Search Technologies for the Internet
Monika Henzinger, *et al.*
Science **317**, 468 (2007);
DOI: 10.1126/science.1126557

Science (print ISSN 0036-8075; online ISSN 1095-9203) is published weekly, except the last week in December, by the American Association for the Advancement of Science, 1200 New York Avenue NW, Washington, DC 20005. Copyright 2007 by the American Association for the Advancement of Science; all rights reserved. The title *Science* is a registered trademark of AAAS.

Search Technologies for the Internet

Monika Henzinger

About 20% of the world's population uses the Web, and a large majority thereof uses Web search engines to find information. As a result, many Web researchers are devoting much effort to improving the speed and capability of search technology.

A Web search engine consists of two parts: an offline part that gathers Web pages and builds an internal representation of them called an (inverted) index, and an online part that serves user requests by finding all matching documents and ordering or ranking them with the goal of presenting the most relevant documents on top (Fig. 1). To this day, index comprehensiveness and good result ranking are the main challenges faced by Web search engines and are the areas in which Web search engines are competing most fiercely. This article describes these challenges and some solutions, concentrating on the information retrieval aspects of Web searching. It does not discuss the questions arising in the design of the infrastructure needed to support large-scale search engines [see, e.g., (*1*, *2*)].

The first Web search engines became available about 15 years ago. They indexed tens of millions of Web pages and served hundreds of thousands of searches per day. These seemed like large numbers at the time, but since then, Web search engines have had to increase their capacity enormously. Currently, they are indexing tens of billions of Web pages and serving hundreds of millions of Web searches per day. In addition, the quality of the search results has improved noticeably. The first Web search engines used text-only ranking algorithms that had been developed in the field of information retrieval during the preceding 30 years. However, these techniques were designed for searching document collections of well-written, homogeneous articles, such as newspaper archives, that are mostly searched by librarians and other search specialists. In this setting, the comprehensiveness of the results is as important as its relevance. On the Web, the pages are heterogeneous and of varying quality, and the majority of searches are performed by novices. The user looks frequently only at the top 10 results (*3*). Thus, for many queries, the relevance of the top results is more important than the comprehensiveness of the result set. Because the text-only techniques employed by the first search engines were not designed for this setting, the quality of the results was frequently poor.

A substantial improvement in the quality of Web search results was possible through the analysis of the hyperlink structure of the Web (*4*, *5*). Hyperlinks are navigation elements in Web pages. When clicked on, a hyperlink loads into the browser window a different part of the current Web page or a different Web page. Hyperlinks in Web pages serve a similar purpose as do references in scientific articles. In 1955, Garfield showed that an analysis of the structure of references can determine the importance of scientific articles and journals (*6*). A similar analysis of the hyperlink structure of the Web gives an estimate for the quality of Web pages. This analysis leads to a query-independent estimate of page quality. To deploy it in a ranking algorithm, it needs to be combined with query-specific signals, such as the frequency of the query terms on the Web page. Google was the first commercial Web search engine to use this kind of hyperlink analysis in its ranking through its PageRank measure (*4*). Mathematically speaking, the PageRank vector contains one entry per Web page and is the Eigenvector of a matrix derived from the hyperlink structure of the Web. If the matrix is seen as a linear transformation of vectors, then the Eigenvector is the vector whose direction is not changed by the transformation. Thus, the PageRank vector can be viewed as an inherent property of the whole Web structure. Informally speaking, hyperlinks are interpreted as recommendations, and PageRank tries to measure how highly recommended a page is. If many hyperlinks point to a page, its PageRank is large. If the pages containing these hyperlinks have high PageRank themselves, that is, are highly recommended, then the PageRank of the page increases even further.

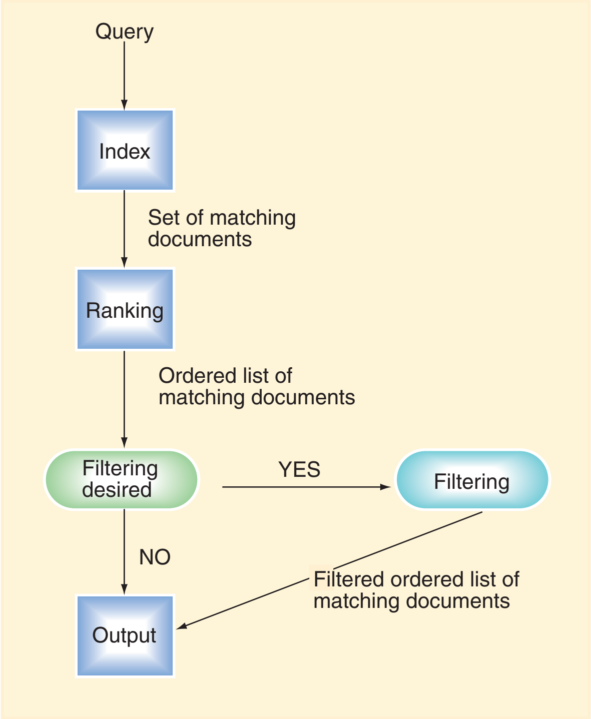

Query
Index
Set of matching documents
Ranking
Ordered list of matching documents
Filtering desired
YES
Filtering
NO
Filtered ordered list of matching documents
Output

Fig. 1. Steps of a query. The filter can be, for example, a filter for inappropriate content or a language filter.

Even though comprehensiveness and result quality of Web search engines have progressed steadily, there is still much room for improvement. Search engines cannot index all Web pages but only the pages that are publicly available and accessible without further "form-filling" actions, like filling in text in boxes or checking buttons on Web pages. By definition, Web pages that are not publicly available are not supposed to be available to the general public. Form filling, however, creates a challenge. Frequently, forms are the only way to access large amounts of information stored in online databases. It is conjectured that this information constitutes a large fraction of the "deep Web," which is the name given to the part of the Web that is not indexed by popular search engines (*7*).

There are many challenges that make ranking difficult. Some of the most important are (i) Many queries are short and underspecified. (ii) Synonyms and homonyms make it difficult to decide whether a page is relevant to a query or not. This classic problem of information retrieval is exacerbated on the Web by homonyms between languages. (iii) Ranking the most authoritative results first is made harder by authors who specifically design their Web pages so as to place them high for certain, mostly commercial searches. This is called search engine spam (not to be confused with e-mail spam). (iv) Users ask for additional features, such as filters for inappropriate content.

Here, I report on some of the ongoing research to address the above questions. I use the term "search engine" to denote a commercial Web search engine and the terms "Web page" or "page" to denote a document publicly accessible on the World Wide Web.

Comprehensiveness

The goal of search engines is to find the most relevant documents for user queries. Because of

Google Switzerland and Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland. E-mail: monika.henzinger@epfl.ch

the great variety in information needs, search engines must be very comprehensive. One way to achieve this is by indexing as many Web pages as possible. However, the larger the index, the higher is the cost per search for a search engine, because more machines are needed to store and search the index (*8*). Additionally, the more Web pages have already been indexed, the harder it becomes to find pages with new content, that is, content that is not contained in already indexed Web pages. Considering the diminishing returns and increasing cost of larger indices, search engines stop gathering pages after certain criteria regarding the size and coverage of various languages have been met.

A plethora of content is stored in databases rather than in typical Web pages. The pages as well as their URLs (*9*) are created in response to a user filling out a form on the Web. Because search engines are unable to emulate this behavior, such dynamically generated pages cannot be indexed. There has been some research on trying to make form-filling automatic (*10*, *11*), but the problem remains largely unsolved. On the other hand, if the search engine knew the URL, then it could request the page directly. Thus, a search engine simply needs a list of all URLs accessible at a site. Following this idea, Google has proposed an open protocol (*12*), that is, a format for Web sites to disclose a list of all URLs they want indexed by a search engine. This service is available for all Web sites, not only for Web databases. In exchange for disclosing the list, Google reports back to the site which URLs it could not access, along with query and user click statistics for the site. Yahoo and MSN, as well as organizations with large databases, such as Wikipedia and the *New York Times*, have already adopted this protocol.

Result Ranking

Together with comprehensiveness, the quality of the result ranking is crucial for the success of a search engine. One useful signal for ranking is anchor text. Anchor text is the text associated with a hyperlink, usually appearing in blue font. Clicking on it brings the user to the Web page of the associated URL, that is, the page to which the hyperlink points. For ranking purposes, many search engines treat the anchor text as if it were part of the text on the page that it points to. This is useful because anchor text often gives a concise description of the page and can thus match queries that also use a concise keyword description to retrieve the page. Additionally, the home pages of many companies consist of much graphics but few words, thus not giving a strong signal that the page is the official company home page. In such situations, anchor text can often be relied on to identify the homepage.

Handling short or underspecified queries. The average query length has not changed much over the years and is less than three terms. Depending on the type of query, short queries may cause a problem. Queries are roughly classified into these three types (*13*): (i) informational queries, whose goal is to obtain information regarding a topic of interest; (ii) navigational queries, whose goal is to find a specific Web page, such as the home page of a company; and (iii) transactional queries, whose goal is to perform a desired action, such as downloading a certain software package. For navigational and transactional queries, short queries are often sufficient. However, informational queries frequently need more information about the user's topic of interest. A recent study found that many informational queries are not specific enough; it showed that the users' information needs vary greatly even when they use the same query terms (*14*). For example, for the query "trailblazer" one user might want information about the car, whereas another user might want information about the basketball team.

To address this problem, either the user needs to be enticed to be more specific, for example, by refining the search, or user-specific information needs to be taken into account. On the Web, most users are reluctant to do additional work. Thus, the area of automatically exploiting user-specific information so as to personalize result rankings has received considerable attention.

To personalize a search, the search engine needs to know what the specific user is currently looking for (short-term interest). If this is not clearly expressed, then the general interests of the user (long-term interests) may be helpful. Thus, algorithms try to build a model of both the short-term and the long-term user interests. The model can either (i) suggest additional search terms or completely new queries to the user [see, e.g., (*15*)] or (ii) reorder the search results automatically. Recall that the ranking of a search engine usually depends on both query-dependent and query-independent signals. Hence, the reordering can personalize either query-dependent signals, for example, by automatically adding words suggested by the model to the query, or query-independent signals, for example, by using a personalized PageRank.

Data sources for the short-term model are queries issued by the user in the same session or the session history of other users with similar queries. Data sources for the long-term model are search-related user information, such as the user's query and browsing history, and search-independent user information, such as documents and e-mail that the user has read and personalized information that the user provided to the search service. For example, based on the knowledge that the user is a car enthusiast or based on the fact that his or her previous query was for "Chevrolet," the system could automatically add the word "car" to the query "trailblazer." The first studies employing automatically created short-term and long-term models to rerank search results with query-dependent signals found noticeable improvements in search quality (*16*–*18*). More research is under way to explore the full strength of this approach.

Personalizing query-independent signals, specifically PageRank, has also received much attention. This is challenging, because the PageRank computation is time and space intensive. It is time intensive because it requires the solution of a linear system with as many equations and variables as there are Web pages. It is space intensive because it requires storage of a PageRank score for every Web page. Thus, storing a personalized PageRank for every user would be very resource expensive. The current state of the art (*19*) in the personalization of PageRank allows the computation of about 100,000 topic-related PageRanks, which can be arbitrarily combined by a user. See (*20*, *21*) for reviews on the topic.

Handling synonyms and homonyms. Web search engine users have come to expect that their exact query terms appear in the documents of the result set. Thus, search engines are reluctant to return documents that contain synonyms of the query terms but are lacking one of the query terms. At best, they suggest alternate queries that contain synonyms of the original query terms.

An interesting question is what results should be returned in the top 10 for homonyms such as "jaguar." One of the top three dominant search services returns seven results on cars, one on the cat, one on a Macintosh OS X version, and one on a quantum chemistry software package called Jaguar. A second search engine returns four results on cars and four on the animal, one on a sports team, and one on a rock band. The third search service returns six results on cars and four on the animal. This points to a constant discussion in Web searching [see, e.g., (*22*)]: How much diversity should there be in search results? Automatically detecting whether more diversity is needed for a given query is still an open research question.

Some problems with homonyms are due to overlap of words or names of people with names of locations. This has led to interesting research with the goal of detecting the geographic context of a query. Such research addresses two issues: determining the geographic context for queries that do not contain a location but have a geographic context, such as "space needle," and detecting the lack of a geographic context in certain queries with location, such as "denzel washington." Simple lookups in geographic dictionaries, called gazetteers, would fail in both cases. The first problem can be addressed by first retrieving the body text of Web pages that have been clicked on by other users for the same query and/or the body text of the top search results; then using a gazetteer to extract all location names; and finally, based on the frequency and spread of these locations, determining a dominant location. For the query "space needle," "Seattle, Washington," would be by far the most frequent location, and the spread of the remaining locations would not show any particular patterns. Thus, "Seattle" would be selected as the dominant location. The second problem can be addressed in a similar way. The results for the query "New York Times," for example, would most likely contain the phrase "New York Times" more often than the phrase "New York" by itself, giving a strong signal that "New York Times" is an unbreakable phrase. A lookup in a gazetteer would then indicate that the query does not have a geographic context, because "New York Times" is not a location. With this approach, the geographic context of about 95% of queries can be detected with an accuracy of about 95% (*23*). The next step is to devise algorithms that exploit this information to improve search results.

Fighting search engine spam. Deciding what constitutes search engine spam is often difficult. Some results are obviously search engine spam, such as a page to purchase a quantum chemistry software package that is returned in the top 10 results for the query "jaguar." Others are less clear, for example, when the query "Hilton San Francisco" returns a page of a travel agency that is not affiliated with Hilton Hotels but which allows users to book a hotel room in the San Francisco Hilton.

Search engine spam usually tries to boost the ranking of a specific page while concealing the boosting from the user. Common boosting techniques are content spamming (or keyword stuffing), which tries to manipulate the query-dependent part of the ranking algorithms, and link spamming, which tries to manipulate the query-dependent signal through the anchor text or the query-independent signal through the hyperlink analysis. Hiding techniques are usually very creative. They either attempt to hide the terms used for spamming from the user, such as the famous "white text on white background" approach, or they use cloaking, in which the spammer supplies the search engine with a page that is different from the page that a normal user sees when visiting the same URL. See (*24*) for more details.

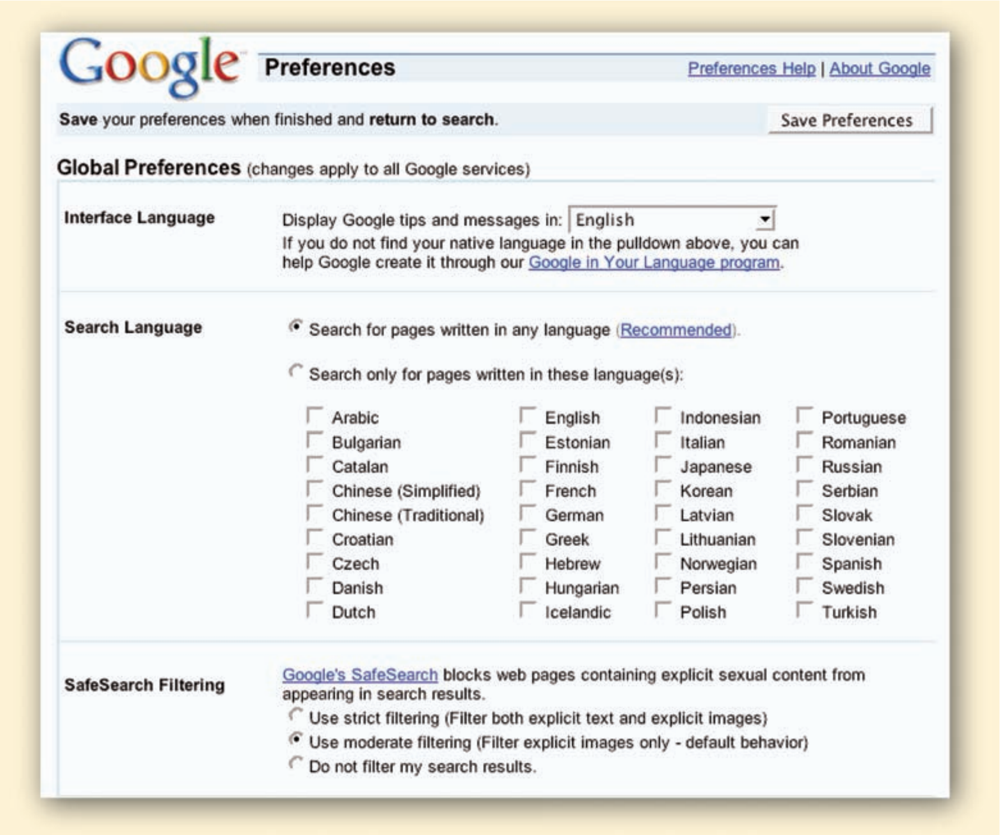
Detecting search engine spam is an ongoing research effort. First results indicate that automatic classifiers can be used to identify 82 to 86% of content spam (*25*) and 80 to 81% of link spam (*26*), with very small false positive rates.

Filters for inappropriate content. What is considered inappropriate content differs from culture to culture, and even from person to person. Thus, the first challenge when building a filter for inappropriate content is to find the right definition of what is inappropriate. There seems to be general agreement that filters for children should eliminate pornography, hate sites, and violence-related as well as drug-related material. Such content can often be detected by a classifier that was trained using machine-learning techniques. For training, the filter software is given a large set of "training" documents, which are documents that are annotated either as inappropriate or as not inappropriate; from this, the software builds a model of what features of documents are good indicators of inappropriateness. This model is then used to filter pages with inappropriate content at query time. These and other filters are available at search engines; see, for example, Fig. 2 for Google's filtering options that apply to all searches a user performs (i.e., personalized filter) and Fig. 3 for Google's filtering options for an individual search. In the future, search engines might provide filters for topics, geographic regions, or genres of Web pages.

Future Prospects

To further improve search results, specialized search engines such as Google Scholar, which contain pages only on a certain topic or a certain genre, have been created. Another thrust of current research on result ranking is to analyze user clicks on search results in the aggregate. Researchers are also experimenting with different search interfaces, such as multifaceted searching. Because no search engine indexes the whole Web, comprehensiveness can be improved by combining search results of various search engines. Rank aggregation is the research area that explores different ways of combining ranked lists of search results.

Other interesting research topics are searches of other types of media, such as images, video, and sounds. Current search engines usually exploit textual information associated with the media, such as the text in and surrounding an image, the closed caption of television channels, or user annotations of images, so-called social tagging. The quality of these information sources is variable, which in turn affects the search quality for these media. See (*27*) for more details.



Preferences

Preferences Help | About Google

Save your preferences when finished and return to search.

Save Preferences

Global Preferences (changes apply to all Google services)

Interface Language

Display Google tips and messages in: English ▾
If you do not find your native language in the pulldown above, you can help Google create it through our Google in Your Language program.

Search Language

● Search for pages written in any language (Recommended).

○ Search only for pages written in these language(s):

□ Arabic
□ Bulgarian
□ Catalan
□ Chinese (Simplified)
□ Chinese (Traditional)
□ Croatian
□ Czech
□ Danish
□ Dutch
□ English
□ Estonian
□ Finnish
□ French
□ German
□ Greek
□ Hebrew
□ Hungarian
□ Icelandic
□ Indonesian
□ Italian
□ Japanese
□ Korean
□ Latvian
□ Lithuanian
□ Norwegian
□ Persian
□ Polish
□ Portuguese
□ Romanian
□ Russian
□ Serbian
□ Slovak
□ Slovenian
□ Spanish
□ Swedish
□ Turkish

SafeSearch Filtering

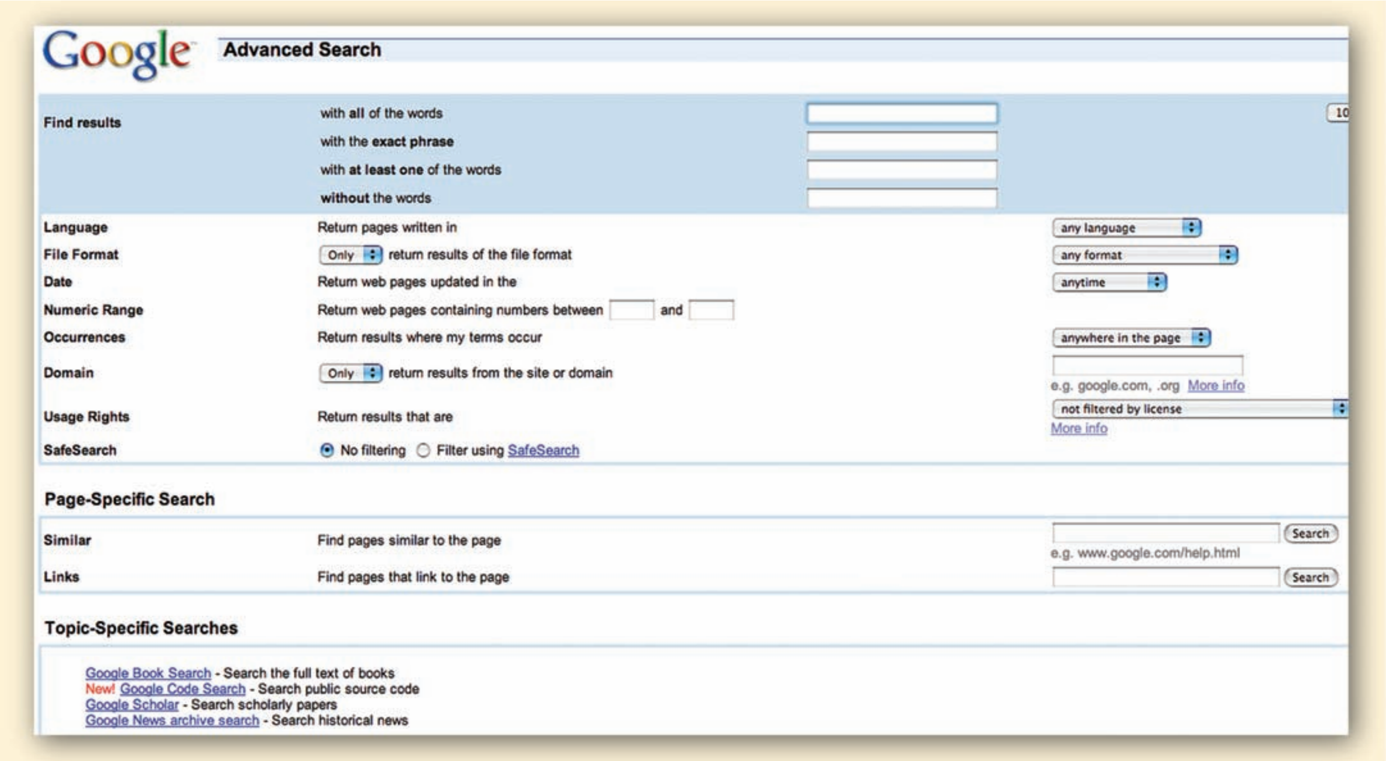
Google's SafeSearch blocks web pages containing explicit sexual content from appearing in search results.

○ Use strict filtering (Filter both explicit text and explicit images)
● Use moderate filtering (Filter explicit images only - default behavior)
○ Do not filter my search results.

Fig. 2. Filtering options of Google applying to all searches of a user.

Current search engines do not understand the semantics of queries or Web pages, nor do they apply any form of reasoning. Researchers currently experiment with augmenting search engines with some simple forms of reasoning: They try to extract facts from the Web and store them in databases (*28*). This would allow a search engine to answer questions of the form "List all objects with the following property," such as "Give me all cities in California with more than 1 million inhabitants." Simple deduction rules such as "A is in relation with B, and B is in relation with C, thus A is in relation with C" could then be applied. Other researchers retrieve facts from a manually compiled hierarchy of facts using a theorem prover and attempt to combine them with matches in documents with the goal of answering simple fact-based queries (*29*). Neither approach is used in search engines today, but might be in a future years.

Extracting facts from Web pages is closely related to searching semistructured data, such as XML (Extensible Markup Language) data. These kinds of searches arise frequently in enterprise



Google Advanced Search

Find results
with all of the words [ ] 10
with the exact phrase [ ]
with at least one of the words [ ]
without the words [ ]

Language — Return pages written in — any language
File Format — Only — return results of the file format — any format
Date — Return web pages updated in the — anytime
Numeric Range — Return web pages containing numbers between [ ] and [ ]
Occurrences — Return results where my terms occur — anywhere in the page
Domain — Only — return results from the site or domain [ ] e.g. google.com, .org More info
Usage Rights — Return results that are — not filtered by license More info
SafeSearch — ● No filtering ○ Filter using SafeSearch

Page-Specific Search

Similar — Find pages similar to the page [ ] Search e.g. www.google.com/help.html
Links — Find pages that link to the page [ ] Search

Topic-Specific Searches

Google Book Search - Search the full text of books
New! Google Code Search - Search public source code
Google Scholar - Search scholarly papers
Google News archive search - Search historical news

Fig. 3. Filtering options of Google applying to individual searches.

searching (searching the web pages internal to an enterprise) and in the search of digital libraries. Traditionally, databases have been used to search structured data, and search engines have been used for searching unstructured data, such as text. With the arrival of semistructured Web pages, the database and the information retrieval communities have started to explore combining their techniques and research efforts to achieve better retrieval results (*30*, *31*). Thus, a new field of research consisting of the combination of the two areas may be created.

References and Notes

1. S. Ghemawat, H. Gobioff, S.-T. Leung, in *Proceedings of the 16th ACM Symposium on Operation System Principles* (ACM Press, New York, 2003), pp. 29–43.
2. J. Dean, S. Ghemawat, in *Proceedings of the 6th Symp. Operating System Design and Implementation* (Usenix Association, Berkeley, CA, 2004), pp. 137–150.
3. L. Granka, T. Joachims, G. Gay, in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, K. Järvelin, J. Allan, P. Bruza, M. Sanderson, Eds. (ACM Press, New York, 2004), pp. 478–479.
4. S. Brin, L. Page, *Comput. Netw.* **30**, 107 (1998).
5. J. Kleinberg, *J. ACM* **46**, 604 (1999).
6. E. Garfield, *Science* **122**, 108 (1955).
7. M. K. Bergman, *J. Electronic Pub.* **7**, (2001); www.press.umich.edu/jep/07-01/bergman.html.
8. One obvious way of reducing index size is to omit duplicate and near-duplicate Web pages. According to studies (*32–34*), about 25 to 30% of Web pages can be discarded in this way, making space for other documents.
9. A uniform record locator (URL) is the equivalent of an address for Web pages.
10. W. Wu, A. Doan, C. Yu, in *Proceedings of the 22nd International Conference on Data Engineering*, L. Liu, A. Reuter, K.-Y. Whang, J. Zhang, Eds. (IEEE, Los Alamitos, CA, 2006), p. 44.
11. A. Ntoulas, P. Zerfox, J. Cho, in *Proceedings of the Joint Conference on Digital Libraries* (ACM Press, New York, 2005), pp. 100–109.
12. www.sitemaps.org.
13. A. Z. Broder, *SIGIR Forum* **36**, 3 (2002).
14. J. Teevan, S. Dumais, E. Horvitz, in *Proceedings of the 1st International Workshop on New Technologies for Personalized Information Access (PIA 2005)*, P. Brusilosky, C. Callaway, A. Nürnberger, Eds. Edinburgh, UK, 24 July 2005, DELOS Network of Excellence on Digital Libraries, pp. 84–92.
15. P. Anick, in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, J. Callan, G. Cormack, C. Clarke, D. Hawking, A. Smeaton, Eds. (ACM Press, New York, 2003), pp. 88–95.
16. X. Shen, B. Tan, C. Zhai, in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, G. Marchionini, A. Moffat, J. Tait, R. Baeza-Yates, N. Ziviani, Eds. (ACM Press, New York, 2005), pp. 43–50.
17. B. Tan, Z. Shen, C. Zhai, in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, T. Eliassi-Rad, L. Ungar, M. Craven, D. Gunopulos, Eds. (ACM Press, New York, 2006), pp. 718–723.
18. J. Teevan, S. Dumais, E. Horvitz, in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, G. Marchionini, A. Moffat, J. Tait, R. Baeza-Yates, N. Ziviani, Eds. (ACM Press, New York, 2005), pp. 449–456.
19. G. Jeh, J. Widom, in *Proceedings of the 12th International World Wide Web Conference* (ACM Press, New York, 2003), pp. 271–279.
20. P. Berkhin, *Internet Math.* **2**, 73 (2005).
21. A. N. Langville, C. D. Meyer, *Internet Math.* **1**, 335 (2005).
22. H. Chen, D. R. Karger, in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, S. Dumais, E. N. Efthimiadis, D. Hawking, K. Järvelin, Eds. (ACM Press, New York, 2006), pp. 429–436.
23. L. Wang *et al.*, in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, G. Marchionini, A. Moffat, J. Tait, R. Baeza-Yates, N. Ziviani, Eds. (ACM Press, New York, 2005), pp. 424–431.
24. Z. Gyöngyi, H. Garcia-Molina, *IEEE Comp. Mag.* **38**, 28 (2005).
25. A. Ntoulas, M. Najork, M. Manasse, D. Fetterly, in *Proceedings of the 2006 World Wide Web Conference*, L. Carr, D. De Roure, A. Iyengar, C. A. Goble, M. Dahlin, Eds. (ACM Press, New York, 2006), pp. 83–92.
26. L. Becchetti, C. Castillo, D. Donato, S. Leonardi, R. Baeza-Yates, in *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web*, B. D. Davison, M. Najork, T. Converse, Eds. (Tech. Rep. LU-CSE-06-027, Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, 2006), pp. 1–8.
27. M. Sahami, V. Mittal, S. Baluja, H. Rowley, in *Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence*, C. Zhang, H. W. Guesgen, W.-K. Yeap, Eds. (Springer, New York, 2004), pp. 3–12.
28. O. Etzioni *et al.*, *Artif. Intell.* **165**, 91 (2005).
29. D. Moldovan, C. Clark, S. Harabagiu, S. Maiorano, in *Proceedings of HLT-NAACL 2003, Human Language Techn. Conference of the North American Chapter of the ACL* (ACL Press, Cambridge, MA, 2003), pp. 87–93.
30. S. Amer-Yahia, P. Case, T. Rölleke, J. Shanmugasundaram, G. Weikum, *SIGMOD Record* **34**, 71 (2005).
31. D. Carmel, Y. S. Maarek, M. Mandelbrod, Y. Mass, A. Soffer, in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, J. Callan, G. Cormack, C. Clarke, D. Hawking, A. Smeaton, Eds. (ACM Press, New York, 2003), pp. 151–158.
32. A. Z. Broder, S. Glassman, M. Manasse, G. Zweig, *Comput. Netw.* **29**, 1157 (1997).
33. D. Fetterly, M. Manasse, M. Najork, in *Proceedings of the 1st Latin American Web Congress (LA-WEB 2003)* (IEEE, Los Alamitos, CA, 2003), pp. 37–45.
34. M. Henzinger, in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, S. Dumais, E. N. Efthimiadis, D. Hawking, K. Järvelin, Eds. (ACM Press, New York, 2006), pp. 284–291.

10.1126/science.1126557