communities," in *Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pp. 104–114, 2008.

[48] J. Caverlee, L. Liu, and S. Webb, "Towards robust trust establishment in Web-based social networks with SocialTrust," in *Proceedings of the 17th International World Wide Web Conference (WWW)*, pp. 1163–1164, ACM, 2008.

[49] J. Caverlee, S. Webb, and L. Liu, "Spam-resilient Web rankings via influence throttling," in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 1–10, 2007.

[50] 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)*, pp. 17–24, August 2006.

[51] K. Chellapilla and A. Maykov, "A taxonomy of JavaScript redirection spam," in *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 81–88, New York, NY, USA: ACM Press, 2007.

[52] A. Cheng and E. Friedman, "Manipulability of PageRank under sybil strategies," in *Proceedings of the First Workshop on the Economics of Networked Systems (NetEcon06)*, 2006.

[53] Y.-J. Chung, M. Toyoda, and M. Kitsuregawa, "A study of link farm distribution and evolution using a time series of Web snapshots," in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 9–16, ACM Press, 2009.

[54] A. Clausen, "The cost of attack of PageRank," in *Proceedings of the International Conference on Agents, Web Technologies and Internet Commerce (IAWTIC)*, July 2004.

[55] G. V. Cormack, "Email spam filtering: A systematic review," *Foundations and Trends in Information Retrieval 1*, pp. 335–455, 2008.

[56] G. V. Cormack, M. Smucker, and C. L. Clarke, "Efficient and effective spam filtering and re-ranking for large web datasets," Unpublished draft, available from http://durum0.uwaterloo.ca/clueweb09spam/spamhunt.pdf, retrieved 10, April 2010.

[57] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey, "An experimental comparison of click position-bias models," in *Proceedings of the First International Conference on Web Search and Data Mining (WSDM)*, pp. 87–94, New York, NY, USA: ACM, 2008.

[58] B. Croft, D. Metzler, and T. Strohman, *Search Engines: Information Retrieval in Practice.* Addison Wesley, 2009.

[59] A. L. C. da Costa-Carvalho, P.-A. Chirita, E. S. de Moura, P. Calado, and W. Nejdl, "Site level noise removal for search engines," in *Proceedings of the 15th International Conference on World Wide Web (WWW)*, pp. 73–82, New York, NY, USA: ACM Press, 2006.

[60] N. Dai, B. Davison, and X. Qi, "Looking into the past to better classify Web spam," in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 1–8, ACM Press, 2009.

[61] N. Daswani and M. Stoppelman, "The anatomy of Clickbot.A," in *Proceedings of the USENIX HOTBOTS Workshop*, April 2007.

[62] B. D. Davison, "Recognizing nepotistic links on the Web," in *Artificial Intelligence for Web Search*, pp. 23–28, AAAI Press, July 2000.

[63] B. D. Davison, M. Najork, and T. Converse, "Adversarial information retrieval on the Web (AIRWeb 2006)," *SIGIR Forum*, vol. 40, no. 2, pp. 27–30, 2006.

[64] J. Douceur, "The sybil attack," in *Proceedings of the First International Peer To Peer Systems Workshop (IPTPS)*, pp. 251–260, Springer: Vol. 2429 of *Lecture Notes in Computer Science*, January 2002.

[65] I. Drost and T. Scheffer, "Thwarting the nigritude ultramarine: Learning to identify link spam," in *Proceedings of the 16th European Conference on Machine Learning (ECML)*, pp. 233–243, Vol. 3720 of *Lecture Notes in Artificial Intelligence*, 2005.

[66] Y. Du, Y. Shi, and X. Zhao, "Using spam farm to boost PageRank," in *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 29–36, New York, NY, USA: ACM, 2007.

[67] O. Duskin and D. G. Feitelson, “Distinguishing humans from robots in Web search logs: Preliminary results using query rates and intervals,” in *Proceedings of the WSDM Workshop on Web Search Click Data (WSCD)*, pp. 15–19, New York, NY, USA: ACM, 2009.

[68] M. Egele, C. Kruegel, and E. Kirda, “Removing web spam links from search engine results,” *Journal in Computer Virology*, In press. Published online 22 August, 2009.

[69] N. Eiron, K. S. Curley, and J. A. Tomlin, “Ranking the Web frontier,” in *Proceedings of the 13th International Conference on World Wide Web*, pp. 309–318, New York, NY, USA: ACM Press, 2004.

[70] E. Enge, “Matt cutts interviewed by eric enge,” Article online at http://www.stonetemple.com/articles/interview-matt-cutts-012510.shtml and retrieved on 11 April 2010, March 2010.

[71] M. Erdélyi, A. A. Benczúr, J. Masanes, and D. Siklósi, “Web spam filtering in internet archives,” in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 17–20, ACM Press, 2009.

[72] J. Feigenbaum, S. Kannan, M. A. McGregor, S. Suri, and J. Zhang, “On graph problems in a semi-streaming model,” in *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 531–543, Springer: Vol. 3142 of *LNCS*, 2004.

[73] D. Fetterly, “Adversarial Information Retrieval: the manipulation of Web content,” *ACM Computing Reviews*, July 2007.

[74] D. Fetterly, M. Manasse, and M. Najork, “Spam, damn spam, and statistics: Using statistical analysis to locate spam Web pages,” in *Proceedings of the Seventh Workshop on the Web and databases (WebDB)*, pp. 1–6, June 2004.

[75] D. Fetterly, M. Manasse, and M. Najork, “Detecting phrase-level duplication on the World Wide Web,” in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 170–177, New York, NY, USA: ACM, 2005.

[76] R. Fishkin, “Lessons learned building an index of the WWW,” Retrieved 15 June 2009 from http://www.seomoz.org/blog/lessons-learned-building-an-index-of-the-www, April 2009.

[77] S. Fox, M. Madden, and A. Smith, “Digital footprints,” Pew Internet and American Life report. Retrieved 15 June 2009 from http://www.pewinternet.org/Reports/2007/Digital-Footprints.aspx, December 2007.

[78] Q. Gan and T. Suel, “Improving Web spam classifiers using link structure,” in *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 17–20, New York, NY, USA: ACM, 2007.

[79] D. Gibson, R. Kumar, and A. Tomkins, “Discovering large dense subgraphs in massive graphs,” in *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, pp. 721–732, VLDB Endowment, 2005.

[80] Y. Gil and D. Artz, “Towards content trust of Web resources,” in *Proceedings of the 15th International Conference on World Wide Web (WWW)*, pp. 565–574, New York, NY, USA: ACM, 2006.

[81] A. Gkanogiannis and T. Kalamboukis, “An algorithm for text categorization,” in *Proceedings of the 31st Annual International ACM SIGIR Conference on*

Research and Development in Information Retrieval, pp. 869–870, New York, NY, USA: ACM, 2008.

[82] A. Gkanogiannis and T. Kalamboukis, "A novel supervised learning algorithm and its use for spam detection in social bookmarking systems," in *Proceedings of the ECML/PKDD Discovery Challenge*, 2008.

[83] H. L. Gomes, B. R. Almeida, A. M. L. Bettencourt, V. Almeida, and M. J. Almeida, "Comparative graph theoretical characterization of networks of spam and legitimate email," April 2005.

[84] M. Goodstein and V. Vassilevska, "A two player game to combat WebSpam," Technical Report, Carnegie Mellon University, 2007.

[85] M. Gori and I. Witten, "The bubble of Web visibility," *Communications of the ACM*, vol. 48, no. 3, pp. 115–117, March 2005.

[86] P. Gramme and J.-F. Chevalier, "Rank for spam detection — ECML discovery challenge," in *Proceedings of the ECML/PKDD Discovery Challenge*, 2008.

[87] A. L. Granka, T. Joachims, and G. Gay, "Eye-tracking analysis of user behavior in www search," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 478–479, New York, NY, USA: ACM, 2004.

[88] J. Grappone and G. Couzin, *Search Engine Optimization: An Hour a Day.* Wiley, 2006.

[89] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *Proceedings of the 13th International Conference on World Wide Web (WWW)*, pp. 403–412, New York, NY, USA: ACM Press, 2004.

[90] D. M. Guinness, H. Zeng, Li, D. Narayanan, and M. Bhaowal, "Investigations into trust for collaborative information. repositories: A Wikipedia case study," in *Proceedings of workshop on Models of Trust for the Web (MTW06)*, May 2006.

[91] Z. Gyöngyi and H. Garcia-Molina, "Link spam alliances," in *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, pp. 517–528, 2005.

[92] Z. Gyöngyi and H. Garcia-Molina, "Spam: It's not just for inboxes anymore," *IEEE Computer Magazine*, vol. 38, no. 10, pp. 28–34, 2005.

[93] Z. Gyöngyi and H. Garcia-Molina, "Web spam taxonomy," in *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 39–47, May 2005.

[94] 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)*, pp. 576–587, Morgan Kaufmann, August 2004.

[95] Z. I. Gyöngyi, "Applications of Web link analysis," PhD thesis, Stanford University, Adviser: Hector Garcia-Molina, 2008.

[96] Z. P. Gyöngyi, P. Berkhin, H. Garcia-Molina, and J. Pedersen, "Link spam detection based on mass estimation," in *Proceedings of the 32nd International Conference on Very Large Databases (VLDB)*, pp. 439–450, 2006.

[97] H. T. Haveliwala, "Topic-sensitive PageRank," in *Proceedings of the Eleventh World Wide Web Conference (WWW)*, pp. 517–526, ACM Press, May 2002.

[98] R. M. Henzinger, R. Motwani, and C. Silverstein, "Challenges in Web search engines," *SIGIR Forum*, vol. 37, no. 2, 2002.

[99] R. Herbrich, T. Graepel, and K. Obermayer, “Large margin rank boundaries for ordinal regression,” in *Advances in Large Margin Classifiers*, (Smola, Bartlett, Schoelkopf, and Schuurmans, eds.), pp. 115–132, Cambridge, MA: MIT Press, 2000.

[100] A. Heydon and M. Najork, “Mercator: A scalable, extensible web crawler,” *World Wide Web*, vol. 2, no. 4, pp. 219–229, 1999.

[101] P. Heymann, G. Koutrika, and H. Garcia-Molina, “Fighting spam on social Web sites: A survey of approaches and future challenges,” *IEEE Internet Computing*, vol. 11, no. 6, pp. 36–45, 2007.

[102] J. Hopcroft and D. Sheldon, “Manipulation-resistant reputations using hitting time,” in *Proceedings of the Workshop on Algorithms and Models for the Web-Graph (WAW)*, pp. 68–81, Springer: Vol. 2863 of *Lecture Notes in Computer Science*, December 2007. Also appears in *Internet Mathematics 5, 5:71–90, 2009*.

[103] J. Hopcroft and D. Sheldon, “Network reputation games,” Techical Report, Cornell University, October 2008.

[104] M. Hu, E.-P. Lim, A. Sun, W. H. Lauw, and B.-Q. Vuong, “Measuring article quality in Wikipedia: Models and evaluation,” in *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management (CIKM)*, pp. 243–252, New York, NY, USA: ACM, 2007.

[105] S. Hutcheon, “Google pardons BMW website,” in *Sydney Morning Herald*, Retrieved 18 June 2009 from http://www.smh.com.au/news/breaking/google-pardons-bmw-website/2006/02/09/1139379597733.html, February 2006.

[106] N. Immorlica, K. Jain, and M. Mahdian, “Game-theoretic aspects of designing hyperlink structures,” in *Proceedings of the 2nd Workshop on Internet and Network Economics (WINE)*, pp. 150–161, Vol. 4286, Springer LNCS, December 2006.

[107] N. Immorlica, K. Jain, M. Mahdian, and K. Talwar, “Click fraud resistant methods for learning click-through rates,” in *Proceedings of the Workshop on Internet and Network Economics (WINE)*, pp. 34–45, Springer, Berlin: Vol. 3828 of *Lecture Notes in Computer Science*, 2005.

[108] M. Jamali and M. Ester, “Trustwalker: A random walk model for combining trust-based and item-based recommendation,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 397–406, New York, NY, USA: ACM, 2009.

[109] J. B. Jansen, “Click fraud,” *Computer*, vol. 40, no. 7, pp. 85–86, 2007.

[110] Q. Jiang, L. Zhang, Y. Zhu, and Y. Zhang, “Larger is better: Seed selection in link-based anti-spamming algorithms,” in *Proceedings of the 17th International Conference on World Wide Web (WWW)*, pp. 1065–1066, New York, NY, USA: ACM, 2008.

[111] N. Jindal and B. Liu, “Analyzing and detecting review spam,” in *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)*, pp. 547–552, 2007.

[112] N. Jindal and B. Liu, “Review spam detection,” in *Proceedings of the 16th International Conference on World Wide Web (WWW)*, pp. 1189–1190, New York, NY, USA: ACM Press, 2007.

[113] N. Jindal and B. Liu, “Opinion spam and analysis,” in *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pp. 219–230, New York, NY, USA: ACM, 2008.

[114] T. Joachims, “Optimizing search engines using clickthrough data,” in *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 133–142, New York, NY: ACM Press, 2002.

[115] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, “Accurately interpreting clickthrough data as implicit feedback,” in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 154–161, New York, NY, USA: ACM Press, 2005.

[116] T. Jones, D. Hawking, and R. Sankaranarayana, “A framework for measuring the impact of Web spam,” in *Proceedings of 12th Australasian Document Computing Symposium (ADCS)*, December 2007.

[117] T. Jones, D. Hawking, R. Sankaranarayana, and N. Craswell, “Nullification test collections for Web spam and SEO,” in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 53–60, New York, NY, USA: ACM, April 2009.

[118] T. Z. Jr., “Gaming the search engine, in a political season,” in *New York Times*, November 2006.

[119] D. S. Kamvar, T. M. Schlosser, and H. Garcia-Molina, “The Eigentrust algorithm for reputation management in P2P networks,” in *Proceedings of the 12th International Conference on World Wide Web (WWW)*, pp. 640–651, New York, NY, USA: ACM Press, 2003.

[120] G. Karypis and V. Kumar, “Multilevel k-way partitioning scheme for irregular graphs,” *Journal of Parallel and Distributed Computation*, vol. 48, no. 1, pp. 96–129, 1998.

[121] T. Katayama, T. Utsuro, Y. Sato, T. Yoshinaka, Y. Kawada, and T. Fukuhara, “An empirical study on selective sampling in active learning for Splog detection,” in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 29–36, ACM Press, 2009.

[122] M. J. Kleinberg, “Authoritative sources in a hyperlinked environment,” *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[123] J. Köhne, “Optimizing a large dynamically generated Website for search engine crawling and ranking,” Master’s thesis, Technical University of Delft, 2006.

[124] P. Kolari, “Detecting Spam Blogs: An Adaptive Online Approach,” PhD thesis, Department of Computer Science and Electrical Engineering, University of Maryland-Baltimore County, 2007.

[125] P. Kolari, T. Finin, A. Java, and A. Joshi, “Splog blog dataset,” Techical Report, UMBC ebiquity, 2006.

[126] P. Kolari, T. Finin, A. Java, and A. Joshi, “Towards spam detection at ping servers,” in *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, AAAI Press. Demo, March 2007.

[127] P. Kolari, A. Java, and T. Finin, “Characterizing the Splogosphere,” in *Proceedings of the 3rd Annual Workshop on the Weblogging Ecosystem*, 2006.

[128] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi, “Detecting spam blogs: A machine learning approach,” in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, July 2006.

[129] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas, “Releasing search queries and clicks privately,” in *Proceedings of the 18th International Conference on World Wide Web (WWW)*, pp. 171–180, New York, NY, USA: ACM, 2009.

[130] M. Koster, “A standard for robot exclusion,” http://www.robotstxt.org/wc/robots.html, 1996.

[131] Z. Kou, “Stacked graphical learning,” PhD thesis, School of Computer Science, Carnegie Mellon University, 2007.

[132] G. Koutrika, A. F. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina, “Combating spam in tagging systems,” in *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 57–64, New York, NY, USA: ACM Press, 2007.

[133] G. Koutrika, A. F. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina, “Combating spam in tagging systems: An evaluation,” *ACM Transactions on the Web*, vol. 2, no. 4, pp. 1–34, 2008.

[134] B. Krause, H. A. Schimitz, and G. Stumme, “The anti-social tagger — detecting spam in social bookmarking systems,” in *Proceedings of the Fourth International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, April 2008.

[135] V. Krishnan and R. Raj, “Web spam detection with anti-TrustRank,” in *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 37–40, 2006.

[136] R. Kumar, J. Novak, B. Pang, and A. Tomkins, “On anonymizing query logs via token-based hashing,” in *Proceedings of the 16th International Conference on World Wide Web (WWW)*, pp. 629–638, New York, NY, USA: ACM Press, 2007.

[137] J. Kupke and M. Ohye, “Specify your canonical,” Retrieved 18 June 2009 from http://googlewebmastercentral.blogspot.com/2009/02/specify-your-canonical.html, February 2009.

[138] N. A. Langville and D. C. Meyer, “Deeper inside PageRank,” *Internet Mathematics*, vol. 1, no. 3, pp. 335–380, 2003.

[139] N. A. Langville and D. C. Meyer, *Google’s PageRank and Beyond: The Science of Search Engine Rankings.* Princeton, NJ: Princeton University Press, 2006.

[140] H.-T. Lee, D. Leonard, X. Wang, and D. Loguinov, “Irlbot: Scaling to 6 billion pages and beyond,” *ACM Transactions on the Web*, vol. 3, no. 3, pp. 1–34, 2009.

[141] R. Lempel and S. Moran, “The stochastic approach for link-structure analysis (SALSA) and the TKC effect,” *Computer Networks*, vol. 33, no. 1–6, pp. 387–401, 2000.

[142] R. Levien and A. Aiken, “Attack-resistant trust metrics for public key certification,” in *Proceedings of the 7th USENIX Security Symposium*, pp. 229–242, 1998.

[143] J. Lewis, “Google bombs,” in *LA Weekly*, Retrieved June 1, 2009 from http://www.laweekly.com/2003-12-25/news/google-bombs, December 2003.

[144] G. Liang, "Surveying Internet usage and impact in five Chinese cities," Report of the Research Center for Social Development, Chinese Academy of Social Sciences. Retrieved 15 June 2009 from http://news.bbc.co.uk/1/shared/bsp/hi/pdfs/10_02_06_china.pdf, November 2005.

[145] M. Lifantsev, "Voting model for ranking Web pages," in *Proceedings of the International Conference on Internet Computing (IC)*, (P. Graham and M. Maheswaran, eds.), pp. 143–148, CSREA Press, June 2000.

[146] J.-L. Lin, "Detection of cloaked Web spam by using tag-based methods," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7493–7499, May 2009.

[147] Y.-R. Lin, H. Sundaram, Y. Chi, J. Tatemura, and L. B. Tseng, "Splog detection using self-similarity analysis on blog temporal dynamics," in *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 1–8, New York, NY, USA: ACM Press, 2007.

[148] Y.-R. Lin, H. Sundaram, Y. Chi, J. Tatemura, and L. B. Tseng, "Detecting splogs via temporal dynamics using self-similarity analysis," *ACM Transations on the Web*, vol. 2, no. 1, pp. 1–35, 2008.

[149] T. Liu, "Analyzing the importance of group structure in the Google PageRank algorithm," Master's thesis, Rensselaer Polytechnic Institute, November 2004.

[150] Y. Liu, R. Cen, M. Zhang, S. Ma, and L. Ru, "Identifying Web spam with user behavior analysis," in *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 9–16, New York, NY, USA: ACM, 2008.

[151] Y. Liu, B. Gao, T.-Y. Liu, Y. Zhang, Z. Ma, S. He, and H. Li, "BrowseRank: Letting web users vote for page importance," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 451–458, New York, NY, USA: ACM, 2008.

[152] J. Ma, K. L. Saul, S. Savage, and M. G. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious urls," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1245–1254, New York, NY, USA: ACM, 2009.

[153] C. C. Mann, "How click fraud could swallow the internet," *Wired*, vol. 14, no. 1, January 2006.

[154] D. C. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[155] B. Markines, C. Cattuto, and F. Menczer, "Social spam detection," in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 41–48, New York, NY, USA: ACM, 2009.

[156] K. Marks and T. Celik, "Microformats: The rel=nofollow attribute," Techical Report, Technorati, 2005. Online at http://microformats.org/wiki/rel-nofollow. Last accessed 29 January 2009.

[157] K. Marks and T. Celik, "Microformats: Vote links," Technical Report, Technorati, 2005. Online at http:// microformats.org/wiki/vote-links. Last accessed 29 January 2009.

[158] S. Marti and H. Garcia-Molina, "Taxonomy of trust: Categorizing P2P reputation systems," *Computer Networks*, vol. 50, no. 4, pp. 472–484, March 2006.

[159] J. Martinez-Romo and L. Araujo, “Web spam identification through language model analysis,” in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 21–28, ACM Press, 2009.

[160] K. Mason, “Detecting Colluders in PageRank: Finding Slow Mixing States in a Markov Chain,” PhD thesis, Department of Engineering Economic Systems and Operations Research, Stanford University, September 2005.

[161] P. Massa and C. Hayes, “Page-reRank: Using trusted links to re-rank authority,” in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 614–617, 2005.

[162] A. Mathes, “Filler Friday: Google Bombing,” Retrieved June 1, 2009 from http://uber.nu/2001/04/06/, April 2001.

[163] T. McNichol, “Engineering Google results to make a point,” *New York Times*, January 2004.

[164] T. P. Metaxas and J. Destefano, “Web spam, propaganda and trust,” in *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, May 2005.

[165] A. Metwally, D. Agrawal, and E. A. Abbadi, “Detectives: Detecting coalition hit inflation attacks in advertising networks streams,” in *Proceedings of the 16th International Conference on World Wide Web (WWW)*, pp. 241–250, New York, NY, USA: ACM Press, 2007.

[166] A. G. Mishne, “Applied Text Analytics for Blogs,” PhD thesis, University of Amsterdam, April 2007.

[167] G. Mishne, D. Carmel, and R. Lempel, “Blocking blog spam with language model disagreement,” in *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, May 2005.

[168] T. Moore and R. Clayton, “Evil searching: Compromise and recompromise of internet hosts for phishing,” in *Financial Cryptography and Data Security*, pp. 256–272, Springer, 2009.

[169] M. Moran and B. Hunt, *Search Engine Marketing, Inc.* Upper Saddle River, NJ: IBM Press, 2006.

[170] A. Moshchuk, T. Bragin, D. S. Gribble, and M. H. Levy, “A crawler-based study of spyware on the web,” in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, pp. 17–33, February 2006.

[171] R. Moulton and K. Carattini, “A quick word about Googlebombs,” Retrieved June 1, 2009 from http://googlewebmastercentral.blogspot.com/2007/01/quick-word-about-googlebombs.html, January 2007.

[172] L. Mui, M. Mohtashemi, and A. Halberstadt, “A computational model of trust and reputation,” in *Proceedings of the 35th Hawaii International Conference on System Science (HICSS)*, 2002.

[173] M. Najork, “System and method for identifying cloaked web servers,” U.S. Patent 6,910,077 (issued June 2005), 2002.

[174] N. Neubauer, R. Wetzker, and K. Obermayer, “Tag spam creates large non-giant connected components,” in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 49–52, ACM Press, 2009.

[175] B. News, "Miserable failure' links to Bush: George W Bush has been Google bombed," http://news.bbc.co.uk/2/hi/americas/3298443.stm, December 2003.

[176] L. Nie, D. B. Davison, and B. Wu, "Incorporating trust into Web authority," Technical Report LU-CSE-07-002, Department of Computer Science and Engineering, Lehigh University, 2007.

[177] L. Nie, B. Wu, and D. B. Davison, "A cautious surfer for PageRank," in *Proceedings of the 16th International Conference on World Wide Web (WWW)*, pp. 1119–1120, New York, NY, USA: ACM Press, 2007.

[178] L. Nie, B. Wu, and D. B. Davison, "Winnowing wheat from the chaff: Propagating trust to sift spam from the Web," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 869–870, July 2007.

[179] Y. Niu, Y.-M. Wang, H. Chen, M. Ma, and F. Hsu, "A quantitative study of forum spamming using context-based analysis," in *Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS)*, pp. 79–92, February 2007.

[180] A. Ntoulas, "Crawling and searching the Hidden Web," PhD thesis, University of California at Los Angeles, Los Angeles, CA, USA, 2006. Adviser-Cho, Junghoo.

[181] A. Ntoulas, J. Cho, and C. Olston, "What's new on the Web?: The evolution of the Web from a search engine perspective," in *Proceedings of the 13th International Conference on World Wide Web*, pp. 1–12, New York, NY, USA: ACM Press, 2004.

[182] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting spam Web pages through content analysis," in *Proceedings of the 15th International Conference on World Wide Web (WWW)*, pp. 83–92, May 2006.

[183] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," Techincal Report, Stanford University, 1998. Available from http://dbpubs.stanford.edu/pub/1999-66.

[184] R. C. Palmer, B. P. Gibbons, and C. Faloutsos, "ANF: A fast and scalable tool for data mining in massive graphs," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 81–90, New York, NY, USA: ACM Press, 2002.

[185] K. Park, V. S. Pai, K.-W. Lee, and S. Calo, "Securing Web service by automatic robot detection," in *Proceedings of the USENIX Annual Technical Conference*, pp. 255–260, 2006.

[186] J.-X. Parreira, D. Donato, C. Castillo, and G. Weikum, "Computing trusted authority scores in peer-to-peer networks," in *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 73–80, ACM Press, May 2007.

[187] L. A. Penenberg, "Click fraud threatens Web," Wired, October 2004.

[188] L. A. Penenberg, "Legal showdown in search fracas," in *Wired*, Retrieved 18 June 2009 from http://www.wired.com/culture/lifestyle/news/2005/09/68799, September 2005.

[189] A. Perkins, "The classification of search engine spam," Available online at http://www.silverdisc.co.uk/articles/spam-classification/, September 2001.

[190] J. Piskorski, M. Sydow, and D. Weiss, "Exploring linguistic features for Web spam detection: A preliminary study," in *Proceedings of the Fourth International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 25–28, New York, NY, USA: ACM, 2008.

[191] G. Price, "Google and Google bombing now included New Oxford American Dictionary," Retrieved June 1, 2009 from http://blog.searchenginewatch.com/blog/050516-184202, May 2005.

[192] N. Provos, P. Mavrommatis, A. M. Rajab, and F. Monrose, "All your iFRAMEs point to us," in *Proceedings of the 17th USENIX Security Symposium*, pp. 1–15, Berkeley, CA, USA: USENIX Association, 2008.

[193] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu, "The ghost in the browser analysis of web-based malware," in *Proceedings of the First Workshop on Hot Topics in Understanding Botnets (HotBots)*, 2007.

[194] X. Qi and D. B. Davison, "Knowing a web page by the company it keeps," in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 228–237, New York, NY: ACM Press, November 2006.

[195] X. Qi and D. B. Davison, "Web page classification: Features and algorithms," *ACM Computing Surveys*, vol. 41, no. 2, February 2009.

[196] X. Qi, L. Nie, and D. B. Davison, "Measuring similarity to detect qualified links," in *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 49–56, May 2007.

[197] R. J. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.

[198] S. R. Rainwater, "Nigritude ultramarine FAQ," Retrieved June 1, 2009 from http://www.nigritudeultramarines.com/, 2005.

[199] Y. Rasolofo and J. Savoy, "Term proximity scoring for keyword-based retrieval systems," in *ECIR*, pp. 207–218, 2003.

[200] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[201] M. Richardson, A. Prakash, and E. Brill, "Beyond PageRank: Machine learning for static ranking," in *Proceedings of the 15th International Conference on World Wide Web (WWW)*, pp. 707–715, New York, NY, USA: ACM Press, May 2006.

[202] G. Roberts and J. Rosenthal, "Downweighting tightly knit communities in World Wide Web rankings," *Advances and Applications in Statistics (ADAS)*, vol. 3, pp. 199–216, 2003.

[203] S. Robertson, S. Walker, M. M. Beaulieu, M. Gatford, and A. Payne, "Okapi at TREC-4," in *NIST Special Publication 500-236: The Fourth Text REtrieval Conference (TREC-4)*, pp. 73–96, 1995.

[204] G. Salton, A. Wong, and S. C. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, November 1975.

[205] Y. Sato, T. Utsuro, Y. Murakami, T. Fukuhara, H. Nakagawa, Y. Kawada, and N. Kando, "Analysing features of Japanese splogs and characteristics of keywords," in *Proceedings of the Fourth International Workshop on Adversarial*

Information Retrieval on the Web (AIRWeb), pp. 33–40, New York, NY, USA: ACM, 2008.

[206] C. Schmidt, "Page hijack: The 302 exploit, redirects and Google," http://clsc.net/research/google-302-page-hijack.htm, 2005.

[207] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[208] D. Sheldon, "Manipulation of PageRank and Collective Hidden Markov Models," PhD thesis, Cornell University, 2009.

[209] G. Shen, B. Gao, T.-Y. Liu, G. Feng, S. Song, and H. Li, "Detecting link spam using temporal information," in *Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM)*, December 2006.

[210] N. Shrivastava, A. Majumder, and R. Rastogi, "Mining (social) network graphs to detect random link attacks," in *Proceedings of the International Conference on Data Engineering (ICDE)*, IEEE CS Press, April 2008.

[211] F. Silvestri, "Mining query logs: Turning search usage data into knowledge," *Foundations and Trends in Information Retrieval*, vol. 3, 2009.

[212] A. Singhal, "Challenges in running a commercial search engine," Keynote presentation at SIGIR 2005, August 2005.

[213] M. Sirivianos, X. Yang, and K. Kim, "FaceTrust: Assessing the credibility of online personas via social networks," Technical Report, Duke University, 2009. Retrieved 15 June 2009 from http://www.cs.duke.edu/~msirivia/publications/facetrust-tech-report.pdf.

[214] M. Sobek, "PR0 — Google's PageRank 0 penalty," http://pr.efactory.de/e-pr0.shtml, 2002.

[215] A. Stassopoulou and D. M. Dikaiakos, "Web robot detection: A probabilistic reasoning approach," *Computer Networks*, vol. 53, no. 3, pp. 265–278, February 2009.

[216] A.-J. Su, C. Y. Hu, A. Kuzmanovic, and C.-K. Koh, "How to improve your Google ranking: Myths and reality," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pp. 50–57, Vol. 1, IEEE, 2010.

[217] M. K. Svore, Q. Wu, C. J. C. Burges, and A. Raman, "Improving Web spam classification using rank-time features," in *Proceedings of the Third International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 9–16, New York, NY, USA: ACM Press, 2007.

[218] P.-N. Tan and V. Kumar, "Discovery of Web robot sessions based on their navigational patterns," *Data Mining and Knowledge Discovery*, vol. 6, no. 1, pp. 9–35, 2002.

[219] E. Tardos and T. Wexler, "Network formation games and the potential function method," in *Algorithmic Game Theory*, (N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, eds.), Cambridge University Press, 2007.

[220] C. Tatum, "Deconstructing Google bombs: A breach of symbolic power or just a goofy prank?," *First Monday*, vol. 10, no. 10, October 2005.

[221] A. Thomason, "Blog spam: A review," in *Proceedings of Conference on Email and Anti-Spam (CEAS)*, August 2007.

[222] A. Toffler, "The Third Wave," Bantam Books, 1980.

[223] N. Tran, B. Min, J. Li, and L. Submaranian, “Sybil-resilient online content voting,” in *Proceedings of the 6th Symposium on Networked System Design and Implementation (NSDI)*, 2009.

[224] T. Urvoy, E. Chauveau, P. Filoche, and T. Lavergne, “Tracking Web spam with HTML style similarities,” *ACM Transactions on the Web*, vol. 2, no. 1, 2008.

[225] T. Urvoy, T. Lavergne, and P. Filoche, “Tracking Web spam with hidden style similarity,” in *Proceedings of the Second International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, August 2006.

[226] N. Vidyasaga, “India’s secret army of online ad ‘clickers’,” *The Times of India*, May 2004.

[227] L. A. von Ahn and L. Dabbish, “Labeling images with a computer game,” in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pp. 319–326, ACM Press, 2004.

[228] K. Walsh and G. E. Sirer, “Fighting peer-to-peer spam and decoys with object reputation,” in *Proceedings of the ACM SIGCOMM Workshop on Economics of Peer-to-Peer systems (P2PECON)*, pp. 138–143, New York, NY, USA: ACM Press, 2005.

[229] W. Wang, G. Zeng, M. Sun, H. Gu, and Q. Zhang, “EviRank: An evidence based content trust model for Web spam detection,” in *Proceedings of Workshop on Emerging Trends of Web Technologies and Applications WAIM/APWeb*, pp. 299–307, 2007.

[230] Y.-M. Wang, D. Beck, X. Jiang, and R. Roussev, “Automated web patrol with Strider HoneyMonkeys: Finding web sites that exploit browser vulnerabilities,” in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, February 2006.

[231] Y.-M. Wang, M. Ma, Y. Niu, and H. Chen, “Spam double-funnel: Connecting Web spammers with advertisers,” in *Proceedings of the 16th International Conference on World Wide Web (WWW)*, pp. 291–300, New York, NY, USA: ACM Press, 2007.

[232] S. Webb, “Automatic Identification and Removal of Low Quality Online Information,” PhD thesis, College of Computing, Georgia Institute of Technology, December 2008.

[233] S. Webb, J. Caverlee, and C. Pu, “Introducing the Webb spam corpus: Using email spam to identify Web spam automatically,” in *Proceedings of the Third Conference on Email and Anti-Spam (CEAS)*, July 2006.

[234] 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 (CIKM)*, pp. 339–348, New York, NY, USA: ACM, 2008.

[235] H. I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, 1999.

[236] B. Wu, “Finding and Fighting Search Engine Spam,” PhD thesis, Department of Computer Science and Engineering, Lehigh University, March 2007.

[237] B. Wu and K. Chellapilla, “Extracting link spam using biased random walks from spam seed sets,” in *Proceedings of the 3rd International Workshop on*

Adversarial Information Retrieval on the Web (AIRWeb), pp. 37–44, New York, NY, USA: ACM Press, 2007.

[238] B. Wu and D. B. Davison, "Cloaking and redirection: A preliminary study," in *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005.

[239] B. Wu and D. B. Davison, "Detecting semantic cloaking on the Web," in *Proceedings of the 15th International World Wide Web Conference (WWW)*, pp. 819–828, ACM Press, 2006.

[240] B. Wu and D. B. Davison, "Undue influence: Eliminating the impact of link plagiarism on Web search rankings," in *Proceedings of The 21st ACM Symposium on Applied Computing (SAC)*, pp. 1099–1104, April 2006.

[241] B. Wu and D. B. Davison, "Identifying link farm spam pages," in *Special interest tracks and posters of the 14th International Conference on World Wide Web (WWW)*, pp. 820–829, New York, NY, USA: ACM Press, 2005.

[242] B. Wu, V. Goel, and D. B. Davison, "Propagating trust and distrust to demote Web spam," in *Workshop on Models of Trust for the Web (MTW)*, May 2006.

[243] B. Wu, V. Goel, and D. B. Davison, "Topical TrustRank: Using topicality to combat Web spam," in *Proceedings of the 15th International World Wide Web Conference (WWW)*, pp. 63–71, ACM Press, May 2006.

[244] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[245] H. Yu, M. Kaminsky, B. P. Gibbons, and A. Flaxman, "SybilGuard: Defending against sybil attacks via social networks," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pp. 267–278, New York, NY, USA: ACM Press, 2006.

[246] K. Yusuke, W. Atsumu, K. Takashi, B. B. Bahadur, and T. Toyoo, "On a referrer spam blocking scheme using Bayesian filter," *Joho Shori Gakkai Shinpojiumu Ronbunshu*, vol. 1, no. 13, pp. 319–324, In Japanese, 2005.

[247] H. Zhang, A. Goel, R. Govindan, K. Mason, and B. V. Roy, "Making eigenvector-based reputation systems robust to collusion," in *Proceedings of the Third Workshop on Web Graphs (WAW)*, pp. 92–104, Springer: Vol. 3243 of *Lecture Notes in Computer Science*, October 2004.

[248] L. Zhang, Y. Zhang, Y. Zhang, and X. Li, "Exploring both content and link quality for anti-spamming," in *Proceedings of the Sixth IEEE International Conference on Computer and Information Technology (CIT)*, IEEE Computer Society, 2006.

[249] Y. Zhang and A. Moffat, "Some observations on user search behavior," *Australian Journal of Intelligent Information Processing Systems*, vol. 9, no. 2, pp. 1–8, 2006.

[250] L. Zhao, Q. Jiang, and Y. Zhang, "From good to bad ones: Making spam detection easier," in *Proceedings of the Eighth IEEE International Conference on Computer and Information Technology Workshops (CIT)*, IEEE Computer Society, 2008.

[251] B. Zhou, “Mining page farms and its application in link spam detection,” Master’s thesis, Simon Fraser University, 2007.

[252] B. Zhou and J. Pei, “Sketching landscapes of page farms,” in *Proceedings of the 7th SIAM International Conference on Data Mining (SDM)*, SIAM, April 2007.

[253] B. Zhou, J. Pei, and Z. Tang, “A spamicity approach to Web spam detection,” in *Proceedings of the SIAM International Conference on Data Mining (SDM)*, April 2008.

[254] D. Zhou, C. J. C. Burges, and T. Tao, “Transductive link spam detection,” in *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 21–28, New York, NY, USA: ACM Press, 2007.

[255] C.-N. Ziegler and G. Lausen, “Propagation models for trust and distrust in social networks,” *Information Systems Frontiers*, vol. 7, no. 4–5, pp. 337–358, December 2005.

[256] R. P. Zimmermann, *The Official PGP User’s Guide.* Cambridge, MA: MIT Press, 1995.

[257] J. Zittrain, *The Future of the Internet — And How to Stop It.* Yale University Press, April 2008.