Recuperação de Informação B

Cap. 13.4.4: Ranking
13.4.5: Crawling the Web
13.4.6: Indices
December 06, 1999
More used models: Boolean or Vector and their variations

Ranking has to be performed without accessing the text, just the index

About ranking algorithms, all information is “top secret”, it is almost impossible to measure recall, as the number of relevant pages can be quite large for simple queries
Yuwono and Lee (1997) proposed three ranking algorithms (plus tf-idf scheme):

- Boolean spread (Classical Boolean plus “simplified link analysis”)
- Most-cited (based only on the terms included in pages having a link to the pages in the answer)
- Vector Spread Activation (Vector space model and spread activation model) (relatively superior, 76%)
- Vector Space Model (TF x IDF) (word distribution statistics) (relatively superior, 75%)
Comparison of these techniques considering 56 queries over a collection of 2400 Web pages indicates that the vector model yields a better recall-precision curve, with an average precision of 75%.

Some of the new ranking algorithms also use hyperlink information.

Important difference between the Web and normal IR databases the number of hyperlinks that point to a page provides a measure of its popularity and quality.

Links in common between pages often indicates a relationship between those pages.
Ranking

- Three examples of ranking techniques based in link analysis
  - WebQuery, which also allows visual browsing of Web pages. WebQuery takes a set of Web pages (for example, the answer to a query) and ranks them based on how connected each Web page is
Ranking

- WebQuery
Kleinberg ranking scheme depends on the query and considers the set of pages S that point to or are pointed by pages in the answer.

- Pages that have many links pointing to them in S are called authorities.
- Pages that have many outgoing links are called hubs.

Better authority pages come from incoming edges from good hubs and better hub pages come from outgoing edges to good authorities.
Ranking

\[ H(p) = \sum_{u \in S | p \rightarrow u} A(u) \]

\[ A(p) = \sum_{v \in S | v \rightarrow p} H(v) \]
PageRank simulates a user navigating randomly in the Web who jumps to a random page with probability $q$ or follows a random hyperlink (on the current page) with probability $1 - a$

This process can be modeled with a Markov chain, from where the stationary probability of being in each page can be computed.

Let $C(a)$ be the number of outgoing links of page $a$ and suppose that page $a$ is pointed to by pages $p_1$ to $p_n$.
PageRank: Bringing Order to the Web

Your browser does not support frames. Try Internet Explorer
3.0 or later or Netscape Navigator 2.0 or later.

John Reece’s Home Page
Internet search, retrieval, resource discovery and knowledge management is my game... --
John Reece Contents Resume/Career Information Search Engine Links Ancient Computer...
Crawling the web

- Start with a set of URLs and from there extract other URLs which are followed recursively in a breadth-first or depth-first fashion.
- Search engines allow users to submit top Web sites that will be added to the URL set
  - A variation is to start with a set of populars URLs, because we can expect that they have information frequently requested
Crawling the Web

- Both cases work well for one crawler, but it is difficult to coordinate several crawlers to avoid visiting the same page more than once.
- Another technique is to partition the Web using country codes or Internet names, and assign one or more robots to each partition, and explore each partition exhaustively.
- Considering how the Web is traversed, the index of a search engine can be thought of as analogous to the stars in an sky. What we see has never existed, as the light has traveled different distances to reach our eye.
Crawling the web

- Similarly, Web pages referenced in an index were also explored at different dates and they may not exist any more.
- How fresh are the Web pages referenced in an index? The pages will be from one day to two months old. For that reason, most search engines show in the answer the date when the page was indexed.
- The percentage of invalid links stored in search engines vary from 2 to 9%.
Crawling the Web

- User submitted pages are usually crawled after a few days or weeks.
- Some engines traverse the whole Web site, while others select just a sample of pages or pages up to a certain depth. Non-submitted pages will wait from weeks up to a couple of months to be detected.
- There are some engines that learn the change frequency of a page and visit it accordingly.
- The current fastest crawlers are able to traverse up to 10 million Web pages per day.
Crawling the Web

- The order in which the URLs are traversed is important
  - Using a breadth first policy, we first look at all the pages linked by the current page, and so on. This matches well Web sites that are structured by related topics. On the other hand, the coverage will be wide but shallow and a Web server can be bombarded with many rapid requests
  - In the depth first case, we follow the first link of a page and we do the same on that page until we cannot go deeper, returning recursively
  - Good ordering schemes can make a difference if crawling better pages first (PageRank)
Crawling the Web

- Due to the fact that robots can overwhelm a server with rapid requests and can use significant Internet bandwidth a set of guidelines for robot behavior has been developed.
- Crawlers can also have problems with HTML pages that use frames or image maps. In addition, dynamically generated pages cannot be indexed as well as password protected pages.
Indices

- Most indices use variants of the inverted file
  - An inverted file is a list of sorted words (vocabulary), each one having a set of pointers to the pages where it occurs
  - Some search engines use elimination of stopwords to reduce the size of the index. Normalization operations may include removal of punctuation and multiple spaces, etc
  - To give the user some idea about each document retrieved, the index is complemented with a short description of each Web page
Indices

- Assuming that 500 bytes are required to store the URL and the description of each Web page, we need 50 Gb to store the description for 100 million pages.
- As the user initially receives only a subset of the complete answer to each query, the search engine usually keeps the whole answer set in memory, to avoid having to recompute it if the user asks for more documents.
Indices

- Indexing techniques can reduce the size of an inverted file to about 30% of the size of the text (less if stopwords are used). For 100 million pages, this implies about 15 Gb of disk space.
- A query is answered by doing a binary search on the sorted list of words of the inverted file.
- Searching multiple words, the results have to be combined to generate the final answer.
- Problem: frequency of the word.
Indices

- Inverted files can also point to the actual occurrences of a word within a document in space for the Web (too costly), because each pointer has to specify a page and a position inside the page (word numbers can be used instead of actual bytes).

- Having the positions of the words in a page, we can answer phrase searches or proximity queries by finding words that are near each other in a page.

- Currently, some search engines are providing phrase searches, but the actual implementation is not known.
Indices

- Finding words which start with a given prefix requires two binary searches in the sorted list of words.

- More complex searches can be performed by doing a sequential scan over the vocabulary.

- The best sequential algorithms for this type of query can search around 20 Mb of text stored in RAM in one second (5 Mb is more or less the vocabulary size for 1 Gb of text).

- For the Web this is still too slow but not impossible. Using Heaps' law and assuming $\beta$ (beta) = 0.7 for the Web, the vocabulary size for 1 Tb is 630 Mb which implies a searching time of half a minute.
Indices

- Pointing to pages or to word positions is an indication of the granularity of the index
- The index can be less dense if we point to logical blocks instead of pages
  - Reduce the variance of the different document sizes, by making all blocks roughly the same size
    - Reduces the size of the pointers (because there are fewer blocks than documents)
    - Reduces the number of pointers (because words have locality of reference)
Indices

- This idea was used in Glimpse
- Queries are resolved as for inverted files, obtaining a list of blocks that are then searched sequentially (exact sequential search can be done over 30 Mb per second in RAM)
- Glimpse originally used only 256 blocks, efficient up to 200 Mb for searching words that were not too frequent, obtaining an index of only 200Mb of the text
- Not used in the Web because sequential search cannot be afforded, as it implies a network access. However, in a distributed architecture where the index is also distributed, logical blocks make sense
Indices

Announcements and News

Overview

WebGlimpse adds search capabilities to your WWW site automatically and easily. It attaches a small search box to the bottom of every HTML page, and allows the search to cover the neighborhood of that page or the whole site. With WebGlimpse there is no need to construct separate search pages, and no need to interrupt the users from their browsing. Pages remain unchanged except for the extra search capabilities. It is even possible for the search to efficiently cover pages linked from your pages. (WebGlimpse will collect such remote pages to your disk and index them.) Installation, customization (e.g., deciding which pages to collect and which ones to index), and maintenance are easy.
Nowadays search engines use, basically, Boolean or Vector models and their variations.

Link Analysis Techniques seem to be the “next generation” of the search engines.

“Bots” will be “Pokemons”.

Indices: Compression and distributed architecture are keys.
References

- GOOGLE, www.google.com
- TODOBR, www.todobr.com.br
- WEBGLIMPSE, glimpse.cs.arizona.edu/webglimpse/