Web Search Engines

Chapter 27, Part C

Based on Larson and Hearst's slides at UC-Berkeley

http://www.sims.berkeley.edu/courses/is202/f00/

Search Engine Characteristics

- Unedited – anyone can enter content
  - Quality issues; Spam
- Varied information types
  - Phone book, brochures, catalogs, dissertations, news reports, weather, all in one place!
- Different kinds of users
  - Lexis-Nexis: Paying, professional searchers
  - Online catalogs: Scholars searching scholarly literature
  - Web: Every type of person with every type of goal
- Scale
  - Hundreds of millions of searches/day; billions of docs

Web Search Queries

- Web search queries are short:
  - ~2.4 words on average (Aug 2000)
  - Has increased, was 1.7 (~1997)
- User Expectations:
  - Many say "The first item shown should be what I want to see!"
  - This works if the user has the most popular/common notion in mind, not otherwise.

Directories vs. Search Engines

- Directories
  - Hand-selected sites
  - Search over the contents of the ***descriptions*** of the pages
  - Organized in advance into categories
- Search Engines
  - All pages in all sites
  - Search over the contents of the ***pages themselves***
  - Organized in response to a query by relevance rankings or other scores

What about Ranking?

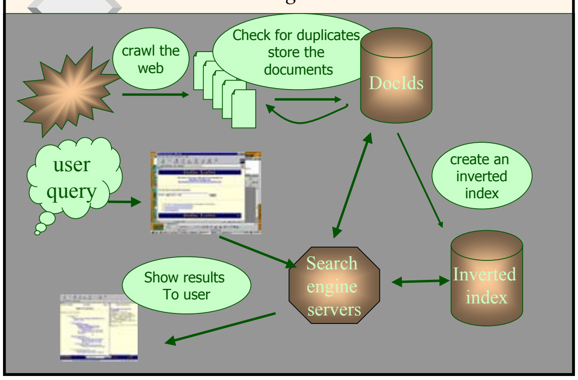
- Lots of variation here
  - Often messy; details proprietary and fluctuating
- Combining subsets of:
  - IR-style relevance: Based on term frequencies, proximities, position (e.g., in title), font, etc.
  - Popularity information
  - Link analysis information
- Most use a variant of vector space ranking to combine these. Here's how it might work:
  - Make a vector of weights for each feature
  - Multiply this by the counts for each feature

Relevance: Going Beyond IR

- Page "popularity" (e.g., DirectHit)
  - Frequently visited pages (in general)
  - Frequently visited pages as a result of a query
- Link "co-citation" (e.g., Google)
  - Which sites are linked to by other sites?
  - Draws upon sociology research on bibliographic citations to identify "authoritative sources"
  - Discussed further in Google case study

Web Search Architecture

Standard Web Search Engine Architecture



Inverted Indexes the IR Way

How Inverted Files Are Created

- Periodically rebuilt, static otherwise.
- Documents are parsed to extract tokens. These are saved with the Document ID.

Doc 1

Now is the time for all good men to come to the aid of their country

Doc 2

It was a dark and stormy night in the country manor. The time was past midnight

Term	Doc #
now	1
is	1
the	1
time	1
for	1
all	1
good	1
men	1
to	1
come	1
to	1
the	1
aid	1
of	1
their	1
country	1
it	2
was	2
a	2
dark	2
and	2
stormy	2
night	2
in	2
the	2
country	2
manor	2
the	2
time	2
was	2
past	2
midnight	2

How Inverted Files are Created

- After all documents have been parsed the inverted file is sorted alphabetically.

Term	Doc #
now	1
is	1
the	1
time	1
for	1
all	1
good	1
men	1
to	1
come	1
to	1
the	1
aid	1
of	1
their	1
country	1
it	2
was	2
a	2
dark	2
and	2
stormy	2
night	2
in	2
the	2
country	2
manor	2
the	2
time	2
was	2
past	2
midnight	2

Term	Doc #
a	2
aid	1
all	1
and	2
come	1
country	1
country	2
dark	2
for	1
good	1
in	2
is	1
it	2
manor	2
men	1
midnight	2
night	2
now	1
of	1
past	2
stormy	2
the	1
the	1
the	2
the	2
their	1
time	1
time	2
to	1
to	1
was	2
was	2

How Inverted Files are Created

- Multiple term entries for a single document are merged.
- Within-document term frequency information is compiled.

Term	Doc #
a	2
aid	1
all	1
and	2
come	1
country	1
country	2
dark	2
for	1
good	1
in	2
is	1
it	2
manor	2
men	1
midnight	2
night	2
now	1
of	1
past	2
stormy	2
the	1
the	1
the	2
the	2
their	1
time	1
time	2
to	1
to	1
was	2
was	2

Term	Doc #	Freq
a	2	1
aid	1	1
all	1	1
and	2	1
come	1	1
country	1	1
country	2	1
dark	2	1
for	1	1
good	1	1
in	2	1
is	1	1
it	2	1
manor	2	1
men	1	1
midnight	2	1
night	2	1
now	1	1
of	1	1
past	2	1
stormy	2	1
the	1	2
the	2	2
their	1	1
time	1	1
time	2	1
to	1	2
was	2	2

How Inverted Files are Created

- Finally, the file can be split into
  - A Dictionary or Lexicon file

  and
  - A Postings file

How Inverted Files are Created

Term	Doc #	Freq
a	2	1
aid	1	1
all	1	1
and	2	1
come	1	1
country	1	1
country	2	1
dark	2	1
for	1	1
good	1	1
in	2	1
is	1	1
it	2	1
manor	2	1
men	1	1
midnight	2	1
night	2	1
now	1	1
of	1	1
past	2	1
stormy	2	1
the	1	2
the	2	2
their	1	1
time	1	1
time	2	1
to	1	2
was	2	2

Dictionary/Lexicon

Term	N docs	Tot Freq
a	1	1
aid	1	1
all	1	1
and	1	1
come	1	1
country	2	2
dark	1	1
for	1	1
good	1	1
in	1	1
is	1	1
it	1	1
manor	1	1
men	1	1
midnight	1	1
night	1	1
now	1	1
of	1	1
past	1	1
stormy	1	1
the	2	4
their	1	1
time	2	2
to	1	2
was	1	2

Postings

Doc #	Freq
2	1
1	1
1	1
2	1
1	1
1	1
2	1
2	1
1	1
1	1
2	1
1	1
2	1
2	1
1	1
2	1
2	1
1	1
1	1
2	1
2	1
1	2
2	2
1	1
1	1
2	1
1	2
2	2

Inverted indexes

- Permit fast search for individual terms
- For each term, you get a list consisting of:
  - document ID
  - frequency of term in doc (optional)
  - position of term in doc (optional)
- These lists can be used to solve Boolean queries:
  - country -> d1, d2
  - manor -> d2
  - country AND manor -> d2
- Also used for statistical ranking algorithms

Inverted Indexes for Web Search Engines

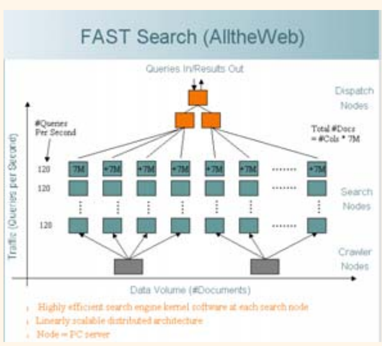
- Inverted indexes are still used, even though the web is so huge.
- Some systems partition the indexes across different machines. Each machine handles different parts of the data.
- Other systems duplicate the data across many machines; queries are distributed among the machines.
- Most do a combination of these.

In this example, the data for the pages is partitioned across machines. Additionally, each partition is allocated multiple machines to handle the queries.

Each row can handle 120 queries per second

Each column can handle 7M pages

To handle more queries, add another row.



From description of the FAST search engine, by Knut Risvik http://www.infonortics.com/searchengines/sh00/risvik_files/frame.htm

Cascading Allocation of CPUs

- A variation on this that produces a cost-savings:
  - Put high-quality/common pages on many machines
  - Put lower quality/less common pages on fewer machines
  - Query goes to high quality machines first
  - If no hits found there, go to other machines

Web Crawling

Web Crawlers

- How do the web search engines get all of the items they index?
- Main idea:
  - Start with known sites
  - Record information for these sites
  - Follow the links from each site
  - Record information found at new sites
  - Repeat

Web Crawling Algorithm

- More precisely:
  - Put a set of known sites on a queue
  - Repeat the following until the queue is empty:
    - Take the first page off of the queue
    - If this page has not yet been processed:
      - Record the information found on this page
        - Positions of words, links going out, etc
      - Add each link on the current page to the queue
      - Record that this page has been processed
- Rule-of-thumb: 1 doc per minute per crawling server

Web Crawling Issues

- Keep out signs
  - A file called norobots.txt lists "off-limits" directories
  - Freshness: Figure out which pages change often, and recrawl these often.
- Duplicates, virtual hosts, etc.
  - Convert page contents with a hash function
  - Compare new pages to the hash table
- Lots of problems
  - Server unavailable; incorrect html; missing links; attempts to "fool" search engine by giving crawler a version of the page with lots of spurious terms added ...
- Web crawling is *difficult* to do robustly!

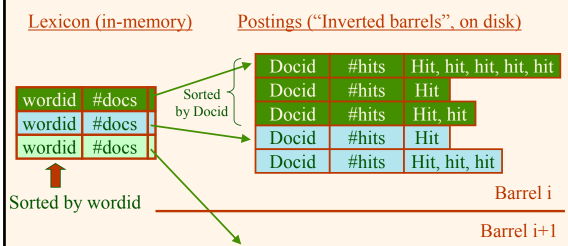
Google: A Case Study

Google's Indexing

- The *Indexer* converts each doc into a collection of "hit lists" and puts these into "barrels", sorted by docID. It also creates a database of "links".
  - Hit: <wordID, position in doc, font info, hit type>
  - Hit type: Plain or fancy.
  - Fancy hit: Occurs in URL, title, anchor text, metatag.
  - Optimized representation of hits (2 bytes each).
- *Sorter* sorts each barrel by wordID to create the inverted index. It also creates a lexicon file.
  - Lexicon: <wordID, offset into inverted index>
  - Lexicon is mostly cached in-memory

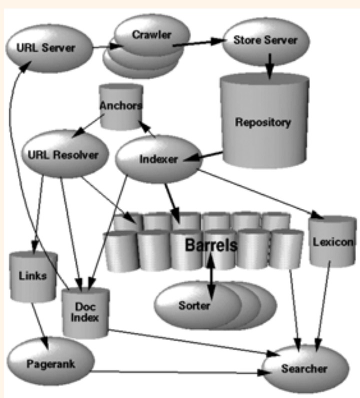
Google's Inverted Index

Each "barrel" contains postings for a range of wordids.



Google

- Sorted barrels = inverted index
- Pagerank computed from link structure; combined with IR rank
- IR rank depends on TF, type of "hit", hit proximity, etc.
- Billion documents
- Hundred million queries a day
    - AND queries



Link Analysis for Ranking Pages

- Assumption: If the pages pointing to this page are good, then this is also a good page.
    - References: Kleinberg 98, Page et al. 98
- Draws upon earlier research in sociology and bibliometrics.
    - Kleinberg's model includes "authorities" (highly referenced pages) and "hubs" (pages containing good reference lists).
    - Google model is a version with no hubs, and is closely related to work on influence weights by Pinski-Narin (1976).

Link Analysis for Ranking Pages

- Why does this work?
  - The official Toyota site will be linked to by lots of other official (or high-quality) sites
  - The best Toyota fan-club site probably also has many links pointing to it
  - Less high-quality sites do not have as many high-quality sites linking to them

PageRank

- Let A1, A2, …, An be the pages that point to page A. Let C(P) be the # links out of page P. The PageRank (PR) of page A is defined as:

$$PR(A) = (1-d) + d \, (PR(A1)/C(A1) + \ldots + PR(An)/C(An))$$

- PageRank is principal eigenvector of the link matrix of the web.
- Can be computed as the fixpoint of the above equation.

PageRank: User Model

- PageRanks form a probability distribution over web pages: sum of all pages' ranks is one.
- User model: "Random surfer" selects a page, keeps clicking links (never "back"), until "bored": then randomly selects another page and continues.
  - PageRank(A) is the probability that such a user visits A
  - d is the probability of getting bored at a page
- Google computes relevance of a page for a given search by first computing an IR relevance and then modifying that by taking into account PageRank for the top pages.

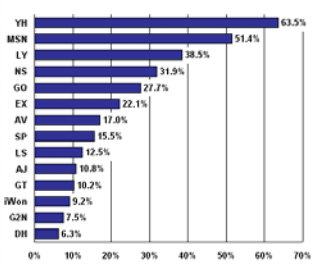
Web Search Statistics

Searches per Day

Service	Searches Per Day	As Of/Notes
AltaVista	50 million	9/00 (as reported to me by AltaVista, for its site and queries through partners)
Inktomi	47 million	4/00 (still reflects queries from Yahoo, which will be handled by Google from July 2000).
Google	40 million	8/00 (14 million of these are at Google.com, 15 million are probably generated through Google's partnership with Yahoo, and the remainder come through Google partner sites, such as Netscape Search)
GoTo	5 million	4/00 (as reported by GoTo to a reader, who forwarded the information to me. Includes queries through affiliates and partners).
Ask Jeeves	4 million	3/00
Voila	1.5 million	1/00 (as reported to me by Voila, for its entire network of sites)

Web Search Engine Visits

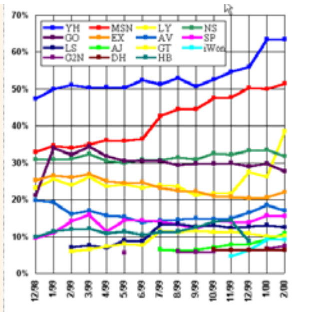
Below is a look at the latest Media Metrix's ratings. They show audience reach, which is the percentage of web surfers estimated to have visited each search engine during the month. Because a web surfer may visit more than one service, the combined totals exceed percent.



KEY: YH=Yahoo, MSN=MSN, LY=Lycos, NS=Netscape, GO=Go (Infoseek), EX=Excite, AV=AltaVista, SP=Snap, LS=LookSmart, AJ=AskJeeves, GT=GoTo, iWon=iWon, G2N=Go2Net, DH=Direct Hit. *Also use this key for charts below. See the* Major Search Engines *page for links to these services.*

The chart below shows audience reach over time. Data for some of the less-popular services is not made available each month, which is why there are gaps on the chart. An analysis of changes in the past few months is shown below the chart.

Percentage of web users who visit the site shown



Search Engine Size (July 2000)

Obscure Terms

The first test checked on how well each search engine did in finding four obscure terms. By obscure, I mean that these were words unusual enough that no search engine found more than 100 matches. A separate page listed at the end of this article shows exactly what terms were used and the scoring methodology. The chart below summarizes the test. Search engines are listed in order of performance, with the best at the top of the list.

Search Engine	Reported Size	Expected Score	Actual Score	Rank
Google	560	1.0	1.0	1
FAST	340	2.0	1.8	2
Northern Light	265	3.0	2.3	3
HotBot	110	4.0	2.3	3
iWon	110	4.0	2.3	3
AltaVista	350	2.0	2.5	4
Yahoo-Google	560	1.0	3.0	5
Excite	250	3.0	3.0	5
Yahoo-Inktomi	110	4.0	4.3	6

The first column shows you how many millions of pages each search engine claims to have indexed. The "Expected Score" column is based on this. You

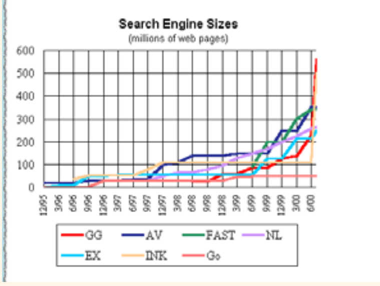
Does size matter? You can't access many hits anyhow.

For the curious, here's a list of the total number of results you can possibly recover from the search engines tested:

Search Engine	Max. Results
FAST	4,010
AltaVista	1,000
Excite	1,000
Google	1,000
HotBot	1,000
iWon	1,000
Yahoo "Web Pages"	199
Northern Light	couldn't determine

Increasing numbers of indexed pages, self-reported

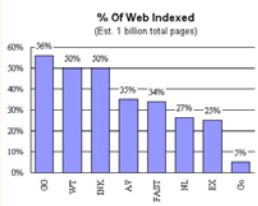
Search Engine Sizes Over Time



Web Coverage

Coverage Of The Web

In Feb. 2000, a joint study published by Inktomi and the NEC Research Institute estimated that there were 1 billion indexable pages on the web. The chart below uses this estimate as a basis for showing the percentage of the web currently covered by each search engine's full text index.



NOTE: This chart assumes some Inktomi partners will go live with its new index in July 2000 and represents greater coverage on those partners. It does not show the extended coverage that Google's link analysis gives that search engine. Were that to be included, Google would actually cover more pages that this study estimated to exist earlier this year.

Size Growth

	Servers	Unique (IP based)	Total pages
March 2000 (ATW Crawl)	20,7M	3,61M	850,1 M
Nature 1999		2.8M	800M (est)
Nov. 1995 - OpenText		223.851	11,4M

Information is also duplicated
- AllTheWeb.com has 340M pages from 850M crawled
- Trend is increasing number of duplicates
- (Duplicate hostnames, duplicate servers, duplicate docs)

- Web estimate should thus be less than 800M documents of unique data.
- Crawling one doc per minute from a server gives an upper bound for freshness (> 1 week)

From description of the FAST search engine, by Knut Risvik
http://www.infonortics.com/searchengines/sh00/risvik_files/frame.htm

Directory sizes

Directory Sizes

Directories are usually human-compiled guides to the web, where sites are organized by category. The chart below compares the size of directories at various services, along with other key data. A ? symbol indicates where information is not known or hasn't been released.

Service	Type	Editors	Cats	Links...	As Of
Open Directory	D	28,000	304,000	2 million	8/00
LookSmart	D	200	200,000	2 million	8/00
Yahoo	D	100+	?	1.5 to 1.8 million	8/00
NBCi (Snap)	D	30-50	70,000	1 million+	8/00
Go (Infoseek)	SE	10,000	50,000	500,000+	1/00
AskJeeves	AS	30	n/a	7 million answers	11/98
AltaVista	SE	See LookSmart			
Excite	SE	See LookSmart			
HotBot	SE	See Open Directory			
Lycos	D	See Open Directory			
MSN Search	SE	See LookSmart			
Netscape	SE	See Open Directory			