Web Search

Ricardo Baeza-Yates and Carlos Castillo

Center for Web Research

Dept. of Computer Science

University of Chile

{rbaeza,ccastill}@dcc.uchile.cl

Abstract

The World Wide Web has become in a few years into the largest cultural endeavour of all times. The Web is a distributed repository of information without a central point of control, and thus can be seen as a vast, diverse, rapidly changing and unstructured database.

The low cost of publishing information on the Web is a key part of its success, but implies that searching information on the Web will always be inherently more difficult that searching information in traditional, closed repositories.

Suggested Keywords: Web search, World Wide Web, Information retrieval, Link analysis, Web crawling, Web characterization

Suggested Cross-references: Computer-mediated communication; Disambiguation, lexical; Document Retrieval, Automatic; Parsing, statistical methods; Stemming; Indexing, Automatic; Text Mining

1 Introduction

A Web search engine takes a user need, usually stated in the form of a few keywords, and returns a list of Web pages that can be considered relevant for the given keywords. These Web pages are a short list of usually few hundred items selected from a vast repository of thousands of millions of pages.

The "easy" part of Web search is to find which documents in the Web are related to a given query, because most queries are very broad, and there are thousands of pages relevant to most basic concepts. The hard part of Web search is to rank those documents by relevance and select the, say, top 10, to show the first result page to the user.

Although there was an important body of information retrieval algorithms and techniques published before the invention of the World Wide Web, there are unique characteristics of this new medium that made those techniques unsuitable or insufficient for Web search.

> "Information retrieval algorithms were developed for relatively small and coherent collections such as newspaper articles or book catalogs in a (physical) library. The Web, on the other hand, is massive, much less coherent, changes more rapidly, and is spread over geographically distributed computers ..." (Arasu et al., 2001)

The Web can be considered as divided into two parts, the "closed Web" and the "open Web" (Brooks, 2003). The "closed Web" comprises a few high-quality controlled collections on which a search engine can fully trust. The "open Web" includes the vast majority of Web pages, which lack an authority asserting their quality. In the open Web, traditional information retrieval techniques, concepts and methods are challenged.

To partly overcome this problem, hyperlinks between pages can be used in the same way as

citations can be used in academic literature to find the most important papers in an area. Link analysis techniques can be used to exploit hyperlinks and extract useful information from them.

However, one of the main challenges that the open Web poses to search engines is "search engine spamming", i.e.: malicious attempts to get an undeserved high ranking in the list of results. This has created a whole branch of research called "**adversarial information retrieval**" that deals with retrieving information from collections in which a subset of the collection has been manipulated to influence the outcome of ranking algorithms.

In the next sections we present the main issues related to indexing, searching, ranking, and crawling the Web. We end with a section about the characteristics with the Web and another about research issues. Because of space restrictions our coverage of details and bibliography is by no means complete. For further details about Web retrieval we recommend (Baeza-Yates and Ribeiro-Neto, 1999).

2 Indexing and querying Web pages

The Web search process has two main parts: off-line and on-line.

The off-line part is executed periodically by the search engine, and consists in downloading a sub-set of the Web to build a collection of pages, which is then transformed into a searchable **index**.

The on-line part is executed every time a user query is executed, and uses the index to select some candidate documents that are sorted according to an estimation on how relevant they are for the user's need. This process is depicted in Figure 1.

Web pages come in many different formats such as plain text, HTML pages, PDF documents, and other proprietary formats. The first stage for indexing Web pages is to extract a standard logical view from the documents. The most used logical view for documents in search engines is

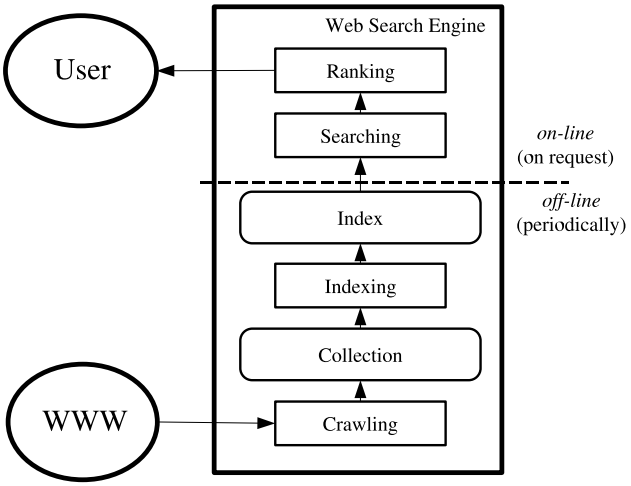

Figure 1: A Web search engine periodically downloads and indexes a sub-set of Web pages. This index is used for searching and ranking in response to user queries.

the "bag of words" model, in which each document is seen only as an unordered set of words. In modern Web search engines, this view is extended with extra information concerning word frequencies and text formatting attributes, as well as meta-information about Web pages including embedded descriptions and explicit keywords in the HTML markup.

There are several **text normalization operations** (Baeza-Yates, 2004) that are executed for extracting keywords, the most used ones are: tokenization, stopword removal and stemming .

Tokenization involves dividing the stream of text into words. While in some languages like English this is very straighforward and involves just splitting the text using spaces and punctuation, in other languages like Chinese finding words can be very difficult.

Stopwords are words that carry little semantic information, usually functional words that appear in a large fraction of the documents and therefore have little discriminating power for asserting relevance. In information retrieval stopwords are usually discarded also for efficiency reasons, as storing stopwords in an index takes considerable space because of their high frequency.

Stemming extracts the morphological root of every word. In global search engines, the first problem with stemming is that it is language dependent, and while an English rule-based stemming works well, in some cases like Spanish, a dictionary-based stemmer has to be used, while in other languages as German and Arabic stemming is quite difficult.

Other, more complex operations such as synonim translation, detecting multiword expressions, phrase identification, named entity recognition, word sense disambiguation, etc. are used in some application domains. However, some of these operations can be computationally expensive and if they have large error rates, then they can be useless and even harm retrieval precision.

2.1 Inverted index

An inverted index is composed of two parts: a vocabulary and a list of occurrences. The vocabulary is a sorted list of all the keywords, and for each term in the vocabulary, a list of all the "places" in which the keyword appears in the collection is kept. Figure 2 shows a small inverted index, considering all words including stopwords. When querying, the lists are extracted from the inverted index and then merged. Queries are very fast because usually hashing in memory is used for the vocabulary, and the lists of occurrences are pre-sorted by some global relevance criteria.

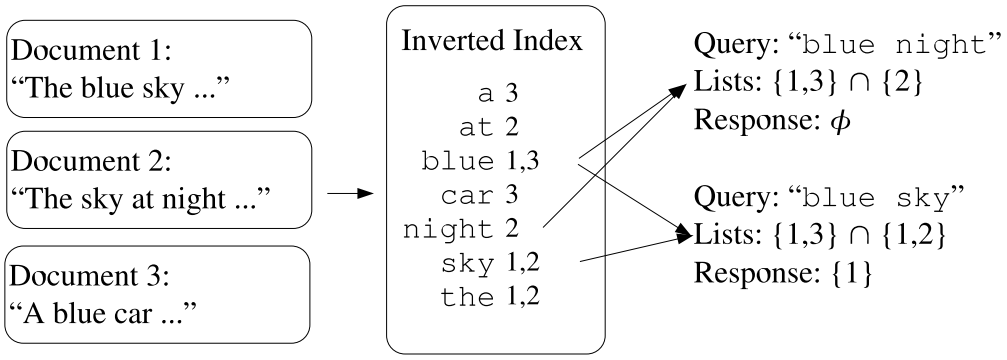

Figure 2: A sample inverted index.

The granularity of the choice of the items in the list of occurrences determines the size of the index, and a small size can be obtained by storing only the document identifiers of the corresponding documents. If the search engine also stores the position where the term appears on each page the index is larger, but can be used for solving more complex queries such as queries for exact phrases, or proximity queries.

While the vocabulary grows sub-linearly with the collection size, the list of occurrences can be very large. The complete inverted index can occupy from 10% to 20% of the space occupied by the actual collection. An inverted index does not fit in main memory for a Web collection, so several partial indices are built. Each partial index represents only a subset of the collection and are later merged into the full inverted index.

In Figure 3 the main stages of the indexation process are depicted. During parsing, links are extracted to build a Web graph, and they can be analyzed later to generate link-based scores that can be stored along withe the rest of the metadata.

2.2 Distributing query load

Query response time in today's search engines requires to be very fast, and should be done in a parallel way involving several machines. For parallelization, the inverted index is usually distributed among several physical computers. To partition the inverted index, two techniques are used: global inverted file and local inverted file (Tomasic and Garcia-Molina, 1993).

When using a global inverted file, the vocabulary is divided into several parts containing roughly the same amount of occurrences. Each computer is assigned a part of the vocabulary and all of its occurrences. Whenever a query is received, the query is sent to the computers holding the query terms, and the results are merged afterwards. Hence, load balancing is not easy.

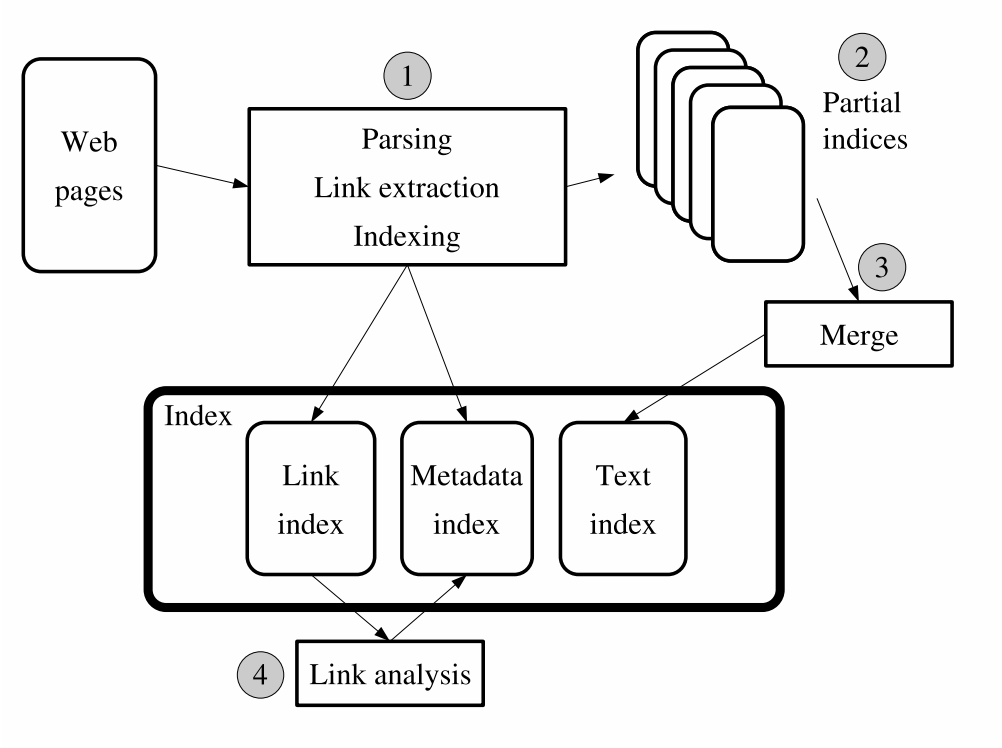

Figure 3: Indexing for Web search. (1) Pages are parsed and links and extracted. (2) Partial indices are written on disk when main memory is exhausted. (3) Indices are merged into a complete text index. (4) Off-line link analysis can be used to calculate static link-based scores.

When using a local inverted file, the document identifiers are divided, but each computer gets the full vocabulary. That is, step 3 in figure 3 is ommitted. A query is then broadcasted to all computers, obtaining good load balance. This is the architecture used in main search engines today, as building and maintaining a global index is hard.

Query processing involves a central "broker" that is assigned the task of distributing incoming queries and merging the results. As the results are usually shown in groups of 10 or 20 documents per page, the broker does not need to request or merge full lists, only the top most results from each partial list.

Search engines exploit the fact that users seldom go past the first or second page of results. Search engines provide approximate result counts because they never perform a full merge of the partial result lists, so the total number of documents in the intersection can only be estimated. For this reason, when a user asks for the second or third page of results for a query, it is common that the full query is executed again.

3 Ranking

3.1 Text-based ranking

The vector space model (Salton, 1971) is the standard technique for ranking documents according to a query. Under this model, both a document and a query are seen as a pair of vectors in a space with as many dimensions as terms as the vocabulary. In a space defined in this way, the similarity of a query to a document is given by a formula that transforms each vector using certain weights and then calculates the cosine of the angle between the two weighted vectors:

$$sim_{(q,d)} = \frac{\sum_t w_{t,q} \times w_{t,d}}{\sqrt{\sum_t w_{t,q}^2} \times \sqrt{\sum_t w_{t,d}^2}}$$

In a text-based information retrieval systems, documents are shown to the user in decreasing order using this similarity measure.

A weighting scheme uses statistical properties from the text and the query to give certain words more importance when doing the similarity calculation. The most used scheme is the **TF-IDF weighting scheme** (Salton and Buckley, 1988), that uses the frequency of the terms in both queries and documents to compute the similarity.

TF stands for **term frequency**, and the idea is that a that if a term appears several times

in a document it is better as for describing the contents of that document. The TF is usually normalized with respect to document length, that is, the parameter used is the frequency of term t divided by the frequency of the most frequent term in document d:

$$tf_{t,d} = \frac{freq_{t,d}}{\max_\ell freq_{\ell,d}}$$

IDF stands for **inverse document frequency** and reflects how frequent a term is in the whole collection. The rationale is that a term that appears in a few documents gives more information that a term that appears in many documents. If N is the number of documents and n_t if the number of documents containing the query term t, then $idf_t = \log \frac{N}{n_t}$.

Using these measures, the weight of each term in given by:

$$w_{t,q} = \left(\frac{1}{2} + \frac{1}{2} tf_{t,q}\right) idf_t \ , \qquad w_{t,d} = tf_{t,d}$$

The 0.5 added is added to avoid a query term having 0 weight. Several alternative weighting schemes have been proposed, but this weighting scheme is one of the most used and gives good results in practice.

3.2 Connectivity-based ranking

Web pages sharing a link are more likely to be topically related that unconnected Web pages (Davison, 2000). The key hypothesis of connectivity-based ranking goes one step further, and asserts that a hyperlink from a page p' to a page p, means, in a certain way, that the content of page p is endorsed by the author of page p'.

By the same reasons why self-citations in academic literature should not confer authority, link analysis techniques must be aware of the fact that thousands of links can be created automatically by the same user:

> "Unlike academic papers which are scrupulously reviewed, web pages proliferate free of quality control or publishing costs. With a simple program, huge numbers of pages can be created easily, artificially inflating citation counts. Because the Web environment contains profit seeking ventures, attention getting strategies evolve in response to search engine algorithms. For this reason, any evaluation strategy which counts replicable features of Web pages is prone to manipulation" (Page et al., 1998).

The algorithms for connectivity-based ranking based on this assumption can be partitioned into (Henzinger, 2001):

- *Query-independent ranking*, that assign a fixed score to each page in the collection.
- *Query-dependent ranking*, or topic-sensitive ranking, that assign a score to each page in the collection in the context of a specific query.

3.2.1 Query-independent ranking

The Pagerank algorithm (Page et al., 1998) is currently an important part of the ranking function used by the Google search engine. The definition of Pagerank is recursive, stating in simple terms that "a page with high Pagerank is a page referenced by many pages with high Pagerank".

To calculate the Pagerank, each page on the Web is modeled as a state in a system, and each hyperlink as a transition between two states. The Pagerank value of a page is the probability of being in a given page when this system reaches its stationary state.

A good metaphor for understanding this is to imagine a "random surfer", a person who visits pages at random, and upon arrival to each page, chooses an outgoing link uniformly at random from the links in that page. The Pagerank of a page is the fraction of time the random surfer

spends at each page.

This simple system can be modeled by the following equation of a "simplified Pagerank". In this and the following equations, p is a Web page, $\Gamma^-(p)$ is the set of pages pointing to p, and $\Gamma^+(p)$ is the set of pages p points to.

$$Pagerank'(p) = \sum_{x \in \Gamma^-(p)} \frac{Pagerank'(x)}{|\Gamma^+(x)|}$$

However, actual Web graphs include many pages with no out-links, which act as "rank sinks" as they accumulate rank but never distribute it to other pages. In stationary state, only they would have Pagerank > 0. This pages can be removed from the system and their rank computed later. Also, we would like pages not to accumulate ranking by using indirect self-references (self-links are easy to remove), not passing all of their score to other pages. For these reasons, most of the implementations of Pagerank add "random jumps" to each page. These random jumps are hyperlinks from every page to all pages in the collection, including itself, which provide a minimal rank to all the pages as well as a damping effect for self-reference schemes.

In terms of the random surfer model, we can state that when choosing the next step, the random surfer either chooses a page at random from the collection with probability ϵ, or chooses to follow a link from the current page with probability $1 - \epsilon$. This is the model used for calculating Pagerank in practice, and it is described by the following equation:

$$Pagerank(p) = \frac{\epsilon}{N} + (1 - \epsilon) \sum_{x \in \Gamma^-(p)} \frac{Pagerank(x)}{|\Gamma^+(x)|}$$

where N is the number of pages in the collection, and the parameter ϵ is typically between 0.1 and 0.2, based on empirical evidence. Pagerank is a global, static measure of quality of a Web page, very efficient in terms of computation time, as it only has to be calculated once at indexing time and is later used repeatedly at query time.

Note that Pagerank can also be manipulated and in fact there are thousands or millions of Web pages created specifically for the objective of deceiving the ranking function:

> "Among the top 20 URLs in our 100 million page Pagerank calculation using teleportation to random pages, 11 were pornographic, and they appear to have all been achieved using the same form of link manipulation. The specific technique that was used was to create many URLs that all link to a single page, thereby accumulating the Pagerank that every page receives from random teleportation, and concentrating it into a single page of interest." (Eiron et al., 2004)

Another paradigm for ranking pages based on a Markov chain is the absorbing model (Amati et al., 2003). In this model, the original Web graph is transformed adding, for each node, a "clone node" with no out-links. Each clone node p' is only linked from one node in the original graph p. When this system reaches stationary state, only the clone nodes have probabilities greater than zero. The probability of the clone node p' is interpreted as the score of the original node p.

A different paradigm for static ranking on the Web is the network flow model (Tomlin, 2003) for ranking pages. A (sub)graph of the Web is considered as carrying a finite amount of fluid, and edges between nodes are pipes for this fluid. The ranking of a page is related to the maximum amount of flow through a node in this network.

Query-independent ranking summarizes each page on the Web with a single number, or a pair of numbers, but as the creators of Pagerank note, "the importance of a Web page is an inherently subjective matter that depends on readers interests, knowledge and attitudes" (Page et al., 1998); this is why query-dependent ranking is introduced to create ranking functions that are sensitive to user's needs.

3.2.2 Query-dependent ranking

In query-dependent ranking, the starting point is a "neighborhood graph": a set of pages that is expected to be relevant to the given query. This graph is built by starting with a set of k pages containing the query terms; this set can be built using the list of results given by a full-text search engine. This *root set* is augmented by its "neighborhood", that comprises all (or a large sample) of the pages directly pointing to, or directly pointed by, pages in the root set.

Figure 4 depicts the process of creation of the neighborhood set. The idea of limiting the number of pages added to the neighborhood set by following back links to at most a parameter d, that was not part of the original unbounded d proposal, but was introduced later (Bharat and Henzinger, 1998).

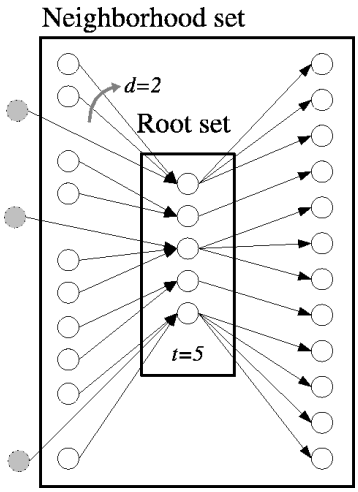

Figure 4: Expansion of the root set with $k = 5$ and $d = 2$. k is the number of pages in the root set, and d is the maximum number of back-links to include in the neighborhood set.

It is customary that when considering links in the neighborhood set, only links in different Web sites are included, as links between pages in the same Web site are usually created by the same authors as the pages themselves, and do not reflect the relative importance of a page for the general

community.

The most linked pages in the neighborhood set are usually not the best candidates (Yuwono and Lee, 1996). Ordering pages by number of in-links performed poorly when compared with pure content-based analysis.

The HITS algorithm (Kleinberg, 1999) is based on considering that relevant pages can be either "authority pages" or "hub pages". An authority page is expected to have relevant content for a subject, and a hub page is expected to have many links to authority pages. These two characteristics have a mutually-reinforcing relationship: a page with high authority is pointed to by many pages with a high "hubness" and vice-versa, as shown in Figure 5.

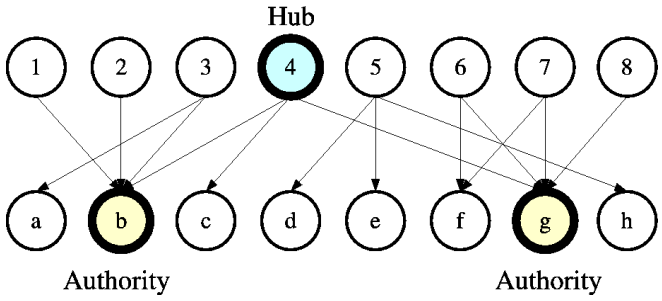

Figure 5: Hubs and authorities in a small graph. Node 4 is the best hub page, as it points to many authorities, and nodes b and g are the best authority pages.

HITS produces two scores for each page, called "authority score" and "hub score", using an iterative method of computation. The algorithm suffers from several drawbacks in its pure form (Bharat and Henzinger, 1998):

(a) Not all the documents in the neighborhood set are about the original topic ("topic drifting").

(b) There are nepotistic, mutually-reinforcing relationships between hosts.

(c) There are many automatically generated links.

Problem (a) is the most important, as while expanding the root set, it is common to include popular pages that are highly-linked, but unrelated to the query topic. The solution is to use analysis of the contents of the documents and/or anchor texts and pruning the neighborhood graph by removing the documents that are too different from the query.

Problems (b) and (c) can be avoided using the following heuristic: if there are k edges from documents on a host to documents in another host, then each edge is given a weight of $1/k$. This gives each document the same amount of influence on the final score, regardless of the number of links in that specific document.

A different approach to query-dependent ranking is topic-sensitive Pagerank (Haveliwala, 2002). In this ranking scheme, scores for each page are pre-computed at indexing time, using an algorithm similar to Pagerank. Each score represents the importance of a page for each topic from a set of pre-defined topics. At query time, the ranking is done using the query to assign weights to the different topic-sensitive Pagerank scores of each page.

4 Crawling the Web

The large volume of the Web implies that any Web crawler can only download a fraction of the existent Web pages within a given time, so it needs to prioritize its downloads. The high rate of change of the Web implies that by the time the crawler is downloading the last pages from a site, it is very likely that new pages have been added to the site, or that pages that have already been updated or even deleted.

Crawling the Web, in a certain way, resembles watching the sky in a clear night: what we see reflects the state of the stars at different times, as the light travels different distances. What a Web crawler gets is not a "snapshot" of the Web, because it does not represents the Web at any given

instant of time (Baeza-Yates and Ribeiro-Neto, 1999). The last pages being crawled are probably very accurately represented, but the first pages that were downloaded have a high probability of have been changed. This idea is depicted in Figure 6.

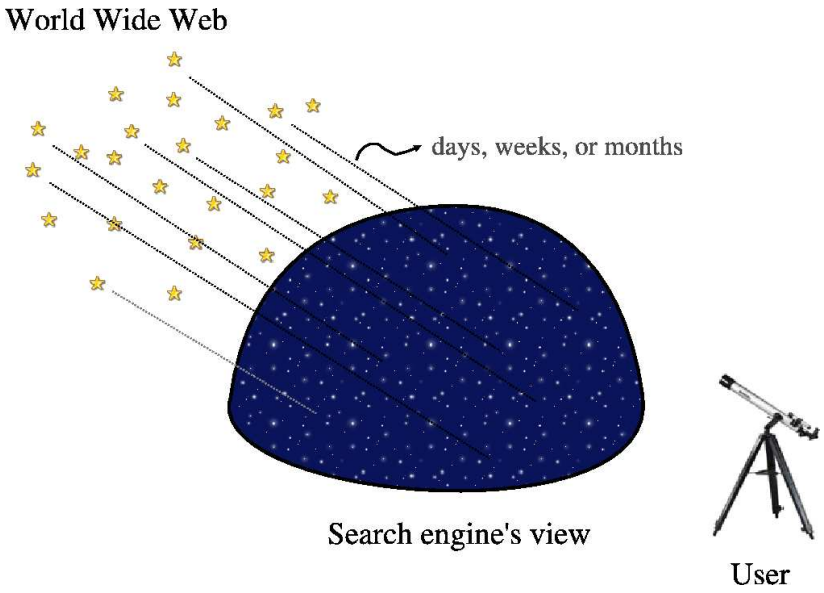

Figure 6: The search engine's view of the Web represents the state of Web pages at different times, as the view of the sky at night presents the state of stars at different times.

A crawler requires a smart scheduling policy for downloading Web pages, and this may become a harder problem in the future, as the amount of information on the Web can potentially grow faster than the available bandwidth for Web crawlers.

> "Given that the bandwidth for conducting crawls is neither infinite nor free it is becoming essential to crawl the Web in a not only scalable, but efficient way if some reasonable measure of quality or freshness is to be maintained." (Edwards et al., 2001)

The behavior of a Web crawler is the outcome of a scheduling policy that is mostly concerned with which pages to download and in which order (selection policy) and how to re-visit pages

(re-visit policy) without overloading Web sites (politeness policy).

4.1 Selection policy

There are several types of Web crawlers. A Web search engine concerned with only one country or one region is called a "vertical search engine" and uses a crawler specifically designed for this purpose. In the case of a vertical crawler or an intranet crawler the problem is easy as the selection policy of pages is mostly related to selecting by a domain name.

On the other end, on a global Web search engine, the selection policy deals mostly with **when to stop** crawling, as the space of Web pages is infinite. In this regard, a usual criteria is link depth, i.e., starting from the home page, follow links up to a certain level (Baeza-Yates and Castillo, 2004).

There is a third kind of selection policy that is used by topical crawlers (for instance: the crawler of a search engine specialized in real estate). In this case, the importance of a page for a crawler can be expressed as a function of the similarity of a page to a given query. This is called "focused crawling" (Chakrabarti et al., 1999). The main problem in focused crawling is that in the context of a Web crawler, we would like to be able to predict the similarity of the text of a given page to the query **before** actually downloading the page. The performance of a focused crawling depends mostly on the richness of links in the specific topic being searched, and a focused crawling usually relies on a general Web search engine for providing starting points.

4.2 Re-visit policy

The Web has a very dynamic nature, and crawling a fraction of the Web can take a long time, usually measured in weeks or months. By the time a Web crawler has finished its crawl, many events could have happened.

¿From the search engine's point of view, there is a cost associated with not detecting an event, and thus having an outdated copy of a resource. The most used cost functions are freshness and age (Cho and Garcia-Molina, 2000).

Freshness This is a binary measure that indicates whether the local copy is accurate or not. The freshness of a page p in the repository at time t is defined as:

$$F_p(t) = \begin{cases} 1 & \text{if } p \text{ is equal to the local copy at time } t \\ 0 & \text{otherwise} \end{cases}$$

Age This is a measure that indicates how outdated the local copy is. The age of a page p in the repository, at time t is defined as:

$$A_p(t) = \begin{cases} 0 & \text{if } p \text{ is not modified at time } t \\ t - \text{modification time of } p & \text{otherwise} \end{cases}$$

The objective of the crawler is to keep the average freshness of pages in its collection as high as possibly, or to keep the average age of pages as low as possible. These objectives are not equivalent: in the first case, the crawler is just concerned with **how many** pages are out-dated, while in the second case, the crawler is concerned with **how old** the local copies of pages are.

Cho and Garcia-Molina (Cho and Garcia-Molina, 2003a) proved the surprising result that, in terms of average freshness, it is better to re-visit pages with a uniform frequency than to re-visit pages with a frequency proportional to their rate of change. The explanation for this result comes from the fact that, when a page changes too often, the crawler will waste time by trying to re-crawl it too fast and still will not be able to keep its copy of the page fresh.

4.3 Politeness policy

The use of Web crawlers is useful for a number of tasks, but comes with a price for the general community. Web crawlers use a lot of network resources and can potentially overload servers with more requests than they can handle.

A partial solution to these problems is the robots exclusion protocol (Koster, 1996), that is a standard for administrators to indicate which parts of their Web servers should not be accessed by robots. As for the interval between accesses to the same Web site, they vary between 20 seconds and 3–4 minutes. It is worth noticing that even when being very polite, and taking all the safeguards to avoid overloading Web servers, some complaints from Web server administrators are received. Brin and Page note that: "... running a crawler which connects to more than half a million servers (...) generates a fair amount of email and phone calls" (Brin and Page, 1998).

4.4 Web crawler architecture

A crawler must have a good crawling strategy, as noted in the previous sections, but it also needs a highly optimized architecture. Building a high-performance crawling system presents a number of challenges related to network efficiency and robustness (Shkapenyuk and Suel, 2002).

Web crawlers are a central part of search engines, and details on their algorithms and architecture are kept as business secrets. When crawler designs are published, there is often an important lack of detail that prevents other from reproducing the work. There are also emerging concerns about "search engine spamming" that prevent major search engines from publishing their ranking algorithms. The typical high-level architecture of Web crawlers is shown in Figure 7.

Some important general-purpose crawlers include the WebCrawler (Pinkerton, 1994), which was used to build the first publicly-available full-text index of the Web, the Internet Archive Crawler

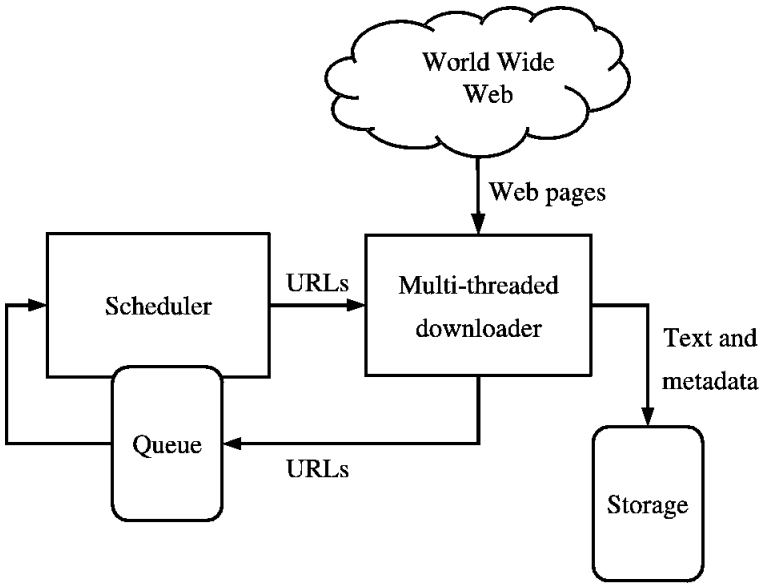

Figure 7: Typical high-level architecture of a Web crawler.

(Burner, 1997), which is used to archive periodic snapshots of a large portion of the Web and the Google Crawler (Brin and Page, 1998) which is described in some detail, but the reference is only about an early version. In addition to these specific crawler architectures, there are general crawler architectures described in (Chakrabarti, 2003; **?**).

5 Web Characterization

5.1 Web size

During the last three years, about 10 million Web sites have appeared yearly, and in November 2004 over 56 million Web sites have been found by Netcraft's Web server survey[1]. As for the number of Web pages, the leading search engine today, Google (`http://www.google.com`, indexed over 8 billion (10^9) pages in November 2004.. This figure can be taken just as a lower bound. Even large

[1]See `http://news.netcraft.com/`

search engines cover only a portion of the publicly available content (Lawrence and Giles, 2000) it is shown that no search engine indexed more than 16% of the Web on 1999.

Currently, most of the content on the Web cannot be found by following links, but is only accessible through query forms. This is called the "Hidden Web" (Raghavan and Garcia-Molina, 2001) and it is believed to hold one or two orders of magnitude more information than the public pages. And, although the amount of information on the Web is certainly finite, Web applications can generate arbitrarily many pages and it can be argued that the number of pages on the Web is potentially infinite (Baeza-Yates and Castillo, 2004).

5.2 Web dynamics

When studying document updates, the data is obtained by repeated access to a subset of pages. In all cases, the results are only an estimation of the actual values because they are obtained by polling for events (changes), not by the resource notifying events, so it is possible that between two crawler accesses a Web page changes more than once.

If changes to a given page occur at independent intervals –if page change is a memory-less process– then it can be modeled as a Poisson process (Brewington et al., 2000). The probability that a search engine's copy of a page is up-to-date at a certain time decreases exponentially if the page is not re-visited. Under this model of page changes, the rate of change of a given page can be estimated based on previous observations (Cho and Garcia-Molina, 2003b), especially if the Web server provides the last modification date of the page every time it is visited.

However, it is worth noticing that most Web page changes exhibit certain periodicity –because most of the updates occur during business hours in the relevant time zone for the studied sample– so the estimators that do not account for this periodicity are more valid in the scales of weeks or

months than on smaller scales.

The methodologies and goals for studies about Web page changes vary widely. Some researchers focus on the lifespan of pages –the time is takes for a page to disappear from the Web– as they are concerned with the availability of Web content. For instance, in the case of scholarly publications, in 4 years about 50% of the links are no longer valid (Spinellis, 2003).

Other studies focus in Web page changes: in a large-scale study of 150 million pages during 10 weeks (Fetterly et al., 2003), 65% of the pages did not change at all, while 30% of the pages had only minor changes and 5% changed substantially. It is also known than the `.com` domain is more dynamic than `.edu` and `.gov` (Cho, 2000) and that highly linked pages change more frequently (Douglis et al., 1997).

5.3 Link structure

The link structure on the Web emerges as the result of collective actions, and as such, it has properties that diverge from a pure random network. "While entirely of human design, the emerging network appears to have more in common with a cell or an ecological system than with a Swiss watch." (Barabási, 2001)

Scale-free networks, as opposed to random networks, are characterized by an uneven distribution of links. These networks are characterized as networks in which the distribution of the number of links $\Gamma(p)$ to a page p follows a power law:

$$Pr(\Gamma(p) = k) \propto k^{-\theta}$$

A scale-free network can be interpreted as a graph in which a few highly-linked nodes that act as "hubs" keeping most of the nodes in the network together, as depicted in Figure 8.

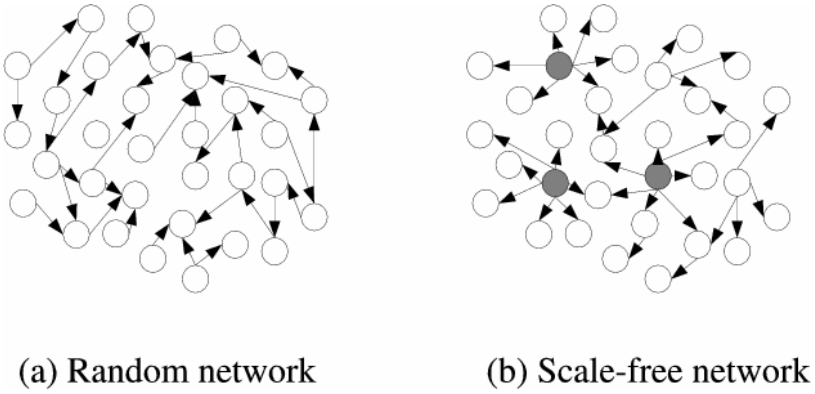

Figure 8: Examples of a random network and a scale-free network, with the major hubs highlighted in the scale-free network. Each graph has the same number of nodes and links. Note that both graphs were chosen to be connected and to look nice on the plane, so they are not entirely random.

On the Internet, the geographical and physical connectivity of nodes form a scale-free network, as well as the network of links between Web pages and of e-mail exchanges. Outside the realm of computer science, scale-free networks appear in several contexts, for inscante: the network of acquaintances, friends and social popularity in human interactions, the graph of citations in scientific publications and the networks of protein interaction in cellular metabolism.

There are certain models of the growth of scale-free networks. The preferential attachment model (Barabási and Albert, 1999), is a "rich get richer" model in which each new Web page creates link to existent Web pages with a probability distribution that is not uniform, but proportional to the current in-degree of existent Web pages. This generates a power-law but the resulting graph differs from the actual Web graph in other properties such as the presence of small, tightly connected communities.

Another generative model is the "copy" model (Kumar et al., 2000), in which new nodes choose an existent node at random and copy a fraction of the links of the existent node. This also generates

a power law. These two models can be combined into a generative model that mixes preferential attachment with a baseline probability of gaining a link (Pennock et al., 2002).

One of the characteristics of the Web that is more difficult to reproduce using generative models is its macroscopic structure. The most complete study of this subject (Broder et al., 2000) focuses on the connectivity of a subset of 200 million Web pages from the Altavista search engine.

The study starts by identifying in the Web graph a single large strongly connected component –all of the pages in this component can reach one another along directed links. They call the larger strongly connected component "MAIN". Starting in MAIN, if we follow links forward we find OUT, and if we follow links backwards we find IN. All of the Web pages with are part of the graph but do not fit neither MAIN, IN, nor OUT are part of a fourth component called TENTACLES. Figure 9 shows all these components.

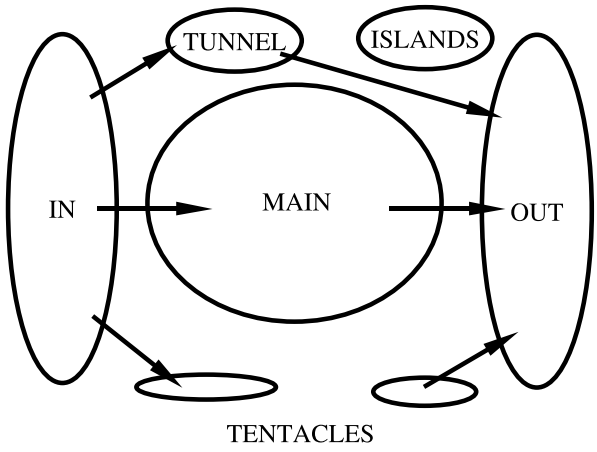

Figure 9: Macroscopic structure of the Web.

Components MAIN, IN, OUT and TENTACLES are roughly the same size, and as components in IN cannot be reached starting from MAIN, it is clear that for Web crawling, a good set of starting URLs is required. There is also a significant fraction of pages that can only be reached if the exact

domain name is known, this is the ISLANDS part of the Web, and many Web sites start by being isolated and later join one of the other components (Baeza-Yates and Poblete, 2004).

6 Research issues

There are plenty of research issues in Web search. They can be divided roughly into two classes: data problems and user problems (**?**).

Among the data problems we have the volume, the fast change, the adversarial nature of the Web (spamming of metadata, content, and links), the diversity of languages (e.g. cross-lingual retrieval), diversity of content (ranging from information and e-commerce sites to blogs and other new forms of Web publishing), web server cooperation schemes for search engines, multimedia retrieval, to name a few.

On the user side, better query languages, user interfaces and result visualizations are needed to cope with the information overload on the client side. The main open problem is to assess the quality of the results. They are usually relevant, but we will never know if we have all the relevants results, or if we have the best ones.

References

Amati, G., Ounis, I., and Plachouras, V. (2003). The dynamic absorbing model for the web. Technical Report TR-2003-137, Department of Computing Science, University of Glasgow.

Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., and Raghavan, S. (2001). Searching the Web. *ACM Transactions on Internet Technology (TOIT)*, 1(1):2–43.

Baeza-Yates, R. (2004). Challenges in the interaction of information retrieval and natural language

processing. In *Proceedings of 5th international conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, volume 2945 of *Lecture Notes in Computer Science*, pages 445–456. Springer.

Baeza-Yates, R. and Castillo, C. (2004). Crawling the infinite Web: five levels are enough. In *Proceedings of the third Workshop on Web Graphs (WAW)*, volume 3243 of *Lecture Notes in Computer Science*, pages 156–167, Rome, Italy. Springer.

Baeza-Yates, R. and Poblete, B. (2004). Dynamics of the Chilean Web structure. In *Proceedings of the 3rd International Workshop on Web Dynamics*, pages 96 – 105, New York, USA.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley.

Barabási, A.-L. (2001). The physics of the web. *PhysicsWeb.ORG, online journal*.

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

Bharat, K. and Henzinger, M. R. (1998). Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104–111, Melbourne, Australia. ACM Press, New York.

Brewington, B., Cybenko, G., Stata, R., Bharat, K., and Maghoul, F. (2000). How dynamic is the web? In *Proceedings of the Ninth Conference on World Wide Web*, pages 257 – 276, Amsterdam, Netherlands.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. (2000). Graph structure in the web: Experiments and models. In *Proceedings of the Ninth Conference on World Wide Web*, pages 309–320, Amsterdam, Netherlands.

Brooks, T. A. (2003). Web search: how the Web has changed information retrieval. *Information Research*, 8(3):(paper no. 154).

Burner, M. (1997). Crawling towards eternity - building an archive of the world wide web. *Web Techniques*, 2(5).

Chakrabarti, S. (2003). *Mining the Web*. Morgan Kaufmann Publishers.

Chakrabarti, S., van den Berg, M., and Dom, B. (1999). Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11–16):1623–1640.

Cho, J. (2000). The evolution of the web and implications for an incremental crawler. In *Proceedings of 26th International Conference on Very Large Databases (VLDB)*, pages 527–534, Cairo, Egypt. Morgan Kaufmann Publishers.

Cho, J. and Garcia-Molina, H. (2000). Synchronizing a database to improve freshness. In *Proceedings of ACM International Conference on Management of Data (SIGMOD)*, pages 117–128, Dallas, Texas, USA.

Cho, J. and Garcia-Molina, H. (2003a). Effective page refresh policies for web crawlers. *ACM Transactions on Database Systems*, 28(4).

Cho, J. and Garcia-Molina, H. (2003b). Estimating frequency of change. *ACM Transactions on Internet Technology*, 3(3).

Davison, B. D. (2000). Topical locality in the web. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval*, pages 272–279. ACM Press.

Douglis, F., Feldmann, A., Krishnamurthy, B., and Mogul, J. C. (1997). Rate of change and other metrics: a live study of the world wide web. In *USENIX Symposium on Internet Technologies and Systems*, pages 147–158, Monterey, California, USA.

Edwards, J., McCurley, K. S., and Tomlin, J. A. (2001). An adaptive model for optimizing performance of an incremental web crawler. In *Proceedings of the Tenth Conference on World Wide Web*, pages 106–113, Hong Kong. Elsevier Science.

Eiron, N., McCurley, K. S., and Tomlin, J. A. (2004). Ranking the web frontier. In *Proceedings of the 13th international conference on World Wide Web*, pages 309–318. ACM Press.

Fetterly, D., Manasse, M., Najork, M., and Wiener, J. L. (2003). A large-scale study of the evolution of web pages. In *Proceedings of the Twelfth Conference on World Wide Web*, pages 669 – 678, Budapest, Hungary. ACM Press.

Haveliwala, T. H. (2002). Topic-sensitive pagerank. In *Proceedings of the Eleventh World Wide Web Conference*, pages 517–526, Honolulu, Hawaii, USA. ACM Press.

Henzinger, M. (2001). Hyperlink analysis for the web. *IEEE Internet Computing*, 5(1):45–50.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

Koster, M. (1996). A standard for robot exclusion. http://www.robotstxt.org/wc/exclusion.html.

Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., and Upfal, E. (2000). Stochastic models for the web graph. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 57–65. IEEE CS Press.

Lawrence, S. and Giles, C. L. (2000). Accessibility of information on the web. *Intelligence*, 11(1):32–39.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). The Pagerank citation algorithm: bringing order to the web. Technical report, Stanford Digital Library Technologies Project.

Pennock, D. M., Flake, G. W., Lawrence, S., Glover, E. J., and Giles, C. L. (2002). Winners don't take all: Characterizing the competition for links on the web. *Proceedings of the National Academy of Sciences*, 99(8):5207–5211.

Pinkerton, B. (1994). Finding what people want: Experiences with the WebCrawler. In *Proceedings of the first World Wide Web Conference*, Geneva, Switzerland.

Raghavan, S. and Garcia-Molina, H. (2001). Crawling the hidden web. In *Proceedings of the Twenty-seventh International Conference on Very Large Databases (VLDB)*, pages 129–138, Rome, Italy. Morgan Kaufmann.

Salton, G. (1971). *The SMART retrieval system - experiments in automatic document processing.* Prentice-Hall.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, 24(5):513–523.

Shkapenyuk, V. and Suel, T. (2002). Design and implementation of a high-performance distributed web crawler. In *Proceedings of the 18th International Conference on Data Engineering (ICDE)*, pages 357 – 368, San Jose, California. IEEE CS Press.

Spinellis, D. (2003). The decay and failures of web references. *Communications of the ACM*, 46(1):71–77.

Tomasic, A. and Garcia-Molina, H. (1993). Performance of inverted indices in shared-nothing distributed text document informatioon retrieval systems. In *Proceedings of the second international conference on Parallel and distributed information systems*, pages 8–17. IEEE Computer Society Press.

Tomlin, J. A. (2003). A new paradigm for ranking pages on the world wide web. In *Proceedings of the Twelfth Conference on World Wide Web*, pages 350–355, Budapest, Hungary. ACM Press.

Yuwono, B. and Lee, D. L. (1996). Search and ranking algorithms for locating resources on the world wide web. In *Proceedings of the twelfth International Conference on Data Engineering (ICDE)*, pages 164–171, Washington, DC, USA. IEEE CS Press.