

Recent results in automatic Web resource discovery

Soumen Chakrabarti
Department of Computer Science and Engineering
Indian Institute of Technology Bombay

Final version started August 16, 1999
Last modified December 16, 1999

1 Introduction

Classical information retrieval (IR) is concerned with indexing a collection of documents and answering queries by returning a ranked list of relevant documents [14, 21, 24]. With the growth of the web, the problems of ambiguity, context sensitivity, synonymy (two terms with the same meaning) and polysemy (one term with different meanings) that are inherent in natural languages, together with the *abundance* of web pages related to prominent topics, have exacerbated the difficulty of fulfilling the user's information need.

Most search sites have added directory-based topic browsing. The web is organized as a tree of topics, similar to the Dewey decimal system, the Library of Congress catalog, or the US Patent and Trademarks Office subject codes. Tree nodes are maintained by paid ontologists and/or specialist volunteers, such as at Yahoo!, The Mining Co., WWW Virtual Library, and Open Directory Project. This strategy may be biased because of sparsity of experts; at any rate it is biased away from the most accomplished and busiest people.

1.1 The need for focused resource discovery

Directory sites are very popular. Because of human judgement, pages placed in a topic directory tend to be not only relevant, but also exemplary and influential. However, human judgement is slow, subjective and noisy. It is labor-intensive to keep these directories comprehensive. Consequently, there is a great need for automatic discovery, analysis and compilation of topical resources.

Standard crawlers and search engines do not provide adequate support for automatic resource discovery. In 1997, a study by Bharat and Bröder revealed that the best crawlers running on heavy-duty servers covered 35–40% of pages on the Web, which numbered over 340 million at that time [1]. In February 1999 Lawrence and Lee Giles [19] estimated that the coverage had dropped to only 18%, because the Web had grown to over 800 million pages.

Therefore, there is little hope for maintaining generic portals using crawlers and search engines. In any case, generic portals, with their one-size-fits-all philosophy, have limited application: they are good for preliminary exploration, but are rarely used regularly by experts. (Generic search portals are still very popular, but part of the popularity may be owing to free email, chat, and e-commerce.) Seasoned users who use the Web for serious research increasingly prefer deep, topic-specific *portholes* to shallow, generic *portals* [15, 22, 20].

In this survey I will trace the development of a new suite of hypertext resource discovery and analysis tools which work together as a *focused crawler* to build a topic-specific portal. There are two main modules in a focused crawler: a *classifier* and a *distiller*. The classifier

learns to recognize hypertext pertaining to the user's topic(s) of interest. The distiller identifies compilations of relevant hyperlinks and exploits them to rapidly expand the crawl without crawling irrelevant documents. Both the classifier and the distiller exploit the fact the hyperlinks on the Web are a *social* phenomenon.

1.2 Social network analysis

The web is an example of a *social network*. Social networks have been extensively researched [26]. Social networks are formed between academics by co-authoring, advising, serving on committees; between movie personnel by directing and acting; between musicians, football stars, friends and relatives; between people by making phone calls and transmitting infection; between papers through citation, and between web pages by hyperlinking to other web pages.

Social network theory is concerned with properties related to connectivity and distances in graphs, with diverse applications like epidemiology, espionage, citation indexing, etc. In the first two examples, one might be interested in identifying a few nodes to be removed to significantly increase average path length between pairs of nodes. In citation analysis, one may wish to identify influential or central papers; this turns out to be quite symmetric to finding good survey papers; this symmetry has been explored by Mizruchi and others [23]. IR literature includes insightful studies of citation, co-citation, and influence of academic publication [18].

2 Topic distillation

Starting in 1996, a series of applications of social network analysis were made to the web graph, with the purpose of identifying the most authoritative pages related to a user query. These are surveyed briefly here; a more detailed analysis appears elsewhere [9].

2.1 A collection of distillation techniques

Google: If one wanders on the web for infinite time, following a random link out of each page, then different pages will be visited at different rates; popular pages with many in-links will tend to be visited more often. PageRank and Google¹, invented by Brin and Page [4], crawl the web and simulate such a random walk on the web graph in order to estimate the visitation rate, which is used as a score of popularity. Given a keyword query, matching documents are ordered by this score. Note that the popularity score is precomputed independent of the query, hence Google can potentially be as fast as any relevance-ranking search engine.

HITS: Hyperlink induced topic search (HITS) [16] is slightly different: it does not crawl or pre-process the web, but depends on a search engine. A query to HITS is forwarded to a search engine such as Alta Vista, which retrieves a subgraph of the web whose nodes (pages) match the query. Pages citing or cited by these pages are also included. This expanded graph is analyzed for popular nodes using a procedure similar to Google, the difference being that not one, but two scores emerge: the measure of a page being an authority, and the measure of a page being a *hub* (a compilation of links to authorities, or a "survey paper" in bibliometric terms). Because of the query-dependent graph construction, HITS is slower than Google. A variant of this technique has been used by Dean and Henzinger to find similar pages on the Web using link-based analysis alone [12]. They improve speed by fetching the Web graph from a *connectivity server* which has pre-crawled substantial portions of the Web [2].

¹Google! search is online at <http://google.com>.

ARC and CLEVER: HITS's graph expansion sometimes leads to topic *contamination* or *drift*. E.g., the community of *movie awards* pages on the web is closely knit with highly cited (and to some extent relevant) home pages of movie *companies*. Although *movie awards* is a finer topic than *movies*, the top movie companies emerge as the victors upon running HITS. This is partly because in HITS (and Google) all edges in the graph have the same importance. Contamination can be reduced by recognizing that hyperlinks that contain *award* or *awards* near the anchor text are more relevant for this query than other edges. Such heuristic modification of edge weights significantly improve the quality of query results. In user studies, the results compared favorably with lists compiled by humans, such as Yahoo! and Infoseek [7].

Outlier filtering: Bharat and Henzinger have invented another way to integrate textual content and thereby avoid contamination of the graph to be distilled. They model each page according to the "vector space" model [24]. During the graph expansion step, unlike HITS, they do not include all nodes at distance one from the preliminary query result. Instead they prune the graph expansion at nodes whose corresponding term vectors are outliers with respect to the set of vectors corresponding to documents directly retrieved from the search engine [3]. In the example above, one would hope that the response to the query `movie award` from the initial keyword search would contain a majority of pages related to awards and not companies; thus the distribution of keywords on these pages will enable the Bharat and Henzinger algorithm to effectively prune away as outliers nodes in the neighborhood that are about movie companies. Apart from speeding up their system by using the Connectivity Server [2], they describe several heuristics that cut down the query time substantially.

It is possible to fabricate queries that demonstrate the strengths and weaknesses of each of these systems. ARC, CLEVER, and Outlier Filtering have been shown to be better (as judged by testers) than HITS. There has not been a systematic comparison between Google and the HITS family. This would be of great interest given the basic difference in graph construction and consequent greater speed of Google.

2.2 Topic distillation \neq resource discovery

All the distillation systems depend on large, comprehensive web crawls and indices. We will argue that topic distillation is therefore a *post-processing* operation and therefore not suited for resource discovery.

Topic distillation systems work well for well-connected communities concerning broad topics. It is thus tempting to use a topic distillation system to generate a web taxonomy in the following trivial way: with each node in the taxonomy, associate a keyword query that describes the topic, and run a distillation program. While the simplicity is appealing, it is not easy to make this succeed.

First, constructing the query involves trial and error, and a fair amount of thought. As an example, none of the homepages of AT&T, MCI and Sprint appear in the top 10 responses from Alta Vista to the query `north american telecommunication companies`. As another example, the query `"power suppl*" "switch* mode" smps -multiprocessor* "uninterrupt* power suppl*" ups -parcel` was needed to populate the Yahoo! node `/Business&Economy /Companies /Electronics /PowerSupplies` with acceptable quality. (SMPS stands for Switch Mode Power Supply, but also matches pages containing SMPs, or Symmetric Multi-Processors! Similar comments apply to UPS and parcel.) In a study with 966 nodes from Yahoo! [9, 10], in order to match (in the opinion of blind testers) the quality of Yahoo!, queries had to be tuned by hand until the average query had 7.03 terms and 4.34 operators ("+-*"), in sharp contrast to the average Alta Vista query having 2.35 words and only 0.41 operators [25]. These queries are

not a one-time effort, because inclusion of additional topic vocabulary, which may not be known a priori, improved the results. E.g., good results were obtained for “European Airlines” using the query `+lufthansa +iberia +klm` (the fourth response from Alta Vista was itself a hub).

The second issue arises from the aforesaid susceptibility to contamination from popular but irrelevant nodes. The contamination problem can be addressed in a few ad-hoc ways: stop-sites, term weighting and edge weighting. Stop-sites are nodes forcibly removed from the graph before the iterations. Query terms can be assigned weights by humans to be used for better ranking. The weights of links incident with example pages, or lexically close to links to example pages, can be increased artificially. These fixes are ad-hoc; there are no principles for setting edge weights guided by a precise model of hyperlinkage.

3 Hypertext classification: learning from example

In the experiments described above, the most successful way to combat contamination has been the use of examples. In 86% of the test cases, specifying an example page and forcibly including it into the graph to be distilled (even if it did not match the keyword search) improved the results as perceived by testers. For example, placing the exemplary nodes `http://www.att.com` and `http://www.sprint.com` (and their distance-1 neighbors) in the graph to be distilled for “north American telecommunication companies” enhances the quality of the answer remarkably.

However, an example page offers much more than just a forced node in the distilled graph. It has textual content, and is linked to neighbors with more textual content. In fact, extensive literature in relevance feedback, automatic feature selection and classification suggests that given examples, it is not even necessary to provide a keyword query—the learning process will implicitly recognize the important terms and compute the decision boundaries that can be used to determine whether a given web page is relevant to a topic.

These decision boundaries can be derived implicitly and without any human effort, given the example documents. Most types of classifiers, such as nearest neighbor (NN) [5], bayesian [6, 17], support vector [13], are capable of more reliable discrimination than keyword search, even if the keyword search includes boolean constructs. (Hand-constructed boolean search induces hard and brittle rules that match known examples well, but do not generalize well to unknown documents, a phenomenon called *overfitting* in statistical Machine Learning. Effective learning algorithms protect against overfitting by design.)

In the hypertext domain, it is extremely important to build models and classifiers that take link-based features into account. The topic of a page influences its text and the topics of pages in its neighborhood. This is a bibliometric phenomenon which couples content and hyperlinks, and is more general than the simple linear endorsement model used in topic distillation. The phenomenon can be characterized using robust statistical models such as *Markov Random Fields* (MRF’s). MRF’s can capture and exploit citation behavior such as “documents about cardiovascular health may refer to documents about swimming and cream-cakes, but is unlikely to cite a document about repairing washing machines.”

Thus, knowledge of the topics of documents in the link vicinity of a test document gives valuable clues as to the topic of the test document. Note that this characterization of topical influence is circular, but the circularity can be resolved by an *iterative relaxation* algorithm such as HyperClass [8]. Classification error reduced from 36% to 21% in an experiment with US Patents. With Yahoo!, a more dramatic reduction from 69% to 20% was observed.

4 Putting it together for resource discovery

Provided a crawler is started off from connected examples of topics, it can be guided and scheduled by HyperClass. The goal of a focused crawler is to selectively seek out pages that are relevant to a pre-defined set of topics. The topics are specified not using keywords, but using exemplary documents that are analyzed by HyperClass. The focused crawler analyzes its crawl boundary to find the links that are likely to be most relevant for the crawl. It avoids irrelevant regions of the web. This leads to significant savings in hardware and network resources, and helps keep the crawl more up-to-date.

Extensive focused-crawling experiments have been performed using several topics at different levels of specificity [11]. The system acquires relevant pages steadily while standard crawling quickly loses its way, even though they are started from the same root set. Focused crawling is robust against large perturbations in the starting set of URLs. It discovers largely overlapping sets of resources in spite of these perturbations. In contrast with topic distillation systems, it is also capable of exploring out and discovering valuable resources that are dozens of links away from the start set, while carefully pruning the millions of pages that may lie within this same radius. These results suggest that focused crawling is very effective for building high-quality collections of web documents on specific topics, using modest desktop hardware.

5 Conclusion

In this survey I have emphasized the importance of scalable automatic resource discovery on the ever-expanding Web, argued that common search engines are not adequate to achieve such resource discovery, and described a recently invented *focused crawling* system that can build a collection of resources focused on specific topics by learning from example. In ongoing work² we are exploring how to derive the examples automatically from the browsing patterns of a user community, and to personalize the outcome of the focused crawl for these users.

References

- [1] K. Bharat and A. Broder. A technique for measuring the relative size and overlap of public web search engines. In *7th World-Wide Web Conference (WWW7)*, 1998. Online at <http://www7.scu.edu.au/programme/fullpapers/1937/com1937.htm>; also see an update at <http://www.research.digital.com/SRC/whatsnew/sem.html>.
- [2] K. Bharat, A. Bröder, M. Henzinger, P. Kumar, and S. Venkatasubramanian. The Connectivity Server: Fast access to linkage information on the Web. In *7th World Wide Web Conference*, Brisbane, Australia, 1998. Online at http://www.research.digital.com/SRC/personal/Andrei_Broder/cserv/386.html and <http://decweb.ethz.ch/WWW7/1938/com1938.htm>.
- [3] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104–111, Aug. 1998. Online at <ftp://ftp.digital.com/pub/DEC/SRC/publications/monika/sigir98.pdf>.

²The Focus project home page is at <http://www.cs.berkeley.edu/~soumen/focus/>.

- [4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th World-Wide Web Conference (WWW7)*, 1998. Online at <http://decweb.ethz.ch/WWW7/1921/com1921.htm>.
- [5] E. Brown. Execution performance issues in full-text information retrieval. Technical Report TR95-81, University of Massachusetts, Amherst, 1995. Online at <ftp://ftp.cs.umass.edu/pub/techrept/techreport/1995/UM-CS-1995-081.ps>.
- [6] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. *VLDB Journal*, Aug. 1998. Invited paper, online at http://www.cs.berkeley.edu/~soumen/VLDB54_3.PDF.
- [7] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic resource compilation by analyzing hyperlink structure and associated text. In *7th World-wide web conference (WWW7)*, 1998. Online at <http://www7.scu.edu.au/programme/fullpapers/1898/com1898.html>.
- [8] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD Conference*. ACM, 1998. Online at <http://www.cs.berkeley.edu/~soumen/sigmod98.ps>.
- [9] S. Chakrabarti, B. E. Dom, S. Ravi Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg. Mining the Web's link structure. *IEEE Computer*, 32(8):60–67, Aug. 1999. Feature article.
- [10] S. Chakrabarti, M. van den Berg, and B. Dom. Distributed hypertext resource discovery through examples. In *VLDB*, Edinburgh, Scotland, Sept. 1999.
- [11] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31:1623–1640, 1999. First appeared in the 8th International World Wide Web Conference, Toronto, May 1999. Available online at <http://www8.org/w8-papers/5a-search-query/crawling/index.html>.
- [12] J. Dean and M. R. Henzinger. Finding related pages in the world wide web. In *8th World Wide Web Conference*, Toronto, May 1999.
- [13] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *7th Conference on Information and Knowledge Management*, 1998. Online at <http://www.research.microsoft.com/~jplatt/cikm98.pdf>.
- [14] W. B. Frakes and R. Baeza-Yates. *Information retrieval: Data structures and algorithms*. Prentice-Hall, 1992.
- [15] D. Gillmor. Small portals prove that size matters. San Jose Mercury News, Dec. 1998. Online at <http://www.sjmercury.com/columnists/gillmor/docs/dg120698.htm> and <http://www.cs.berkeley.edu/~soumen/focus/DanGillmor19981206.htm>.
- [16] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *ACM-SIAM Symposium on Discrete Algorithms*, 1998. Online at <http://www.cs.cornell.edu/home/kleinber/auth.ps>.

- [17] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *International Conference on Machine Learning*, volume 14. Morgan-Kaufmann, July 1997. Online at <http://robotics.stanford.edu/users/sahami/papers-dir/ml97-hier.ps>.
- [18] R. Larson. Bibliometrics of the world wide web: An exploratory analysis of the intellectual structure of cyberspace. In *Annual Meeting of the American Society for Information Science*, 1996. Online at <http://sherlock.berkeley.edu/asis96/asis96.html>.
- [19] S. Lawrence and C. Lee Giles. Accessibility of information on the web. *Nature*, 400:107–109, July 1999.
- [20] D. Lidsky and N. Sirapyan. Find it on the web. ZDNet, Jan. 1999. Online at <http://www.zdnet.com/products/stories/reviews/0,4161,367982,00.html> and http://www.cs.berkeley.edu/~soumen/focus/Lidsky_0_4161_367982_00.html.
- [21] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. Miller, and R. Teng. Five papers on WordNet. Online at <ftp://ftp.cogsci.princeton.edu/pub/wordnet/5papers.pdf>, Princeton University, Aug. 1993.
- [22] M. Mirapaul. Well-read on the web. *The New York Times*, Dec. 1998. Online at <http://www.nytimes.com/library/tech/98/12/circuits/articles/24port.html> and <http://www.cs.berkeley.edu/~soumen/focus/MatthewMirapaul19981224.html>.
- [23] 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.
- [24] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
- [25] C. Silverstein, M. Henzinger, H. Marais, and M. Moricz. Analysis of a very large AltaVista query log. Technical Report 1998-014, COMPAQ System Research Center, Oct. 1998. Online at <http://gatekeeper.dec.com/pub/DEC/SRC/technical-notes/abstracts/src-tn-1998-014.html>.
- [26] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, 1994.