Criteria for Evaluating Information Retrieval Systems in Highly Dynamic Environments

Judit Bar-Ilan
School of Library, Archive and Information Studies
The Hebrew University of Jerusalem
P.O. Box 1255, Jerusalem, 91904, Israel
e-mail: judit@cc.huji.ac.il

Abstract

This paper proposes a set of measures to evaluate search engine functionality over time. When coming to evaluate the performance of Web search engines, the evaluation criteria used in traditional information retrieval systems (precision, recall, etc.) are not sufficient. Web search engines operate in a highly dynamic, distributed environment, therefore it becomes necessary to assess search engine performance not just at a single point in time, but over a whole period.

The size of a search engine's database is limited, and even if it grows, it grows more slowly than the Web. Thus the search engine has to decide whether and to what extent to include new pages in place of pages that were previously listed in the database. The optimal solution is that all new pages are listed, and no old ones are removed - but this of course is usually unachievable. The proposed metrics that evaluate search engine functionality in presence of dynamic changes include the percentage of newly added pages, and the percentage of the removed pages, which still exist on the Web. The percentage of non-existent pages (404 errors, nonexistent server, etc.) out of the set of retrieved pages indicates the timeliness of the search engine.

The ideas in this paper elaborate on some of the measures introduced in a recently published paper (Bar-Ilan, 2002). I'd like to take advantage of the opportunity to discuss the problem of search engine evaluation in dynamic environments with the participants of the Web Dynamics Workshop.

Introduction

The World Wide Web is a very different environment from the usual setting, in which traditional information retrieval (IR) systems operate. Traditional systems operate in a highly controlled, centralized and relatively stable environment. New documents can be added, but in a controlled fashion. Sometimes old documents are removed or moved (for example in case the "current" database of a bibliographic retrieval system contains only documents from the last two years, and the older ones are moved to an archive); and documents or document representations may change - mistakes can be corrected. The major point is that all these processes are controlled.

The Web, on the other hand is uncontrolled, distributed and highly dynamic - in short, total chaos. The situation was aptly expressed by Chakrabarti et al. (1999) “the Web has evolved into a global mess of previously unimagined proportions”. On the Web almost anyone can publish almost anything. Later the author of the page or the publishing site can decide to: change the content; remove the page or move it to another directory on the same server or to a different server; or publish the same content at another URL. All these changes occur continuously and the search engines are not being notified of any of them. They have to cope not only with the dynamic changes caused by the authors and the publishers (the servers), but also with problems on the way to these pages: communication or server failures.

Web search engines operate in a chaotic environment, which is rather different from the stable, controlled setting of classical IR systems. Still, up till now most studies evaluating search engines used traditional IR evaluation criteria. The best known IR evaluation measures are *precision* and *recall*. A large number of studies evaluated *precision* or top-ten precision (e.g. Leighton & Srivastava, 1999 or Gordon & Pathak, 1999); while only a very few attempted to estimate *recall* (e.g. Clarke & Willett, 1997). Precision and recall are the most widely used measures of effectiveness; but other criteria were also used to assess Web search performance (e.g. Zhu & Gauch, 2000; Singhal & Kaszkiel, 2001). The search engines' *coverage* of the Web has also been estimated (Bharat & Broder, 1998; Lawrence & Giles, 1998 and 1999). A number of early studies compared the *search capabilities* of the different search tools, usually based on the documentation provided by the search tools (e.g. Courtois, 1996). User studies assessed *satisfaction* [e.g. the NDP survey reported by Sullivan (2000a)] and *search behavior* (e.g. Watson, 1998; Jansen, Spink & Saracevic, 2000; or Holscher & Strube, 2000). Other IR evaluation criteria include studies on output form (e.g. Tombros & Sanderson, 1995; Zamir & Etzioni, 1999) or *usability issues* like interface design (e.g. Berenci et al., 1999).

Issues related to searching in a dynamic environment have already been addressed, but not from the evaluation perspective. Some works studied the rate of change of Web pages in order to assess the benefits of caching (e.g., Douglis et al., 1997;) or to devise refresh schedules for the search engines (e.g., Brewington & Cybenko, 2000; Cho & Garcia-Molina, 2000) to be as *fresh* and *timely* as possible using available resources, or to characterize changes occurring to Web pages and sites over time (e.g. Koehler, 1999 & 2002, Bar-Ilan & Peritz, 1999). Several works (e.g., Bar-Ilan, 1999; Rousseau, 1999; Bar-Ilan, 2000) reported huge fluctuations over time in the number of results search engines retrieve for given queries.

Lawrence and Giles (1999) raised an interesting point: "There may be a point beyond which it is not economical for them [the search engines, J. B] to improve their coverage and timeliness." Thus in addition to the technical and algorithmic difficulties, the financial aspects must also be taken into account. We are not only facing the question whether the search engines *can* cope with the growth and changes on the Web, but also whether they *want* to cope.

The major issues of interest when coming to evaluate search engine performance in a dynamic environment are:

1) Timeliness/freshness

   Percentage of broken links

   Percentage of pages with where the indexed copy differs from the Web copy

   Percentage of "recently" created pages in the database

2) Stability over time

   Are there great fluctuations in the number results for a given query?

   Does the search engine "drop" from its database existing URLs relevant to the query?

In the next session we describe the necessary framework for evaluation, then we formally define the metrics and discuss their meaning.

The Framework

In order to evaluate search engine performance over a period of time, the query/queries have to be asked periodically from the search engine. The

query/queries are run in *search rounds*. The *search period* is the span of time during which the searches were carried out. The search rounds should be equidistant. From our experience it is sufficient to run the query/queries once a month.

We experienced with running the query once a week, but the observed changes were not very significant. An exception was an experiment we carried out in September-October, 1999 when huge daily fluctuations were observed in the results of HotBot. Notess (n.d.) reports that AltaVista has an ongoing problem with the number of results: because of unreported timeouts, it may retrieve a different number of results each time the search button is pressed. We have not encountered such problems with AltaVista during our searches. However, we made sure that the queries were run at a time when Internet communication is known to be low, on Sunday early mornings (around 5:00-7:00 GMT).

In order to compute the percentage of newly added pages, the percentage of "dropped" pages and the percentage of broken links, all the URLs the search results point to must be visited in every search round. The best solution is to download all the pages the search results point to immediately after the query is run. This way the results can be examined in a more leisurely fashion, and more importantly, they can be viewed as they were seen at the time the searches were carried out by anyone wishing to inspect the results at a later time.

The above requirement restricts the queries on which the search engine can be evaluated. The entire set of search results must be examined, thus the query has to be such, that the search engine presents all the hits for the given query. Most search engines limit the number of displayed results - they usually do not display more than the first 1000 results (AltaVista displays only 200, but this problem can be partially solved by carrying out several searches limited to different dates of creation of the URLs). Further steps must be taken in order to retrieve all the hits for search engines that cluster the search results.

We also need a method to decide which of the retrieved documents are "relevant" to the query. Relevance is a very difficult notion and has been heavily discussed by the IR community [see for example (Saracevic, 1975) or (Mizzaro, 1997)] - there is no general agreement on how to judge relevance, even though relevance is the basis for computing the most widely used IR evaluation measures: precision, recall and coverage.

Human relevance judgment in case of periodic searches with a large number of results is not feasible, thus we defined a more lenient measure, called *technical relevance* that can be computed automatically. A document is defined to be *technically relevant* if it fulfills all the conditions posed by the query: all search terms and phrases that suppose to appear in the document do appear, and all terms and phrases that are supposed to be missing from the document - terms preceded by a minus sign or a NOT operator, do not appear in the document. A URL is called a *technically relevant URL*, if it contains a technically relevant document (Bar-Ilan, 2002). Lawrence and Giles also took this approach (1999), even though they point out: "search engines can return documents that do not contain the query terms (for example documents with morphological variants or related terms)." It is advisable to choose query terms with as few morphological variants as possible (Northern Light, for example, did not differentiate between pages in which the query term appears in singular or in plural - in case of simple plural). From our experience, currently, related terms or concepts are very rarely substituted for the original query terms. Thus the notion of technical relevance provides a fast and easy method to differentiate between pages "about" the search topics and pages that clearly have nothing to do with the query (including broken links and otherwise inaccessible pages).

The Metrics - Definitions

To evaluate the percentage of broken links, we define:

broken(q,i) = (# broken links) / (total # results retrieved for query *q* in search round *i)*

There are temporary communication failures, which may result in 404 messages, thus a second attempt must be made (at a slightly later time) to download these pages before deciding that they are really missing or inaccessible.

Next we introduce *new* which counts the number of newly added URLs for *i*>1:

new(q,i) = |{technically relevant URLs retrieved in round *i*} - {technically relevant URLs retrieved by the search engine in search round *j* where *j*<*i*}|

This measure is influenced both by the growth of the subject on the Web and by the rate at which the search engine adds new pages to its database. A *new* page may be added to the search engine's database for two reasons: 1) the page has been created recently or its content was recently changed, so that the page became relevant for the query; 2) the page had already existed and had been relevant to the query for a "long" time, but the search engine only recently discovered and added it to its database. In order to try to differentiate between the two factors influencing *new,* we may run the same query on several large search engines in parallel, and try to create an "exhaustive" pool of pages technically relevant to the query for each search round. Then we can partition *new(q,i)* into *totally-new(q,i,s)* and *newly-discovered(q,i,s),* where

totally-new(q,i,s) =|{ technically relevant URLs retrieved by the search engine *s* in search round *i* }- {URLs in the pool of URLs retrieved before round *i*}|

newly-discovered(q,i,s) = new(q,i)- totally-new(q,i,s)

There are no easy means to decide whether the search engine's information is outdated, except, perhaps, in case the document is totally unrelated with the query (not even the same concept). Some partial conclusions may be drawn form the search engine's summaries. An exception is Google, which caches most of the URLs it visited, thus is possible to compare the downloaded pages with the cached version. In order to carry out this comparison, the cached documents should also be downloaded - to compare the local and the Web copy, as they existed at the given point in time. These suggestions are "work-arounds", thus we have not defined a measure to evaluate the extent to which the search engine's information is outdated. Measures like *freshness* (Cho & Garcia-Molina, 2000) can be used also for evaluation, in case the evaluator has access to the search engine's database and the page as seen by the crawler can be reconstructed.

In order to evaluate stability, we introduce the following measures for *i*>1:

forgotten(q,i)=|{technically relevant URLs retrieved in round *(i-1)*, that exist on the Web and are technically relevant at round *i*, but are *not* retrieved in round *i*}|

A *dropped URL* is a URL that disappeared from the search engine's database, even though it still exists on the Web and is continues to be technically relevant; *forgotten(q,i)* counts the number of *dropped* URLs in round *i*. A URL *u* that was dropped in round *i*, may reappear in the database at some later round *j* (our experience shows that this does happen). Such URLs are called *rediscovered URL*s. *Recovered(q,j)* counts the number of *rediscovered* URLs in round *i*>1:

recovered(q,j)=|{technically relevant URLs retrieved in round *j* that were *dropped* in round *i*, *i*<*j* AND were not retrieved in round *j-1*}|

If a URL *u* appeared for the first time in round *k*, and was *dropped* in round *i*>*k*, and reappeared in round *j*>*i*, and was retrieved again in round *j+1*, it will be *rediscovered*

in round *j*, but not in round *j+1* i.e., a URL is counted in recovered in the first round it reappears after being dropped.

It may be the case that the URL was dropped in round *i* because the server on which it resides was down at the time the crawler tried to visit it. This may account for some (small part) of forgotten. There are two other possible explanations. The first: the search engine has limited resources, and it has to keep the balance between the newly discovered pages and the old pages in its database. The second explanation relates to the crawling policy of the search engine. If the search engine uses shadowing (see (Arasu, 2000): it has a database that serves the queries, and another database that is being built based upon the current crawling; the "new" database replaces the old one at some point of time), it is possible that the new database covers a substantially different set from the old one.

It is well known that there is a lot of content duplication on the Web. Some of it is intentional (mirror sites), some results from simple copying of Web pages, and some is due to different aliases of the same physical address. Thus it is conceivable that the search engine dropped a given URL *u*, because it located another URL *u'* in its database with exactly the same content. To evaluate the extent to which content is lost we define:

lost(q,i)=|{*dropped* URLs in round *i*, for which there is no other URL which was retrieved for *q* in round *i*, with exactly the same content}|

Lost URLs are those URLs, which were dropped, and the search engine did not retrieve any content duplicates of these URLs in the current search round. These URLs cause real information loss for the users, information that was accessible through the search engine before, is not accessible anymore, even though the information is still available and pertinent on the Web. The results of a case study (Bar-Ilan, 2002) show that not only a high percentage of the URLs are *dropped* and *rediscovered*, but a significant portion of them were also *lost*.

A URL can be *dropped* and then *rediscovered* several times during the search period. In order to assess the search performance over the whole search period we define:

well-handled(q)=|{technically relevant URLs retrieved for *q* that were never *dropped* during the *search period*}|

The URLs counted in well-handled are not necessarily retrieved during the whole search period. Such a URL can first appear in round *i>1*, and disappear from the list of retrieved URL in round *j>i*, if it disappears from the Web or ceases to be technically relevant to the query. A *mishandled* URL is a URL that was *dropped* at least once during the *search period*. Recall that *dropped* means that the URL wrongfully disappeared from the list of URLs retrieved for the query.

mishandled(q)=|{U *dropped URLs*, for *i>1*}|

The set of *mishandled URLs* can be further partitioned into *mishandled-forgotten(q)* - these are the URLs that were not rediscovered at some later time, and to *mishandled-recovered(q)*.

The last two measures assess the variability of the search results over time, and supplement the measures *new* and *forgotten*:

self-overlap(q,i,j)=|{technically relevant URLs that were retrieved both in round *i* and in round *j*}| / |{technically relevant URLs retrieved in round *j*}|

Let *All(q)* denote the set of all technically relevant URLs that were retrieved for the query *q* during the whole search period. Note that this is a virtual set; since it may include URLs that never coexisted on the Web at the same time.

self-overlap(q,i)=|{technically relevant URLs that were retrieved in round *i* }| / |{*All(q)*}|

High self-overlap for all search rounds indicates that the search engine results for the given query are stable. Note that very high values of self-overlap not only indicate stability, but may also be warning signs that the search engine's database is becoming out of date - has not changed substantially for a long period of time. Thus for this measure the "optimal" values are neither very high nor very low.

Evaluating Using these Measures and Future Work

An initial study (Bar-Ilan, 1999) was carried out for a period of five months in 1998 with the query "informetrics OR informetric", using the six largest search engines at the time (AltaVista, Excite, HotBot, Infoseek, Lycos and Northern Light). In this study forgotten and recovered were computed, and Excite "forgot" 72% of the technically relevant URLs retrieved by it. In each of the search rounds Excite retrieved almost the same number of results (158 URLs on the average), but when we compared the sets of URLs we were rather surprised to discover that the overlap between the sets was very small - during the whole search period Excite retrieved a total of 535 technically relevant URLs. This result shows that it is not sufficient to look at the number of results only, but the URLs must also be examined.

We carried out a second case study (Bar-Ilan, 2002) evaluating most of the above-defined measures for a whole year during 2000. The query in the case study was "aporocactus". This word has few (if any) morphological variants. The query was run on six search engines (AltaVista, Excite, Fast, Google, HotBot and Northern Light) in parallel. The search engines mishandled between 33 and 89 percent of the technically relevant URLs retrieved by them during the whole search period. This time the search engines that mishandled the largest percentages of URLs were Google (89%) and HotBot (51%); even though Google retrieved by far the largest number of technically relevant URLs during the whole period - Google covered more than 70% of the set *All*. Except for Northern Light, almost all of the *forgotten URL*s were also *lost*, i.e. we were unable to locate in the search results another URL with exactly the same content.

Naturally, we cannot draw any definite conclusions about specific search engines based on two queries during two search periods; but the case studies indicate the usefulness of the evaluation criteria defined in this paper.

The "optimal search engine" should have high values for new, corresponding to the growth of the subject on the Web. Ideally the number of mishandled URLs should be zero, but as we explained before, the search engine has to decide on how to utilize its available resources, and has to compromise between adding new pages and removing old ones. The number of broken links should also approach zero, while self-overlap should neither be very high nor very low.

When counting dropped and lost URLs we may also want to look at the rank of these URLs. Are these mostly lowly ranked URLs or is the distribution more or less uniform? Dropping lowly ranked URLs would correspond to the policy announced by Inktomi (Sullivan, 2000b) of removing non-popular URLs from the database. The above-mentioned case study did not look at the ranks of the dropped URLs.

The criteria introduced here are a first step in defining a set of measures for evaluating search engine performance in dynamic environments. Future work should be carried out both in the theoretical and in the practical directions. We need to define additional criteria and to refine existing ones; and to carry out additional larger scale experiments to study the usefulness and the applicability of the measures.

Reference

Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., & Raghavan, S. (2000). Searching the Web. Stanford University Technical Report 2000-37. [Online] Available: http://dbpubs.stanford.edu/pub/2000-37

Bar-Ilan, J. (1999). Search Engine Results over Time - A Case Study on Search Engine Stability. *Cybermetrics*, 2/3. [Online]. Available: http://www.cindoc.csic.es/cybermetrics/articles/v2i1p1.html

Bar-Ilan, J. (2000). Evaluating the Stability of the Search Tools Hotbot and Snap: A Case Study. *Online Information Review*, vol, 24(6): 430-450

Bar-Ilan, J. (2002). Methods for Measuring Search Engine Performance over Time. *JASIST,* 54(3).

Bar-Ilan, J., & Peritz B. C. (1999). The Life Span of a Specific Topic on the Web; the Case of 'Informetrics': a Quantitative Analysis. *Scientometrics*, 46(3): 371-382.

Berenci, E., Carpineto, C., Giannini, V., & Mizzaro, S. (1999). Effectiveness of keyword-based display and selection of retrieval results for interactive searches. In *Lecture Notes in Computer Science*, 1696: 106-125.

Bharat, K. and Broder, A. (1998). A Technique for Measuring the Relative Size and Overlap of Public Web Search Engines. In *Proceedings of the 7th International World Wide Web Conference*, April 1998, [Also online]. Available: http://decweb.ethz.ch/WWW7/1937/com1937.htm

Brewington, B. E., & Cybenko, G. (2000). How Dynamic is the Web? In *Proceedings of the 9th International World Wide Web Conference*, May 2000. [Also online]. Available: http://www9.org/w9cdrom/264/264.html

Chakrabarti, S., Dom B., Kumar, R. S., Raghavan, P., Rajagopalan, S., Tomkins, A., Kleinberg, J. M., & Gibson, D. (1999). Hypersearching the Web. *Scientific American*, 280(6): 54-60. [Also online]. Available: http://www.sciam.com/1999/0699issue/0699raghavan.html.

Cho, J., & Garcia-Molina, H. (2000). Synchronizing a Database to Improve Freshness. *SIGMOD RECORD*, 29(2): 117-128.

Chu, H. & Rosenthal, M. (1996). Search Engines for the World Wide Web: A Comparative Study and Evaluation Methodology. *ASIS96*. [Online]. Available: http://www.asis.org/annual-96/Electronic-Proceedings/chu.htm

Clarke, S.J., & Willett, P. (1997). Estimating the Recall Performance of Web Search Engines. *Aslib Proceedings*, 49(7), 184-189.

Courtois, M. P. (May/June 1996) Cool Tools for Searching the Web - An Update. *Online,* 29-36.

Douglis, F.; Feldmann, A.; Krishnamurthy, B.; & Mogul, J. (1997). Rate of Change and Other Metrics: A Live Study of the World Wide Web. In Proceedings of the Symposium on Internet Technologies and Systems, Monterey, California, December 8-11, 1997. [Online]. Available: http://www.usenix.org/publications/library/proceedings/usits97/full_papers/douglis_rate

Holscher, C., & Strube, G. (2000). Web Search Behavior of Internet Experts and Newbies. . In *Proceedings of the 9th International World Wide Web Conference*, May 2000. [Online]. Available: http://www9.org/w9cdrom/81/81.html

Gordon, M., & Pathak, P. (1999). Finding Information on the World Wide Web: The Retrieval Effectiveness of Search Engines. *Information Processing and Management*, 35, 141-180.

Jansen, B. J.; Spink, A; & Saracevic, T. (2000). Real Life, Real Users, and Real Needs: A Study and Analysis of User Queries on the Web. *Information Processing and Management*, 36, 207-227.

Koehler, W. (1999). An Analysis of Web Page and Web Site Constancy and Permanence. *JASIS*. 50(2): 162-180.

Koehler, W. (2002). Web Page Change and Persistence - A Four Year Longitudinal Study. *JASIST*. 50(2): 162-180.

Lawrence, S. and Giles, C. L. (1998). Searching the World Wide Web. *Science*, 280, 98-100.

Lawrence, S. and Giles, C. L. (1999). Accessibility and Distribution of Information on the Web. *Nature*, 400: 107-110.

Leighton, H.V., & Srivastava, J. (1999). First 20 precision among World Wide Web search services (search engines). *JASIS*, 50, 870-881.

Mizzaro, S. (1997). Relevance: The Whole History. *JASIS*, 48(9): 810-832.

Notess, G. (no date). AltaVista Inconsistencies. [Online]. Available: http://searchengineshowdown.com/features/av/inconsistent.shtml

Rousseau, R. (1999). Daily Time Series of Common Single Word Searches in AltaVista and NorthernLight. *Cybermetrics*, 2/3(1), paper 2, [Online]. Available: http://www.cindoc.csis.es/cybermetricc/articles/v2i1p2.html

Saracevic, T. (1975). RELEVANCE: A review of and a Framework for the Thinking on the Notion in Information Science. *JASIS*, November-December, 1975, 321-343.

Sherman, C. (1999). The Search engines Speak. In *Web Search.* [Online]. Available: http://websearch.about.com/internet/websearch/library/weekly/aa120399.htm

Singhal, A. & Kaszkiel, M. (2001). A Case Study in Web Search Using TREC Algorithms. . In *Proceedings of the 10th International World Wide Web Conference*, May 2001. [Online]. Available: http://www10.org/cdrom/papers/317/paper.html

Sullivan, D. (2000a). NDP Search and Portal Site Study. In *Search Engine Watch Reports*. [Online]. Available: http://searchenginewatch.com/reports/npd.html

Sullivan, D. (2000b). *The Search Engine Update*, August 2, 2000, Number 82. [Online]. Available: http://searchenginewatch.com/subscribers/updates/000802su.html

Tombros, A., & Sanderson, M. (1998). The advantages of query-biased summaries in Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* , 2-10.

Watson, J. S. (1998) "If You Don't Have It, You Can't find It." A Close Look at Students' Perceptions of Using Technology. *JASIS,* 49(11), 1024-1036.

Zamir, O., & Etzioni, O. (1999). Grouper: A Dynamic Clustering Interface to Web Search Results. In *Proceedings of the 8th International World Wide Web Conference*, May 2000. [Online]. Available: http://www8.org/w8-papers/3a-search-query/dynamic/dynamic.html

Zhu, X., & Gauch, S. (2000). Incorporating Quality Metrics in Centralized/Distributed Information Retrieval on the World Wide Web. In *Proceedings of the 23rd International ACM SIGIR Conference.* July 2000, Athens, Greece, 288-295.