Recognizing Nepotistic Links on the Web

Brian D. Davison
Department of Computer Science
Rutgers, The State University of New Jersey
New Brunswick, NJ 08903 USA
davison@cs.rutgers.edu

Abstract

The use of link analysis and page popularity in search engines has grown recently to improve query result rankings. Since the number of such links contributes to the value of the document in such calculations, we wish to recognize and eliminate *nepotistic links* — links between pages that are present for reasons other than merit. This paper explores some of the issues surrounding the question of what links to keep, and we report high accuracy in initial experiments to show the potential for using a machine learning tool to automatically recognize such links.

Introduction

Recently there has been growing interest in the research community in using analysis of link information of the Web (Bharat & Henzinger 1998; Brin & Page 1998; Chakrabarti *et al.* 1998b; 1998a; Chakrabarti, Dom, & Indyk 1998; Gibson, Kleinberg, & Raghavan 1998a; 1998b; Kleinberg 1998; Page *et al.* 1998; Chakrabarti *et al.* 1999a; Chakrabarti, van den Berg, & Dom 1999; Chakrabarti *et al.* 1999b; Henzinger *et al.* 1999; Kleinberg *et al.* 1999; Davison *et al.* 1999; Dean & Henzinger 1999). Much of this effort is to improve the retrieval mechanisms available so that search engines can better rank the results of a query. At least two commercial efforts (Google Inc. 2000; IBM Almaden Research Center 2000) are based predominantly on these ideas, and many of the more established search engines have adopted popularity and link analysis components to their ranking methods (Sullivan 2000).

Popularity and link analysis algorithms are based significantly on the links that point to a document. Since the number of such links contributes to the value of the document, we wish to recognize *nepotistic links* so that their effect can be reduced or eliminated. While a typical definition of nepotism includes favoritism to relatives, we are using the broader interpretation of "bestowal of patronage in consideration of relationship, rather than of merit." (C. & G. Merriam Co. 1913). Thus we include links between pages that are related because of common administrative control but also links that effect some reward (e.g. advertising or paid links). Nepotistic links are typically considered undesirable, and so it is useful to eliminate them before calculating the popularity or status of a page.

Finding link nepotism is similar to, but distinct from the problem of identifying duplicate pages or mirrored sites (Bharat & Broder 1999; Bharat *et al.* 1999; Shivakumar & Garcia-Molina 1998). While mirrored pages are an instance of the problem we are trying to address (that one entity gets to "vote" more often than it should), we are only considering (in this paper) the cases in which pages (mirrored or not) are pointing to a target for reasons other than merit.

Instead of link popularity, some services (e.g. DirectHit (Ask Jeeves, Inc. 2000) which is used by many of the major engines including Lycos (Lycos, Inc. 2000), HotBot (Wired Digital Inc. 2000), and MSN Search (Microsoft Corporation 2000)) collect and analyze usage popularity. While calculations based on usage popularity may benefit from knowing about nepotistic links, we have not considered their effects.

In this short paper, we explore the issues surrounding the question of what links to keep, and we report preliminary results on the use of a machine learning tool operating on a hand-generated feature set to automatically recognize nepotistic links.

The Issues

Typically, some form of popularity or status is calculated, based in large part on the number of links that point to that document. However, calculations that use all links that point to a document are liable to be misled by a number of cases, such as:

- A site with persistent, integrated navigational menu on all pages. For such a site with n pages, means that all pages on the menu have at least n incoming links.
- Link-based spam: content creators that recognize the ranking method may generate lots of pages with links to the pages to be ranked higher.

The second is an example of the problem of "search engine spam", in which pages submitted to search engines are manipulated in ways to achieve (undeserved) higher rankings. Some link-oriented search engines have claimed a resistance to traditional spam techniques[1], which have focussed on textual content (irrelevant keywords, invisible text, repeated

[1] E.g. Google, at http://www.google.com/adv_features.html says: "Our technology prevents manipulation of returned results by marketers or spammers."

terms, etc), but are instead vulnerable to linkage spam. This influence can be found in varying amounts from sites with various levels of content quality.

`About.com`, for example, is a high-quality site that not only uses an integrated navigational menu, but separates much of its content into URLs with distinct host-names. Intentional or not, this has the effect of getting more pages indexed and raising its relevance rankings with many search engines, based on the matching of keywords in host-names and by getting around simple heuristics to recognize internal links (see below). At Google, for example, out of the first 200 pages listed as pointing to `www.about.com`, only 5 were from domains other than `about.com`. The rest were subject pages within the site, such as #108: `http://billiardspool.about.com/sports/billiardspool/mbody.htm`.

Another example is the `doorkey.com` site, which (as of this writing) consists of a large set of pages that link to one another, but have little or no actual content (and sometimes include obvious keyword placement) but just link to advertisers and other doorkey-operated sites (many of which are domain names that are only a typo away from a popular domain, such as `eba6.com` and `mapquesy.com`). Finally, there are networks of sites in which otherwise unrelated sites that point to one another (for the explicit purpose of raising search engine rankings), such as `linkstoyou.com`.

There are a number of general approaches to combat this problem, including:

- maintain an exception list of pages that are abusing the incoming links
- use a heuristic to drop "internal" links (pre-processing)
- recognize when the results of a search are influenced by the 'spam' and adjust the results accordingly (post-processing)

It is not always clear exactly what methods the commercial search engines are using; however many researchers have suggested some form of the second method (e.g. (Kleinberg 1998)). Unfortunately, any simple heuristic is likely to have significant exceptions. For example:

- marking links between pages with identical host-names as internal
  - drops links between different users' home pages at the same site.
  - preserves links between different hosts in the same domain.
- marking links between pages with identical domain names
  - drops links between different users' pages at same site (as above).
  - drops links between departments of same organization (regardless of whether they are on different servers).

Both additionally allow links from different domains and advertising to affect link analysis.

In general, the problem is not really to identify which links are internal, but instead to recognize which links should be used and which should not be used in link calculations. Often this means keeping links between pages from two different people/groups/owners, and dropping those between pages under the same administrative control, regardless of differences in domain name, content, etc. Often, one would like to treat links as votes, and we wish to prevent voting for one's self. Additionally, links that exist as a byproduct of advertising (such as some banner swap services) are unlikely to be useful for link analysis. If links are dropped, the resulting graph is smaller and faster to analyze. Dropping is not the only option; if link weights are supported, then the weights for these links can be reduced significantly in comparison to those that are believed to be useful.

In truth, the issue is even more complex than described above. In link analysis, the links between pages are often considered to be recommendations. And as such, unbiased recommendations are desirable. However, some companies with networks of independent sites (e.g. companies like `internet.com`, `about.com`, `go2net.com`, etc.) may have both biased and unbiased links between sites (which may be indistinguishable), and in some cases, successful links may predate and have encouraged commercial partnerships or corporate takeovers. Additionally, even biased links facilitate traffic flow, which is important for usage popularity. Finally, to prevent negative publicity, search engines are unlikely to use any mechanism that smacks (at least overtly) of censorship. Therefore, we believe most will avoid the use of an exception list, which additionally has the drawback of only covering the previously identified cases.

The use of a post-processing approach has some advantages, such as keeping the primary algorithm and datasets pure, but is highly algorithm-specific, and so we do not consider it further. Instead, we focus on the use of pre-processing to clean and reduce the link dataset before calculations are performed. In the next section, we consider the use of machine learning techniques to learn a potentially complex rule to recognize nepotistic links.

Preliminary Experiments

In order to assess the potential for automatic recognition of nepotistic links, we performed some initial classification experiments. The C4.5 decision tree package (Quinlan 1993) was selected for these preliminary tests because of its value as a popular and well-understood learning system, but a variety of other approaches could be equally useful.

Data Sets

Two data sets were used. For the first, we manually labeled 1536 links (that is, pairs of pages in which the first contains a reference to the second) selected arbitrarily to include data points from many types of pages, and then cleaned the labelings so that each link had exactly one label, specifying whether or not the link is nepotistic. The primary concern with this dataset is that it was selected with the task in mind, and thus is unlikely to be representative of the links on the Web as a whole. Therefore, we also generated a second set of data.

For this second dataset we randomly sampled the link

```
FromPath contains tilde = 0:
|   From contains Simple gTLD = 0:
|   |   Domains are same = 0: -1 (35.0)
|   |   Domains are same = 1: 1 (30.0/1.0)
|   From contains Simple gTLD = 1:
|   |   FromPage has > 200 links = 1: 1 (812.0)
|   |   FromPage has > 200 links = 0:
|   |   |   Same DNS servers = 0:
|   |   |   |   Same contact email = 1: 1 (5.0)
|   |   |   |   Same contact email = 0:
|   |   |   |   |   Pages share > 10% links = 0: -1 (62.0/7.0)
|   |   |   |   |   Pages share > 10% links = 1: 1 (3.0/1.0)
|   |   |   Same DNS servers = 1:
|   |   |   |   Complete hostnames are same = 1: 1 (374.0)
|   |   |   |   Complete hostnames are same = 0:
|   |   |   |   |   Domains are same = 0: 1 (68.0/1.0)
|   |   |   |   |   Domains are same = 1:
|   |   |   |   |   |   Pages share > 20% links = 1: 1 (18.0)
|   |   |   |   |   |   Pages share > 20% links = 0:
|   |   |   |   |   |   |   FromPage has > 100 links = 1: 1 (14.0)
|   |   |   |   |   |   |   FromPage has > 100 links = 0:
|   |   |   |   |   |   |   |   FromPage has <= 5 links = 1: 1 (2.0)
|   |   |   |   |   |   |   |   FromPage has <= 5 links = 0:
|   |   |   |   |   |   |   |   |   1 path components are same = 1: -1 (5.0/2.0)
|   |   |   |   |   |   |   |   |   1 path components are same = 0:
|   |   |   |   |   |   |   |   |   |   FromPage has <= 10 links = 0: 1 (21.0/6.0)
|   |   |   |   |   |   |   |   |   |   FromPage has <= 10 links = 1: -1 (5.0/2.0)
FromPath contains tilde = 1:
|   1 path components are same = 0: -1 (56.0)
|   1 path components are same = 1: 1 (26.0)
```

Figure 1: One decision tree generated by C4.5. Each line shows the test, its validity, and for leaf nodes, whether the rule is to classify as nepotistic (labeled 1) or good (labeled -1), and the training set statistics showing the number of cases correctly covered followed by the number incorrectly covered, if any (in parantheses).

dataset from over 7 million pages of the DiscoWeb search engine (Davison *et al.* 1999) for links in which both the source and the target had been indexed. 750 such links were then manually labeled. While potentially containing less bias, this dataset is likely to under-represent some types of pages because of the small sample size.

The overall class probabilities of the two datasets differ: 89.5% of the links in the first set were labeled nepotistic, vs. 72.8% of the second. When described by all possible features (described below), the first data set contained 255 unique cases, vs. 535 distinct cases in the second, and of these, the cases labeled nepotistic comprised 72.1% and 78.6% respectively.

Feature Definition

A non-exhaustive set of seventy-five binary features were defined manually, including tests to see if:

- Page titles or descriptions were identical
- Page descriptions overlapped at least some percentage of text
- Complete host-names were identical
- Domain names were identical
- Host-names without domains were identical
- Some number of initial IP address octets were identical
- Some number of initial path components were identical
- The parent (from) page had more than some number of outgoing links
- The pages shared at least some percentage of outgoing links
- The parent page points to `linkstoyou.com`
- The parent page description contains the term “LinksToYou”
- The paths contained keywords like “home”, “users”, or the character ‘~’ (tilde)
- The domains were simple Global Top-Level Domains (i.e. `.com`, `.net`, `.org`, and `.edu`)
- The domains had the same contact email address (checked only for the simple gTLD domains)
- The domains had the same DNS servers (checked only for the simple gTLD domains)

These features were defined quickly with the expectation that that additional features may be required, and that some kind of feature selection would be useful, and possibly even feature extraction would be needed to determine additional

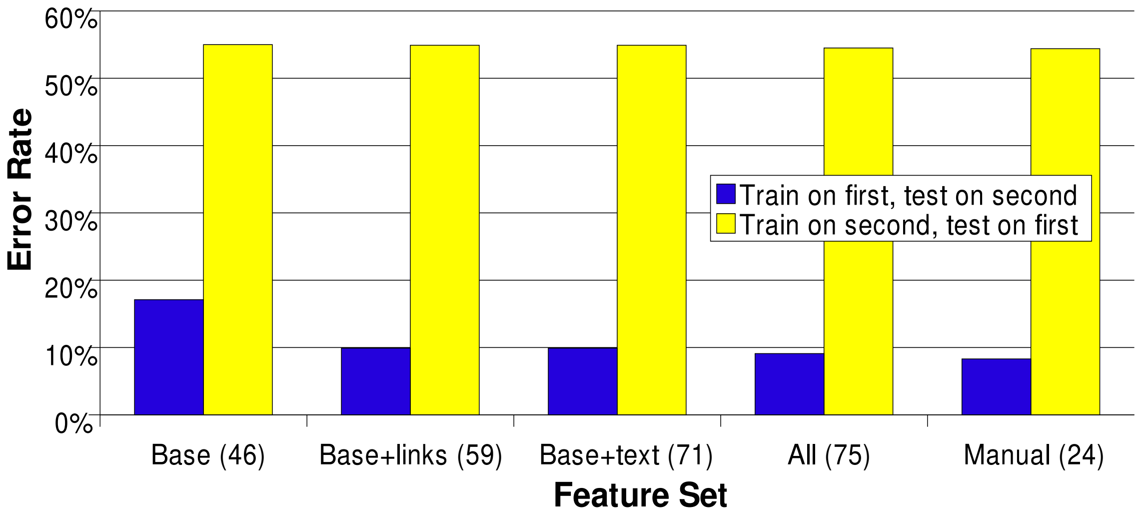

Figure 2: Error-rate of single decision tree for each of five feature sets (with the number of features in parentheses).

strings on which to test. See figure 1 for an example of these features in use in a decision tree.

To begin the process of feature selection based on domain knowledge, the features were placed in the following groups:

- **base set**: features about the URL strings for the pair of pages
- **base+links**: the above plus outgoing links and IP addresses for the pair of pages
- **base+text**: base+links plus the title and description text for the pair of pages
- **all**: everything above plus tests about contact email and dns servers for the pair of page domains (only available for domains in the simple gTLD domains)

These groups roughly correspond to the amount of time required to evaluate them. The base set is simple, requiring only the URL strings themselves. The base+links and base+text sets use information typically already present in a search engine database. The final group contains all features, including some that require access to external (potentially slow) domain databases. A fifth group, termed **manual**, was selected manually using one third of the features to try to improve accuracy (which it does in the single decision tree case).

Results

Experiments were run using C4.5 (using default options, no pruning) with each set of features. Figure 2 shows the error rates for a decision tree generated when trained on one set and evaluated on the other. The large difference in error rates shown in this graph gives strong evidence of a difference in distributions between the two datasets. The example in figure 1 contains the decision tree generated by training on the first dataset and testing on the second, using the manual feature set.

We also performed experiments with ten-fold cross-validation on either data set alone and together, shown in figure 3. Very high average accuracy (>97%) is achieved on the first data set. Average accuracies on second set and combined sets are not quite as good, at >91% and >93%, respectively.

For reference, we have also evaluated some common "internal link" heuristics. When labeling a link containing matching host-names as nepotistic, the overall error rates are 64% for the first data set, and 18% for the second. The allocation of errors are shown below:

first		data set	**second**	
(a)	(b)	<= classified as =>	(a)	(b)
140	21	(a): good link	203	1
963	412	(b): nepotistic link	134	412

When using only using matching domain names as the heuristic for nepotistic links, the overall error rates are 60.6% and 14.4%, respectively for the first and second data sets, as shown below:

first		data set	**second**	
(a)	(b)	<= classified as =>	(a)	(b)
121	40	(a): good	201	3
892	483	(b): nepotistic	103	443

Like the simple heuristics above, the decision trees generated errors that were highly skewed toward retaining links that should have been marked as nepotistic. This is, in fact, a desirable quality — one would prefer to allow a site to some undue influence rather than potentially prevent a site from scoring well.

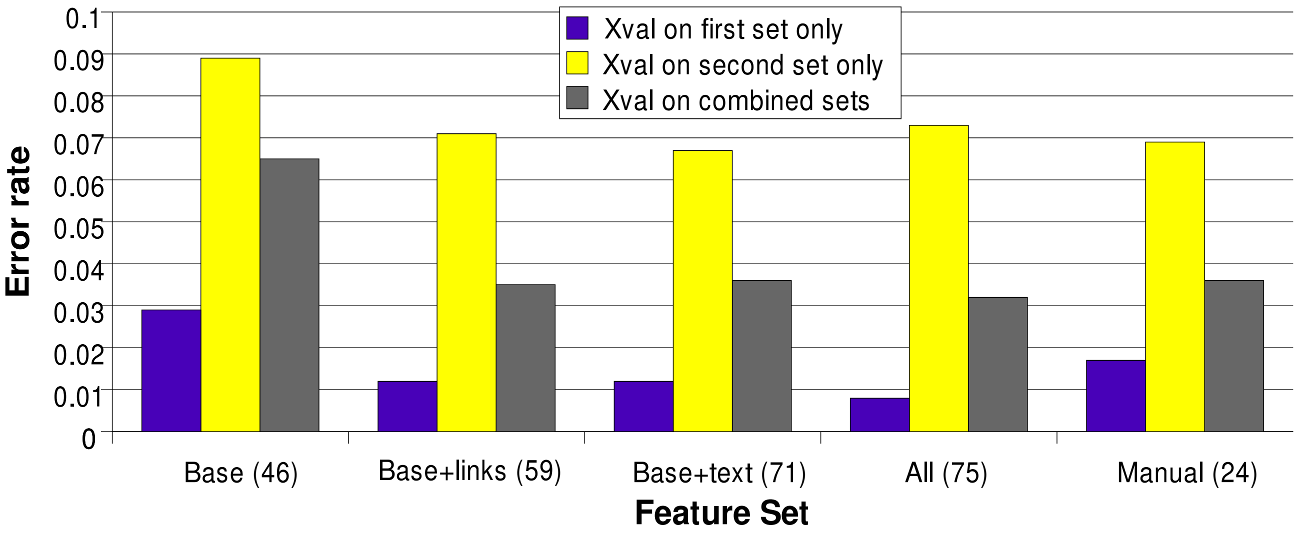

Figure 3: Average error-rate of ten-fold cross-validation for each of five feature sets (with the number of features in parentheses).

Discussion

Data Sets and Results

The two datasets used in this paper represent a typical scenario: A search service, desiring to weed out linkage spam, assembles a representative set of example links that include both good links and nepotistic links that have been encountered. With only this dataset, it is easy to generate rules that cover it well. The second dataset then represents the actual links across the web (or at least across the service's link data). As expected, a decision tree generated from the first data set doesn't do quite as well on the second. The fact that the decision tree trained on the second set performs poorly on the first suggests that the first dataset focuses on a few kinds of examples, and that the second does not cover those kinds well. This too is relevant for two reasons: the second dataset is likely to be too small to be truly representative, and the first dataset (being manually collected from interesting or useful examples) is likely to contain unusual or otherwise difficult to classify cases (because that is what would get them noticed and put into the set in the first place).

Therefore, we take the results presented here as initial evidence for the utility of automated learning methods for this problem. We hesitate to make strong claims about these results as the data sets are quite small samples of the links on the Web, and decision trees but one point in a large space of possible learning methods that could be applied.

Level of Effort

Online commercial search engines are very concerned about response-time and query costs. Some engines will respond with partial data so that response-time is minimized, and to restrict the amount of cpu time dedicated to any one query. We have not seriously considered computational and space requirements for this work, but expect that this process would need to be performed off-line — perhaps incrementally while crawling or en masse for services whose dataset is updated by batches. In any case, collecting and/or calculating some of the features is likely to take enough time that it should stay out of a dynamic request stream, but the features may change over time, so a static analysis is insufficient.

Future Work

While link nepotism addresses the issues of "voting for oneself" and having your vote purchased, there are other perversions of the traditional voting mechanisms. As mentioned earlier, one that is of concern is the issue of duplicate pages and mirror sites — it may be desirable to consider links from those pages as "duplicate votes". In general this problem can manifest itself in multiple links from pages of one organization pointing to a page in a second organization. Such links, whether on duplicate or distinct pages, may produce undue support for the target page. Measuring the pervasiveness of this problem and the effectiveness of enforcing a "one person, one vote" rule is left for future work.

Conclusion

For truly popular pages, the efforts discussed in this paper are unlikely to affect results much, as most of their incoming links will be from unrelated external sites. However, the results of queries on less-popular topics may be undermined by savvy content creators. This paper has presented some of the issues surrounding the idea of eliminating nepotistic links from analysis, and demonstrated the potential for automatic recognition of such links with high accuracy.

Acknowledgments

This work benefits greatly from the systems, work, and discussions of the other members of the DiscoWeb research group: Apostolos Gerasoulis, Wei Wang, Hyun-ju Seo, Konstantinos Kleisouris, and Junyu Tian, and is supported in part by ARPA under Order Number F425 (via ONR contract N6600197C8534).

References

Ask Jeeves, Inc. 2000. Direct Hit home page. http://www.directhit.com/.

Bharat, K., and Broder, A. 1999. Mirror, mirror on the web: a study of host pairs with replicated content. In *Proceedings of the Eighth International World Wide Web Conference*.

Bharat, K., and Henzinger, M. R. 1998. Improved Algorithms for Topic Distillation in Hyperlinked Environments. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, 104–111.

Bharat, K.; Broder, A.; Dean, J.; and Henzinger, M. 1999. A comparison of techniques to find mirrored hosts on the WWW. In *Proceedings of the ACM Digital Library Workshop on Organizing Web Space (WOWS)*.

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*.

C. & G. Merriam Co. 1913. Webster's revised unabridged dictionary. Online at http://www.dictionary.com/.

Chakrabarti, S.; Dom, B. E.; Gibson, D.; Kumar, R.; Raghavan, P.; Rajagopalan, S.; and Tomkins, A. 1998a. Experiments in Topic Distillation. In *ACM SIGIR Workshop on Hypertext Information Retrieval on the Web*.

Chakrabarti, S.; Dom, B. E.; Raghavan, P.; Rajagopalan, S.; Gibson, D.; and Kleinberg, J. M. 1998b. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proceedings of the Seventh International World Wide Web Conference*.

Chakrabarti, S.; Dom, B. E.; Gibson, D.; Kleinberg, J. M.; Kumar, S. R.; Raghavan, P.; Rajagopalan, S.; and Tomkins, A. 1999a. Hypersearching the Web. *Scientific American*.

Chakrabarti, S.; Dom, B. E.; Gibson, D.; Kleinberg, J. M.; Kumar, S. R.; Raghavan, P.; Rajagopalan, S.; and Tomkins, A. 1999b. Mining the web's link structure. *IEEE Computer* 60–67.

Chakrabarti, S.; Dom, B. E.; and Indyk, P. 1998. Enhanced hypertext categorization using hyperlinks. In *Proceedings of ACM SIGMOD*.

Chakrabarti, S.; van den Berg, M.; and Dom, B. E. 1999. Focused crawling: a new approach to topic-specific web resource discovery. In *Proceedings of the Eighth International World Wide Web Conference*.

Davison, B. D.; Gerasoulis, A.; Kleisouris, K.; Lu, Y.; Seo, H.; Wang, W.; and Wu, B. 1999. DiscoWeb: Applying Link Analysis to Web Search. In *Poster proceedings of the Eighth International World Wide Web Conference*, 148–149.

Dean, J., and Henzinger, M. R. 1999. Finding related pages in the world wide web. In *Proceedings of the Eighth International World Wide Web Conference*, 389–401.

Gibson, D.; Kleinberg, J. M.; and Raghavan, P. 1998a. Inferring web communities from link topology. In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia (Hypertext'98)*. Expanded version at http://www.cs.cornell.edu/home/kleinber/.

Gibson, D.; Kleinberg, J. M.; and Raghavan, P. 1998b. Structural Analysis of the World Wide Web. In *World Wide Web Consortium Workshop on Web Characterization*. Invited position paper.

Google Inc. 2000. Google home page. http://www.google.com/.

Henzinger, M. R.; Heydon, A.; Mitzenmacher, M.; and Najork, M. 1999. Measuring index quality using random walks on the web. In *Proceedings of the Eighth International World Wide Web Conference*, 213–225.

IBM Almaden Research Center. 2000. The CLEVER Project. Home page: http://www.almaden.ibm.com/cs/k53/clever.html.

Kleinberg, J. M.; Kumar, S.; Raghavan, P.; Rajagopalan, S.; and Tomkins, A. 1999. The web as a graph: Measurements, models and methods. In *Proceedings of the International Conference on Combinatorics and Computing*. Invited survey.

Kleinberg, J. M. 1998. Authoritative sources in a hyperlinked environment. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA-98)*, 668–677. Expanded version at http://www.cs.cornell.edu/home/kleinber/.

Lycos, Inc. 2000. Lycos: Your personal internet guide. http://www.lycos.com/.

Microsoft Corporation. 2000. MSN Search home page. http://search.msn.com/.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1998. The PageRank citation ranking: Bringing order to the web. Unpublished draft available from http://www-db.stanford.edu/~backrub/pageranksub.ps.

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

Shivakumar, N., and Garcia-Molina, H. 1998. Finding near-replicas of documents on the web. In *Proceedings of Workshop on Web Databases (WebDB'98)*. Held in conjuntion with EDBT'98.

Sullivan, D. 2000. Search Engine Watch. http://www.searchenginewatch.com/.

Wired Digital Inc. 2000. HotBot home page. http://hotbot.lycos.com/.