Searching the World-Wide Web

Lecture 13
Challenges for Web Search

The World-Wide Web is...

- Distributed
- Volatile
- Huge
- Unstructured
- Redundant
- Of variable quality
- Heterogeneous
- Multilingual

In an information system such as this,

- How should a user specify a query?
- How should he understand the results?
Measuring the Web

- Between Fall 1998 and Summer 1999...
  - 40M computers connected to the Internet
  - 2.4-3M web servers
  - >200 countries, >100 languages
  - 200-350M web pages
    - 2-5Kb, 5-15 hyperlinks
    - Most links are local
    - Most pages not pointed to by external servers
  - Formats: HTML, GIF, JPEG, ASCII, Postscript
    - Images average 14Kb
  - 5Kb * 300M = 1.5 terabytes of text on the Web
Growth of the Web

Number of web servers

- 1997
- 1998
- 1999
- 2000
In 2002...

- 162M hosts on the Internet
  - (July 2002, ISC Internet Domain Survey)
- 36M web servers (surveyed?)
  - (Sep 2002, Netcraft)
- Not many recent peer-reviewed surveys
- Growth may be much faster since 2000
The Internet Archive

- http://www.archive.org/
- Crawls from Alexa and Compaq
- 4 billion pages (40TB) in 2001
- In 2002, 100TB and growing at 12TB/month

- Access
  - The Wayback Machine
  - Researcher access via remote login
Definitions from Graph Theory

- **Graph**: set of nodes and edges between them
  - graphs can be undirected or directed
  - **In-degree**: # edges pointing to a node
  - **Out-degree**: # edges pointing out of a node

- **Diameter**
  - Maximum over all ordered pairs \((u,v)\) of the shortest path from \(u\) to \(v\)

- **Connected Component**
  - a set of nodes in an undirected graph which are reachable from each other
  - **Strongly Connected Component (SCC)**: directed
Power Laws on the Web

- Power Law distributions
  - $P(i) \propto 1/i^k$, for small positive values of $k$
  - Zipf’s Law: a power law for ranks
- Power laws describe many things...
  - vocabulary, economics, sociological models, nucleotide sequences
- Including web phenomena
  - access statistics
  - # times users at a single site access particular pages
  - in/out-degree of web pages
“Graph Structure in the Web”

- Broder et al (2000), WWW9
- large-scale graph analysis of the Web
- two crawls from AltaVista
  - May 99: 203M pages, 1.5B links
  - Oct 99: 271M pages, 2.1B links
- Built on previous web characterizations
  - # links pointing to a page follows a power law
  - most pairs of pages separated by a handful of links (about 20)
Results of Broder et al

- Fraction of pages with in-degree $i \propto 1/i^{2.1}$
  - resembles other, smaller studies
  - small webs resemble large webs (fractal)
- Sizes of connected components also follow a power law
- Largest WCC 91%, Largest SCC 26%
- Examined connectivity of the web using breadth-first search with random starting points.
“Bow Tie” model of the Web

IN
44 Million nodes

SCC
56 Million nodes

OUT
44 Million nodes

Tendrils
44 Million nodes

Tubes
Disconnected components
Paths and Connectivity

- Diameter of SCC is at least 28
  - whole web diameter is over 500
- Not all node pairs are connected
  - For random \((u,v)\), \(P(\text{path}(u,v)) = 0.24\)
  - If a directed path exists, average length = 16
  - Undirected paths, length = 6
- But the WWW in general is well-connected
  - Even if nodes with in-degree > 5 are removed, it still contains a weak component of \(~59M\) nodes.
Connectivity Server

- A fast, high-performance link database
- Input: a web crawl
- Creates database of hosts and URLs with all in-links and out-links
  - includes non-crawled URLs references more than five times
  - 10 bytes/URL, 3.4 bytes/link
- 465MHz Compaq Alpha server, 12GB RAM
- Each crawl fits in 9.5GB of disk
Connectivity Server Architecture

- Two lists per URL
  - inlist: pointers to URL
  - outlist: pointers from
- Heavy use of compression
  - front coding for URLs
  - integer coding for pointers
The “Indexable Web”

- Lawrence and Giles (1998)
- Estimated search engine coverage by carefully analyzing query results
- Lower bound on “indexable web”: 320M pages
- Search engines index a small fraction of this
  - Their study found HotBot covered 34%, followed by AltaVista (28%), Northern Light (20%), Excite (14%), Infoseek (10%), and Lycos (3%)
Searching the Web

- Collection is immense (multi-Terabyte)
  - queries must be answered without accessing the source text
  - alternative: store the text (a la Google)
    - It should be possible to decide what to store
    - Only keep the best pages?
  - alternative: search through the network
    - Too slow for “pure” searching
    - Might be optimized if we could search “best-first”
Centralized Search Engines

- **Crawler**
  - Fetches web pages
  - Culls links
  - Prioritizes links (usually BFS variant)
- **Indexed at main server**
AltaVista Architecture

- Circa 1998
  - 20 multiprocessor machines
  - 130 GB RAM, 500 GB disk (probably low)
- Query engine uses 75% of resources
- O($100M) in hardware costs
Google

- **Full-text index**
  - terms sorted into barrels for merging
- **Link database**
  - URLs, in/out links
- **Parallel crawl approach**
  - 100 pages/sec
Distributed Search Engines

- **Idea:** coordinate among several web servers
- **Harvest:** gatherers and brokers
  - Gatherer collects and extracts information from one or more web servers
  - Brokers provide indexing and query interface
    - Receive info from one or more Gatherers
    - Updates indices
    - Can also filter information and send to other brokers
  - Also features caching and replication agents
CARROT

- Cooperative Agent-based Routing and Retrieval of Text
- Individual search engines manage their own collections
- Broker agents gather metadata from the SEs that describe their collection
  - e.g. a centroid, or vector of document freqs
- Broker routes an incoming query based on similarity to metadata
Web Search Interfaces

- Most query interfaces are spare
  - Implicit AND or OR among search terms
  - Users don’t know logical view of text
  - Most engines provide an “advanced” search feature
    - Boolean expressions, phrases, proximity operators, wildcard globs, regular expressions
  - Results pages also don’t give much information
User query behavior

<table>
<thead>
<tr>
<th>Measure</th>
<th>Average</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td># words</td>
<td>2.35</td>
<td>0-393</td>
</tr>
<tr>
<td># operators</td>
<td>0.41</td>
<td>0-958</td>
</tr>
<tr>
<td>Repetitions of each query</td>
<td>3.97</td>
<td>1-1.5 million</td>
</tr>
<tr>
<td>Queries/user session</td>
<td>2.02</td>
<td>1-173,325</td>
</tr>
<tr>
<td>Results screens/query</td>
<td>1.39</td>
<td>1-78,496</td>
</tr>
</tbody>
</table>

- 25% of users query with a single word
- 15% restrict to a prespecified topic
- 80% don’t modify the query after first retrieval
- 85% only look at first results page
- 64% of queries are unique
Ranking Web Pages

- Traditional models
  - vector space, probabilistic, etc.
  - operate on text only
- Hyperlink models
  - link structure, anchor text
- Hard to assess performance of engines
  - proprietary algorithms
  - complicated engineering
  - but in general they are using known ideas
HITS Algorithm (Kleinberg 97)

- Hypertext Induced Topic Search
- How to identify good pages?
  - Authoritative pages are pointed to by many other pages
  - Hub pages point to many pages
- Identifies good hubs and authorities
- Recommends those as best results.
HITS Algorithm (1)

- Find a focused subgraph $S\sigma$ of the web
  - Should be relatively small
  - Should be rich in pages relevant to the user’s query
  - Should contain many good authorities
- To make the focused subgraph:
  - Fetch top $t$ pages from a textual engine: $R\sigma$
  - Expand $R\sigma$ with
    - all pages pointed to by a page in $R\sigma$
    - some pages which point to pages in $R\sigma$ (max $d$ per page)
    - don’t add pages with URLs within same domain name
  - Return as $S\sigma$
Finding Hubs and Authorities

- Now subgraph contains
  - authorities pointed to by initial ranked list
  - good connectivity among results
- How to determine authorities?
  - Simple: order by in-degree
  - Confuses authorities with universally popular pages (large in-degree, but lack relevance to topic)
Refining the Authority Concept

- Sets of authorities on a topic have
  - high in-degree for all authorities
  - significant overlap in the sets of pages that point to them
- These hubs point to multiple relevant authorities
- Mutually reinforcing relationship
  - a good hub points to many good authorities
  - a good authority is a page pointed to by many good hubs
HITS Algorithm (2)

- $H(p) = \text{hub value of node } p$
- $A(p) = \text{authority value of node } p$
  - Initialize $H(p)$ and $A(p)$ to $(1,1,1,...,1)$

\[
A(p) = \sum_{v \in S \mid v \rightarrow p} H(v)
\]

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

normalize $A(p)$ and $H(p)$ after each iteration
Convergence of HITS

- Typically, 20 iterations is sufficient for the largest elements of $H(p)$ and $A(p)$ to be stable
- If $M$ is the adjacency matrix of subgraph
  - $H(p)$ and $A(p)$ converge to the principal eigenvectors of $MM^T$ and $M^TM$, respectively
  - These are also the first columns of $U$ and $V$ from the singular value decomposition of $M$
HITS example

- (java) Authorities

  0.328  http://www.gamelan.com/  Gamelan
  0.251  http://java.sun.com/  JavaSoft home
  0.190  http://www.digitalfocus.com/digitalfocus/faq/howdoi.html  The Java Developer: HowDoI
  0.190  http://lightyear.ncsa.uiuc.edu/~srp/java/javabooks.html  The Java Book Pages
  0.183  http://sunsite.unc.edu/javafaq/javafaq.html  comp.lang.java FAQ
HITS example (2)

• (“search engines”) Authorities
  .346  http://www.yahoo.com/ Yahoo!
  .291  http://www.excite.com/ Excite
  .239  http://www.mckinley.com/ Welcome to Magellan!
  .231  http://www.lycos.com/ Lycos Home Page
  .231  http://www.altavista.digital.com/ AltaVista: Main Page

• Can also be used to find “similar” pages
  • “Find top t pages pointing to p.”
PageRank (Brin and Page, 98)

- Consider a user browsing randomly
  - Will follow a random link on a page with uniform chance \((1-q)\)
  - May get bored, jump to an unlinked page \((q)\)
  - Never uses the “back” button
- Similar to a Markov chain
  - can use to compute the probability of browsing to any page.
PageRank formula

- \( C(a) = \) out-degree of page \( a \)
- \( p_1...p_n \) – pages pointing to page \( a \)
- \( PR(a) = q + (1-q)\sum_{i=1..n} PR(p_i)/C(p_i) \)
  - compute iteratively as in HITS
  - precomputed over all pages in the index
  - \( q \) is typically 0.15
  - converges to principal eigenvector of link matrix
- Underlying ranking formula used by Google
What are the implications of...

The World-Wide Web is...
- Distributed
- Volatile
- Huge
- Unstructured
- Redundant
- Of variable quality
- Heterogeneous
- Multilingual

In an information system such as this,
- How should a user specify a query?
- How should he understand the results?