Web Information Retrieval - an Algorithmic Perspective

Monika Henzinger[1]

Google, Inc., Mountain View, CA 94043 USA,
monika@google.com,
WWW home page: http://www.henzinger.com/monika/index.html

Abstract. In this paper we survey algorithmic aspects of Web information retrieval. As an example, we discuss ranking of search engine results using connectivity analysis.

1 Introduction

In December 1999 the World Wide Web was estimated to consist of at least one billion pages, up from at least 800 million in February 1999 [28]. Not surprisingly finding information in this large set of pages is difficult and many Web users turn to Web search engines for help. An estimated 150 million queries are currently asked to Web search engines per day – with mixed success.

In this paper we discuss algorithmic aspects of Web information retrieval and present an algorithm due to Brin and Page [6] that is very successful in distinguishing high-quality from low-quality Web pages, thereby improving the quality of query results significantly. It is currently used by the search engine Google[1].

This paper is loosely based on part of a tutorial talk that we presented with Andrei Broder at the 39th Annual Symposium on Foundations of Computer Science (FOCS 98) in Palo Alto, California. See [8] for write-up of the remaining parts of this tutorial.

We will use the terms *page* and *document* interchangeably.

2 Algorithmic Aspects of Web Information Retrieval

The goal of general purpose search engines is to index a sizeable portion of the Web, independently of topic and domain. Each such engine consists of three major components:

- A *crawler* (also called *spider* or *robot*) collects documents by recursively fetching links from a set of starting pages. Each crawler has different policies with respect to which links are followed, how deeply various sites are explored, etc. Thus, the pages indexed by various search engines differ considerably [27, 2].

[1] http://www.google.com/

- The *indexer* processes the pages collected by the crawler. First it decides which of them to index. For example, it might discard duplicate documents. Then it builds various data structures representing the pages. Most search engines build some variant of an inverted index data structure (see below). However, the details of the representation differ among the major search engines. For example, they have different policies with respect to which words are indexed, capitalization stemming, whether locations within documents are stored, etc. The indexer might also build additional data structures, like a repository to store the original pages, a Web graph representation to store the hyperlinks, a related-pages finder to store related pages, etc. As a result, the query capabilities and features of the result pages of various engines vary considerably.
- The *query processor* processes user queries and returns matching answers, in an order determined by a *ranking* algorithm. It transforms the input into a standard format (e.g. to lower-case terms), uses the index to find the matching documents, and orders (*ranks*) them.

Algorithmic issues arise in each part. We discuss some of them in the following.

2.1 Crawler

The crawler needs to decide which pages to crawl. One possible implementation is to assign to each page a priority score indicating its crawling importance. and to maintain all pages in a priority queue ordered by priority score. This priority score can be the PageRank score (defined in the next section) of the page [12] if the goal is to maximize the quality of the pages in the index. Alternatively, the score can be a criterion whose goal is to maximize the freshness of the pages in the index [14, 11].

The crawler also has to consider various load-balancing issues. It should not overload any of the servers that it crawls and it is also limited by its own bandwidth and internal processing capabilities. An interesting research topic is to design a crawling strategy that maximizes both quality and freshness and respects the above load-balancing issues.

2.2 Indexer

The indexer builds all the data structures needed at query time. These include the *inverted index*, a *URL-database*, and potentially a *graph representation*, a *document repository*, a *related-pages finder*, and further data structures.

The inverted index contains for each word a list of all documents containing the word, potentially together with the position of the word in the document. This list is sorted lexicographically according to the (document-id, position in the document)-pair. (See [1] for a detailed description of this data structure.)

To save space, documents are represented by document-ids in the index and the other data structures. When results are displayed, these document-ids need to be converted back to the URL. This is done using the URL-database.

The graph representation keeps for each document all the documents pointing to it and all the documents it points to. (See [4] for a potential implementation.)

The document repository stores for each document-id the original document.

The related-pages finder stores for each document-id all document-ids of pages that are related to the document. A related page is a page that addresses the same topic as the original page, but is not necessarily semantically identical. For example, given `www.nytimes.com` other newspapers and news organizations on the Web would be related pages. See [15] for algorithms that find related pages.

Of course all or some of the latter four data structures can be combined.

Before building the data structures the indexer needs to determine which pages to index. For this, it can assign a numerical score to each page and then index a certain number of top-ranked pages. For example this score can be 0 for all but one of a set of duplicate pages. The score might also try to measure the query-independent quality of a page. This can for example be done by the PageRank measure (see below).

An interesting algorithmic question for each of these data structures is how to compress the space as much as possible without affecting the average look-up time. Furthermore, the data structure should remain space-efficient when insertions and deletions of documents are allowed.

2.3 Query Processor

The main challenge of the query processor is to rank the documents matching the query by decreasing value for the user. For this purpose again a numerical score is assigned to each document and the documents are output in decreasing order of the score.

This score is usually a combination of query-independent and query-dependent criteria. A *query-independent* criterion judges the document regardless of the actual query. Typical examples are the length, the vocabulary, publication data (like the site to which it belongs, the date of the last change, etc.), and various connectivity-based techniques like the number of hyperlinks pointing to a page or the PageRank score. A *query-dependent* criterion is a score which is determined only with respect to a particular query. Typical examples are the cosine measure of similarity used in the vector space model [34], query-dependent connectivity-based techniques [25], and statistics on which answers previous users selected for the same query.

The algorithmically most interesting of these techniques are query-independent and query-dependent connectivity-based techniques. We will describe the query-independent approach below. The assumption behind the connectivity-based ranking techniques is that a link from page A to page B means that the author of page A recommends page B. Of course this assumption does not always hold. For example, a hyperlink cannot be considered a recommendation if page A and B have the same author or if the hyperlink was generated for example by a Web-authoring tool.

The idea of studying “referrals” is not new. There is a subfield of classical information retrieval, called bibliometrics, where citations were analyzed. See, e.g., [23, 17, 37, 18]. The field of sociometry developed algorithms [24, 30] very similar to the connectivity-based ranking techniques described in [6, 25]. Furthermore, a large amount of Web-related research exploits the hyperlink structure of the Web.

A Graph Representation for the Web The Web can be represented as a graph in many different ways. Connectivity-based ranking techniques usually assume the most straightforward representation: The graph contains a node for each page u and there exists a directed edge (u, v) if and only if page u contains a hyperlink to page v.

Query-independent connectivity-based ranking The assumption of connectivity based techniques immediately leads to a simple query-independent criterion: The larger the number of hyperlinks pointing to a page the better the page [9]. The main drawback of this approach is that each link is equally important. It cannot distinguish between the quality of a page pointed to by a number of low-quality pages and the quality of a page that gets pointed to by the same number of high-quality pages. Obviously it is therefore easy to make a page appear to be high-quality – just create many other pages that point to it.

To remedy this problem, Brin and Page [6, 31] invented the PageRank measure. The PageRank of a page is computed by weighting each hyperlink proportionally to the quality of the page containing the hyperlink. To determine the quality of a referring page, they use its PageRank recursively. This leads to the following definition of the PageRank $R(p)$ of a page p:

$$R(p) = \epsilon/n + (1 - \epsilon) \cdot \sum_{(q,p) \text{ exists}} R(q)/outdegree(q),$$

where

- ϵ is a dampening factor usually set between 0.1 and 0.2;
- n is the number of pages on the Web; and
- $outdegree(q)$ is the number of hyperlinks on page q.

Alternatively, the PageRank can be defined to be the stationary distribution of the following infinite random walk $p_1, p_2, p_3, \ldots$, where each p_i is a node in the graph: Each node is equally likely to be node p_1. To determine node p_{i+1} a biased coin is flipped: With probability ϵ node p_{i+1} is chosen uniformly at random from all nodes in the graph, with probability $1-\epsilon$ it is chosen uniformly at random from all nodes q such that edge (p_i, q) exists in the graph.

The PageRank measure works very well in distinguishing high-quality Web pages from low-quality Web pages and is used in the Google search engine[2].

[2] `http://www.google.com/`

Recent work refined the PageRank criterion and sped up its computation, see e.g. [33, 19]. In [20, 21] PageRank-like random walks were performed to sample Web page almost according to the PageRank distribution and the uniformly distribution, respectively. The goal was to compute various statistics on the Web pages and to compare the quality, respectively the number, of the pages in the indices of various commercial search engines.

2.4 Other algorithmic work related to Web information retrieval

There are many other algorithmic challenges in Web information retrieval. We list several below, but this list is by no means complete.

Near-duplicates: To save space in the index search engines try to determine near-duplicate Web pages and completely or partially near-duplicate Web hosts (called *mirrors*). See [5, 7, 35] for algorithms for the near-duplicates problem and see [3, 13] for algorithms for the mirrored host problem.

Clustering: Clustering query results has lead to an interesting application of suffix trees [38].

Web page categorization: There are various hierarchical directories of the Web, for example the open directory hierarchy[3]. They are usually constructed by hand. Automatically categorizing a web, i.e. placing it at a node(s) in the hierarchy to which it belongs, is a challenging problem. There has been a lot of work on text-only methods. See [10] for a first step towards a text and link-based approach.

Dynamically generated Web content: Trying to "learn" to crawl dynamically generated Web pages is an interesting topic of future research. A first step in this direction is taken by [16].

Web graph characterization: Characterizing the Web graph has led to a sequence of studies that are algorithmically challenging because of the pure magnitude of the data to be analyzed. See [26] for a survey.

Web user behavior: Mining user query logs shows that web users exhibit a different query behavior than users of classical information retrieval systems. An analysis of different query logs is given in [22, 36].

Modeling: Different users asking the same query can widely disagree on the relevance of the answers. Thus, it is not possible to prove that certain ranking algorithms return relevant answers at the top. However, there has been some recent work on trying to find appropriate models for clustering problems in information retrieval and to use them to explain why certain algorithms work well in practise. See, e.g. [32]. This is certainly an interesting area of future research.

3 Conclusions

There are many interesting algorithmic questions related to Web information retrieval. One challenge is to find algorithms with good performance. The per-

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

formance of these algorithms is usually validated by experimentation. The other challenge is to theoretically model information retrieval problems in order to explain why certain algorithms perform well.

References

1. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley, 1999.
2. K. Bharat and A. Z. Broder. A technique for measuring the relative size and overlap of public Web search engines. In *Proceedings of the Seventh International World Wide Web Conference* 1998, pages 379–388.
3. K. Bharat, A. Z. Broder, J. Dean, and M. Henzinger. A comparison of Techniques to Find Mirrored Hosts on the World Wide Web. To appear in the *Journal of the American Society for Information Science.*
4. K. Bharat, A. Z. Broder, M. Henzinger, P. Kumar, and S. Venkatasubramanian. The connectivity server: Fast access to linkage information on the Web. In *Proceedings of the Seventh International World Wide Web Conference* 1998, pages 469–477.
5. S. Brin, J. Davis, and H. García-Molina. Copy detection mechanisms for digital documents. In M. J. Carey and D. A. Schneider, editors, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 398–409, San Jose, California, May 1995.
6. S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the Seventh International World Wide Web Conference* 1998, pages 107–117.
7. A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the Web. In *Proceedings of the Sixth International World Wide Web Conference* 1997, pages 391–404.
8. A. Z. Broder and M. R. Henzinger. Algorithmic Aspects of Information Retrieval on the Web. In *Handbook of Massive Data Sets*. J. Abello, P.M. Pardalos, M.G.C. Resende (eds.), Kluwer Academic Publishers, Boston, forthcoming.
9. J. Carriere and R. Kazman. Webquery: Searching and visualizing the web through connectivity. In *Proceedings of the Sixth International World Wide Web Conference* 1997, pages 701–711.
10. S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1998, pages 307–318.
11. J. Cho and H. García-Molina. The Evolution of the Web and Implications for an incremental Crawler. *Proceedings of the 26th International Conference on Very Large Databases* (VLDB), 2000.
12. J. Cho, H. García-Molina, and L. Page. Efficient crawling through URL ordering. In *Proceedings of the Seventh International World Wide Web Conference* 1998, pages 161–172.
13. J. Cho, N. Shivakumar, and H. García-Molina. Finding replicated Web collections. *Proceedings of the 2000 ACM International Conference on Management of Data* (SIGMOD), 2000.
14. E. G. Coffman, Z. Liu, and R. R. Weber. Optimal robot scheduling for Web search engines. Technical Report 3317, INRIA, Dec. 1997.
15. J. Dean and M. R. Henzinger. Finding Related Web Pages in the World Wide Web. In *Proceedings of the 8th International World Wide Web Conference* 1998, pages 389–401.

16. R. B. Doorenbos, O. Etzioni, and D. S. Weld. A scalable comparison-shopping agent for the World-Wide Web. In W. L. Johnson and B. Hayes-Roth, editors, *Proceedings of the 1st International Conference on Autonomous Agents*, pages 39–48, New York, Feb. 1997. ACM Press.
17. E. Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178, 1972.
18. E. Garfield. *Citation Indexing.* ISI Press, 1979.
19. T. Haveliwala. Efficient Computation of PageRank. Technical Report 1999-31, Stanford University, 1999.
20. M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. Measuring Search Engine Quality using Random Walks on the Web. In *Proceedings of the 8th International World Wide Web Conference* 1999, pages 213–225.
21. M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. On near-uniform URL sampling. In *Proceedings of the Ninth International World Wide Web Conference* 2000, pages 295–308.
22. B. J. Jansen, A. Spin, J. Bateman, and T. Saracevic. Real Life Information Retrieval: A Study of User Queries on the Web. *SIGIR FORUM,* 32 (1):5–17, 1998.
23. M. M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14, 1963.
24. L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39-43, March 1953.
25. J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, January 1998.
26. J. Kleinberg, S.R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The Web as a graph: Measurements, models and methods. Invited survey at the *International Conference on Combinatorics and Computing*, 1999.
27. S. Lawrence and C. L. Giles. Searching the World Wide Web. *Science*, 280(5360):98, 1998.
28. S. Lawrence and C. L. Giles. Accessibility of Information on the Web. *Nature*, 400(6740):107–109, 1999.
29. Dharmendra S. Modha and W. Scott Spangler. Clustering Hypertext with Applications to Web Searching. *Proceedings of the ACM Hypertext 2000 Conference, San Antonio, TX*, 2000. Also appears as IBM Research Report RJ 10160 (95035), October 1999.
30. M. S. Mizruchi, P. Mariolis, M. Schwartz, and B. Mintz. Techniques for disaggregating centrality scores in social networks. In N. B. Tuma, editor, *Sociological Methodology*, pages 26-48. Jossey-Bass, San Francisco, 1986.
31. L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. *Stanford Digital Library Technologies*, Working Paper 1999-0120, 1998.
32. C. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent Semantic Indexing: A Probabilistic Analysis. In *Proceedings of the 17th ACM Symposium on the Principles of Database Systems*, 1998.
33. D. Rafiei, and A. Mendelzon. What is this page known for? Computing Web page reputations. In *Proceedings of the Ninth International World Wide Web Conference* 2000, pages 823–836.
34. G. Salton. *The SMART System – Experiments in Automatic Document Processing.* Prentice Hall.
35. N. Shivakumar and H. García-Molina. Finding near-replicas of documents on the Web. In *Proceedings of Workshop on Web Databases (WebDB'98)*, March 1998.

36. C. Silverstein, M. Henzinger, H. Marais, and M. Moricz. Analysis of a Very Large AltaVista Query Log. Technical Note 1998-014, Compaq Systems Research Center, 1998. To appear in *SIGIR FORUM.*
37. H. Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *J. Amer. Soc. Info. Sci.*, 24, 1973.
38. O. Zamir and O. Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 46–54.