SEARCH ENGINE INSIDE OUT

From Technical Views

萧碧隆 r86526020  蕭文宏 r88526016
巫有欽 r88526028  蕭新明 b85506013
黃毓曄 b85506010

April 11, 2000

Outline

- Why Search Engine so important
- Search Engine Architecture
  - Crawling Subsystem
  - Indexing Subsystem
  - Search Interface
- Future Trends
- Discussion
Statistics

• 1 in every 28 page views on the Web is a search result pages.  (June 1, 1999, Alexa Insider)

• The most widely traveled path on the web in March 1999 was from home.microsoft.com to www.altavista.com.  (March 1999, Alexa Insider)

• The average work user spends 73 minutes per month at search engines, second only to 97 minutes at news, info and entertainment sites.  (Feb, 1999, Internet World)

• Almost 50% of online users turn to search sites for their online news needs.  (Dec. 1998, Jupiter)

Statistics

How Internet Users Find New Websites

<table>
<thead>
<tr>
<th>Method</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search engines</td>
<td>45.8%</td>
</tr>
<tr>
<td>Word of mouth</td>
<td>20.3%</td>
</tr>
<tr>
<td>Random surfing</td>
<td>19.9%</td>
</tr>
<tr>
<td>4.4% Magazines</td>
<td></td>
</tr>
<tr>
<td>2.1% By accident</td>
<td></td>
</tr>
<tr>
<td>1.4% Newspapers</td>
<td></td>
</tr>
<tr>
<td>1.2% Television</td>
<td></td>
</tr>
<tr>
<td>1.0% Web banner</td>
<td></td>
</tr>
<tr>
<td>0.7% Don’t know</td>
<td></td>
</tr>
<tr>
<td>0.4% Radio</td>
<td></td>
</tr>
</tbody>
</table>

### Statistics

<table>
<thead>
<tr>
<th></th>
<th>Unit: millions/day</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>How Many Searches are performed</strong></td>
<td></td>
</tr>
<tr>
<td>Total Search estimated</td>
<td>94</td>
</tr>
<tr>
<td>Inktomi (Jan. 2000)</td>
<td>38</td>
</tr>
<tr>
<td>Google (Apr. 2000)</td>
<td>12</td>
</tr>
<tr>
<td>4 AskJeeves (Mar. 2000)</td>
<td></td>
</tr>
<tr>
<td>1.2 Voila (Jan. 2000)</td>
<td></td>
</tr>
</tbody>
</table>

Take Inktomi for example, it should accept 440 queries each second.

### Taxonomy

- **General-purpose Search Engine**
  - Altavista, Excite, Infoseek, Lycos, HotBot, ...
- **Hierarchical Directory**
  - Yahoo, Open Directory, LookSmart, ...
- **Meta Search Engine**
  - MetaCrawler, DogPile, SavvySearch, ...
- **Question-Answering**
  - AskJeeves
- **Specialized Search Engines**
  - HomePage Finder, Shopping robots, RealName, ...
- ...
Components

- **Spider**
  Spiders crawl the web, collect the documents through what they have found.

- **Indexer**
  Process and make a logical view of the data.

- **Search Interface**
  Accept user queries and search through the index database. Also, rank the result listing and represent to the user.
Crawling Subsystem

Spider (URL)
{
  # Use the HTTP protocol get method to acquire the web page
  Set HttpConnection = HTTPGet(URL);
  # Verify that information is accurate and not a 404 error
  Set Content = CheckInformation(HttpConnection);
  # Place the information into a database for later processing
  StoreInformation(Content);
}

Measurement of Indexed Pages

<table>
<thead>
<tr>
<th></th>
<th>FAST</th>
<th>AltaVista</th>
<th>Excite</th>
<th>Northern Light</th>
<th>Google</th>
<th>Inktomi</th>
<th>Go</th>
<th>Infosouche</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>300</td>
<td>250</td>
<td>214</td>
<td>211</td>
<td>254</td>
<td>138</td>
<td>6</td>
<td>50</td>
</tr>
<tr>
<td>Unit: Million</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Report Date: Feb.3,2000
Coverage of the Web

Report Date: Feb. 3, 2000

Issues for Crawling (1/3)

• Web Exploration with Priority
  – Decisions about which site/page is explored first
  – Ensuring document quality and coverage
  – Use Random, BFS, DFS (+depth limits) with priority

• Duplications
  – Host-wise duplications
    • Near 30% of the web are syntactically duplicated
    • ?? are semantically duplicated.
  – Single Host duplications
    • The same website with different host name
    • Symbolic links will cause some infinite routes in the web graph
  – Use Fingerprint, limited-depth exploration

• Dynamic Documents
  – Whether retrieve dynamic documents or not?
  – Single dynamic document with different parameters ?!
Issues for Crawling (2/3)

- **Load Balance**
  - **Internal**
    - Response time, size of answers are unpredictable
    - There are additional system constraints (# threads, # open connections, etc)
  - **External**
    - Never overload websites or network links (A well-connected crawler can saturate the entire outside bandwidth of some small country)
    - Support robot standard for politeness.

- **Storage Management**
  - Huge amount of url/document data

- **Freshness**
  - Many web sites (pages) changes oftenly, others nearly remains unchanged
  - Revisit different website with different periods.

Issues for Crawling (3/3)

- **The Hidden Web**
  - Some websites are not popular but valuable
  - Use Fast DNS search for possible explorations.

- **Sample Architecture of Crawling System** (Adapted from a topic-specific crawler)
**Indexer Subsystem**

Index(content, URL) {
    # Search each needed HTML structure
    Set Head = GetHtmlHead(content);
    Set Title = GetHtmlTitle(content);
    Set Keywords = GetHtmlKeyword(content);
    # Get needed keywords
    Loop {
        Set Object = CreateObject(Keywords, Title, Head, URL);
        # Store the keyword, and make internal representation
        StoreKeyword(Object, keyword);
    }
}

---

**Diagram Description**

- **User Interface**
- **Text Operations**
  - **Text**
  - **Logical View**
    - **Query Operations**
      - **Query**
      - **Searching**
      - **Retrieved Docs**
      - **Ranked Docs**
  - **Indexing**
    - **Inverted File**
    - **Index**
    - **DB Manager Module**
  - **User Feedback**
  - **User Need**
Logic View of Docs and Queries from Vector Space Model

- **Documents and Queries are treated as a t-dimension vectors**
  - t is the dimension of the whole index term space.
  - Each vector component is the weight for relevance factor for a specific index term.

- **Typical measurement for relevance**

  \[ \text{sim}(d_j, q) = \frac{d_j \cdot q}{||d_j|| \times ||q||} \]

- **Typical weighting scheme – TFxIDF**

  \[ W_{i,j} = f_{i,j} \times \log \frac{N}{n_i} \]

  - \( f_{i,j} \): term’s frequency in document,
  - \( N \): total number of documents
  - \( n_i \): total number of occurrence in different documents

- **Typical Effectiveness Measurement – Recall/Precision**

  - **Recall** = the fraction of the relevant documents which has been retrieved
  - **Precision** = the fraction of the retrieved documents which is relevant
Inverted Index

<table>
<thead>
<tr>
<th>Vocabulary</th>
<th>Occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>algorithms</td>
<td>(2,23)…</td>
</tr>
<tr>
<td>full</td>
<td>(2,6)…</td>
</tr>
<tr>
<td>heavily</td>
<td>(2,43)…</td>
</tr>
<tr>
<td>index</td>
<td>(2,60)…</td>
</tr>
<tr>
<td>letters</td>
<td>(1,60)…</td>
</tr>
<tr>
<td>made</td>
<td>(1,50)…</td>
</tr>
<tr>
<td>many</td>
<td>(1,28),(2,1)…</td>
</tr>
<tr>
<td>relies</td>
<td>(2,34)…</td>
</tr>
<tr>
<td>search</td>
<td>(2,16)…</td>
</tr>
<tr>
<td>text</td>
<td>(1,11),(1,19),(2,11)…</td>
</tr>
<tr>
<td>words</td>
<td>(1,33),(1,40)…</td>
</tr>
<tr>
<td>weighted</td>
<td>(2,51)…</td>
</tr>
</tbody>
</table>

Many full text search algorithms relies on heavily-weighted index.

Document ID = 1

This is a text. A text has many words. Words are made from letters.

Document ID = 2

Many full text search algorithms relies on heavily-weighted index.

Document 2

Text

This is a text. A text has many words. Words are made from letters.
This is a text. A text has many words. Words are made from letters.

<table>
<thead>
<tr>
<th>Word</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>001 000 110 010</td>
</tr>
<tr>
<td>many</td>
<td>000 010 101 001</td>
</tr>
</tbody>
</table>

| Block Signature | 001 010 111 011 |

Parameter

- $D$: logical block
- $F$: signature size in bits
- $m$: number of bits per word
- $F_d$: false drop probability

Issues for Indexing (1/2)

- **Language Identification**
  - Documents with different languages should be unified into a meta-representation.
  - Code conversion without concept lose.
  - How to identify language type
    - use meta data (`charset, content-encoding`) if available.
    - statistical approaches to identify language type

- **Storage Management**
  - Huge amount of indexes can not be loaded in the memory totally
  - Use cache mechanism, fast secondary storage access…
  - Efficient database structures
  - Using Compression?! Speed and Storage tradeoff
Issues for Indexing (2/2)

• Text Operations
  – Full text or controlled vocabulary
  – Stop list, Stemming, Phrase-level indexing, Thesaurus...
  – Concept discovery, Directory establishment, Categorization
  – Support search with fault tolerances ?!
  – ...

• Query-independent ranking
  – Weighting scheme for query-independent ranking
  – Web graph representation manipulations

• Structure information reservation
  – Document author, creation time, title, keywords, ...

Search Subsystem

```plaintext
Report (query) {
    # Get all relevant URLs in the internal database
    Set Candidates = GetRelevantDocuments(query);
    # Rank the lists according to its relevance scores
    Set Answer = Rank(Candidates);
    # Format the result
    DisplayResults();
}
```
What makes Web Users So Different

• Make poor queries
  – Short queries (2.35 terms for English, 3.4 characters for Chinese)
  – Imprecise terms
  – Sub-optimal syntax (80% queries without operator)

• Wide variance in
  – Needs (Some are looking for proper noun only)
  – Expectations
  – Knowledge
  – Bandwidth

• Specific behavior
  – 85% look over one result screen only
  – 78% of queries are not modified
  – Follow links

Ranking

• Goal
  order the answer set to a query in decreasing order of value

• Types
  – Query-independent : assign an intrinsic value to a document, regardless of the actual query
  – Query-dependent : value is determined only with respect to a particular query
  – Mixed : combination of both valuations

• Examples
  – Query-independent : length, vocabulary, publication data, number of citations(indegree), etc
  – Query-dependent : cosine measurement
Some ranking criteria

- **Content-based techniques**
  Variant of term vector model or probabilistic model

- **Ad-hoc factors**
  Anti-porn heuristics, publication/location data

- **Human annotations**

- **Connectivity-based techniques**
  - Query-independent
    - PageRank \[PBMW '98, BP '98\], indegree [CK'97] …
  - Query-dependent
    - HITS [K’98] …

---

Connectivity-Based Ranking

- **PageRank**
  - Consider a random Web surfer
    - Jumps to random page with probability \( \alpha \)
    - With probability \( 1 - \alpha \), follows a random hyperlink
  - Transition probability matrix is
    \[
    \alpha U + (1 - \alpha) A
    \]
    where \( U \) is the uniform distribution and \( A \) is adjacency matrix
  - Query-independent rank = stationary probability for this Markov chain
    \[
    PR(a) = \alpha + (1 - \alpha) \frac{PR(P_i)}{C(P_i)}
    \]
  - Crawling the Web using this ordering has been shown to be better than other crawling schemes.
Practical Systems
- Altavista

• Altavista configuration ’98
  – Crawler - Scooter
    • 1.5 GB memory
    • 30 GB RAID disk
    • 4x533 MHz AlphaServer
    • 1 GB/s I/O bandwidth
  – Indexing Engine – Vista
    • 2 GB memory
    • 180 GB RAID disk
    • 2x533 MHz AlphaServer
  – Search Engine – Altavista
    • 20 multi-processor machines
    • 130 GB memory
    • 500 GB RAID disk

Don’t be surprised about it!!
• Inktomi uses a cluster of hundreds of Sun Sparc
  workstation with 75 GB RAM, over 1 TB disk.
• It crawls 10 millions pages a day.

How Well does it Perform?
• Index about 0.8TB text
• No stop words
• 37 million queries on weekdays
• Mean response time = 0.6 sec

Practical Systems
- Google

• The power of PageRank
Future Trends

- **Multi-lingual/Cross-Lingual Information Retrieval**
  - Another way toward concept-oriented searching

- **Web Mining**
  - Web content mining: customer behavior analysis, advertisement
  - Web usage mining: web query log analysis

- **Personalized Search Agents**
  - Information filtering, information routing
  - More accurate user concept hierarchy mapping

- **Topic-specific knowledge base creation**

- **Question-Answering system**
  - Intelligent e-Service
  - User modeling research