

Web Search

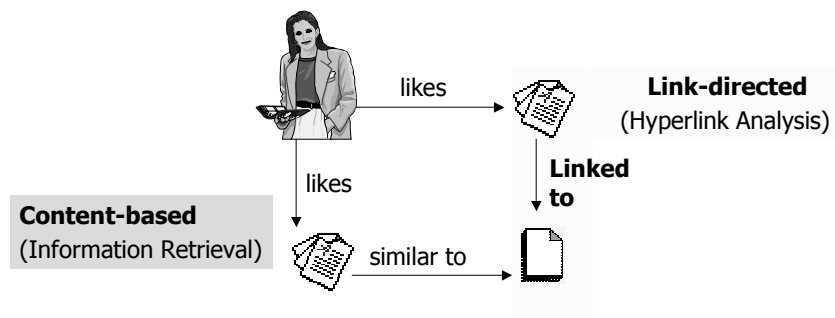
FR Informatik, In-Depth Course
Summer Semester 2004

Dr. Matthias Klusch

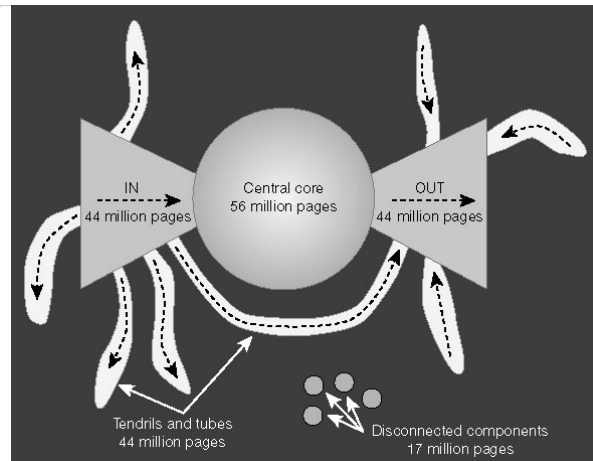


Approaches To Information Search and Recommendation

Basically, there are *link-directed*, *content-based*, or *combined (hybrid)* ways of searching for, and recommending relevant information to the user.



Connectivity of the Web ...



BFS analysis, starting from 570 randomly selected pages

Average depth of

- core in-links 482
- core out-links 434

Broder et al: <http://www9.org/w9cdrom/160/160.html>

... Motivates Link-Directed Search Via Hyperlink Analysis

„The hyperlink structure of the Web provides information on the content that is available in the Web.“

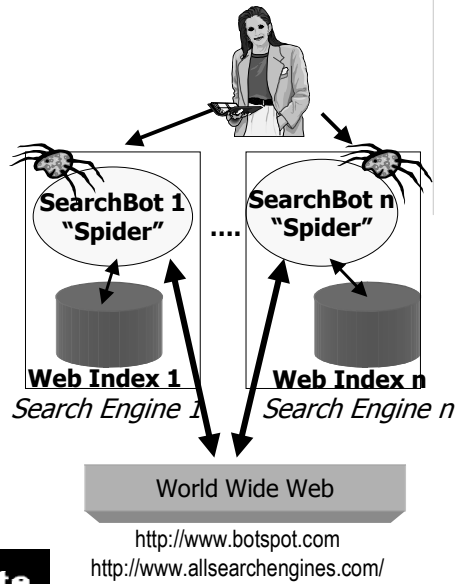
- ◆ The Web may be modeled as a directed graph capturing the hyperlink structure of the pages (URI resources).
- ◆ It is commonly assumed that
 - (1) A hyperlink from page p to page q is an individual recommendation of q by the author of p
 - (2) If pages p and q are linked then they may be relevant to the same topic
 - (3) Pages of different authors were created independently

Goal: Perform structural analysis on the hyperlink graph (HLG) to find relevant information.



Search Engines

- ▶ Use automated crawlers (spiders) to retrieve and index available pages
- ▶ Query index for relevant pages
 - ▶ hyperlink analysis
 - ▶ content-based IR
- ▶ Provide one-time query-answering
- ▶ Queries: Regular expressions

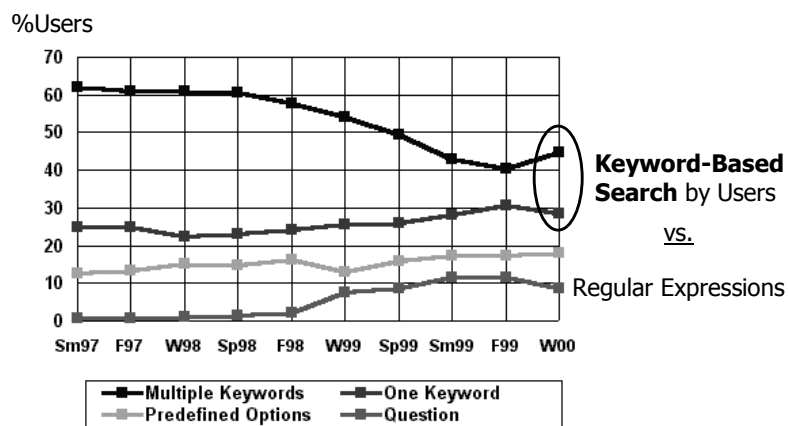


Klusch

Information Agents, C 6132, 2004

5

How Do Users Prefer To Query Search Engines?



Status: 4/2001

Klusch

Information Agents, C 6132, 2004

6

Simple Spidering Algorithm



Initialize queue (Q) with initial set of known URL's.

Until Q empty or #page limit or time limit exhausted:

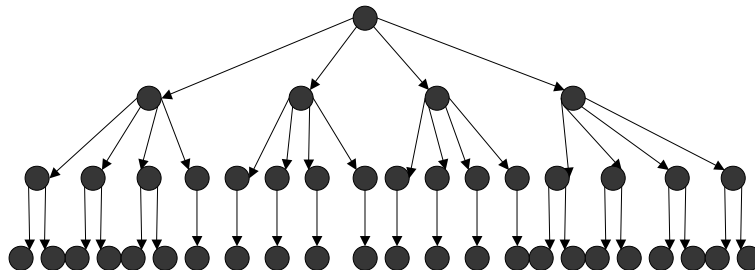
- Pop URL L from front of Q
- If L is not an HTML page (.gif, .jpeg, .ps, .pdf, .ppt...), then continue loop
- If already visited L , then continue loop
- *Download page p for L*
- If cannot download p (e.g. 404 error, robot excluded) then continue loop
- *Index P* (e.g. add to inverted index, or store cached copy)
- *Parse P to obtain list of new links N*
- *Append N to the end of Q .*

Search Strategies of Spiders



Standard spidering method: **Breadth-first search**

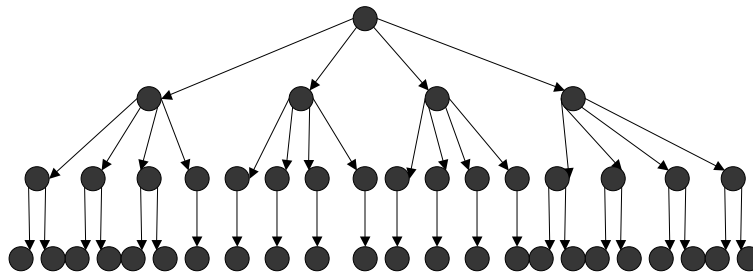
explores uniformly outward from the root page but requires memory of all the nodes on the previous level (exponential in depth)



Search Strategies of Spiders (2)



Depth-first requires memory of only depth times branching-factor (linear in depth) but can get "lost" in pursuing a single thread.



Multi-Threaded Spidering



- ◆ Problem: Main bottleneck is the usual network delay in downloading individual pages to the index of the search engine.
- ◆ Solution approach:
 - Best to have multiple threads running in parallel each requesting a page from a different host.
 - Distribute URL's to threads to guarantee equitable distribution of requests across different hosts to maximize through-put and avoid overloading any single server.
- ◆ For example, early Google spider had multiple co-ordinated crawlers with about 300 threads each, together able to download over 100 pages per second.

Limits of Spidering: The Robots Exclusion Protocol



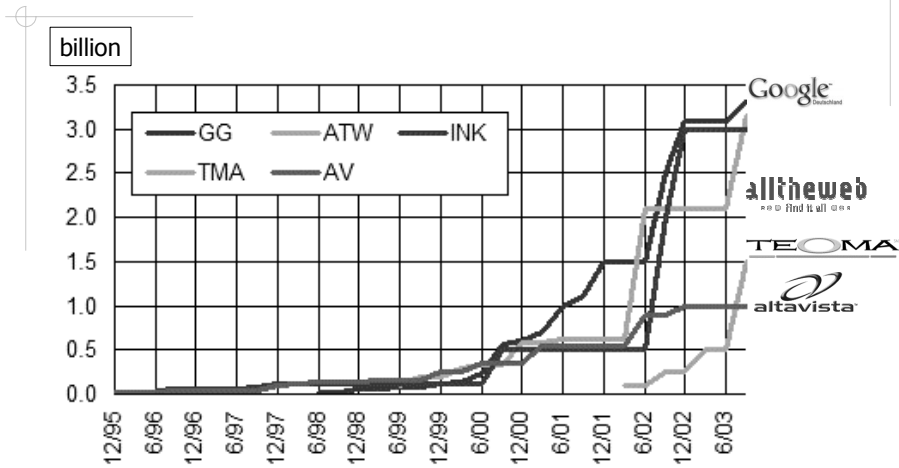
- ◆ Site administrator puts a "robots.txt" file at the root of the host's web directory.
 - Ex: www.ebay.com/robots.txt, http://www.cnn.com/robots.txt
- ◆ File is a list of excluded directories for specified spiders („robots")
 - *Exclude all robots from **the entire site***
User-agent: * Disallow: /
 - *Exclude all robots from **specific directories***:
User-agent: * Disallow: /tmp/ Disallow: /cgi-bin/
 - *Exclude **specific robots from specific directories***
User-agent: GoogleBot Disallow: /
User-agent: * Disallow: /cgi-bin

The Robots Exclusion Protocol: Robots META Tag





- ◆ Include META tag in HEAD section of a specific HTML document
 - Ex: <meta name="robots" content="none">
- ◆ META tag attribute content concerns two aspects of exclusion
 - index | noindex: Allow/disallow *indexing of this page*
 - follow | nofollow: Allow/disallow *following links on this page*
 - Special mmeta tag content attribute values:
 - ◆ "all" = "index, follow"
 - ◆ "none" = "noindex, nofollow"
 - Ex: <meta name="robots" content="noindex, follow">
- ◆ META tag is less well-adopted than use of "robots.txt"

Evolution of Search Engines' Index Size




Web Retrieval: Link-Directed and Content-Based Search

- Download pages p that have been retrieved by spider as to be relevant for given query q based on hyperlink analysis 
- Determine whether pages p are content-based relevant for q
 - Flexible string match of q with *descriptive metadata* of p (cf. "advanced search options of search engine") such as
 - Geographic location (URL of p); Modification date of p ; Language, creator, keyword (Metatags <META >) of p ; Title of p (<TITLE>), Anchor tags <A>, etc.
 - Similarity of p with q using IR model such as
 - cosine similarity measure (vector IR model)
- Parse top-ranked pages p to obtain and index new links (update index) 
- (Sort links in Q for spidering by rankings of p = topic-driven link exploration).

Example: BestFirst

```
BestFirst(topic, starting_urls) {
  foreach link (starting_urls) {
    enqueue(frontier, link);
  }
  while (#frontier > 0 and visited < MAX_PAGES) {
    link := dequeue_link_with_max_score(frontier);
    doc := fetch_new_document(link);
    score := sim(topic, doc);
    foreach outlink (extract_links(doc)) {
      if (#frontier >= MAX_BUFFER) {
        dequeue_link_with_min_score(frontier);
      }
      enqueue(frontier, outlink, score);
    }
  }
}
```



Query-Oriented Hyperlink Analysis

- ◆ Given a query expression Q by the user
 - Transformed to an internal query format of the search engine
- ◆ Create a hyperlink graph $HLG(Q)$ for Q
 - The $HLG(Q)$ consists of pages that may be relevant to Q
- ◆ Return an answer set $S(Q)$ of page links such that
 - $S(Q)$ is reasonably small and determined w/ limited computational efforts on $HLG(Q)$
 - $S(Q)$ is rich in pages that are relevant to Q
 - $S(Q)$ contains all (most) authoritative pages for Q

Hyperlink Graph of the Web

- ◆ Model the Web as a
 - Directed (hyperlink) graph $HLG = (V,E)$
 - ◆ V = finite set of all Web pages
 - ◆ (p,q) in E = Directed link from page p to page q
 - ◆ $outdegree(p)$ = number of outgoing links from p
 - ◆ $indegree(p)$ = number of incoming links to p
 - ◆ The HLG is huge in practice
 - Estimate of more than 10 billion pages
 - Use reasonably tractable sub-graphs $HLG[W] = (W,E')$
 - ◆ Restrict W to an **application dependent (query-oriented) subset** of V
 - ◆ Restrict E' to $W \times W$

How To Build A Query-Oriented Hyperlink Graph?

- ◆ Create a Root Set $RS(Q)$ of selected pages relevant to query Q by
 - Manual selection of relevant pages by human domain experts
 - Query existing subject directories on Q and select top-ranked pages
 - ◆ 1K - 5K pages in $RS(Q)$ are sufficient
 - ◆ of which one of 200 pages is pointing to an authority for Q that is not in the initially created $RS(Q)$
 - ◆ The pages in the root set are most probably not tightly connected: The initial $RS(Q)$ usually has a too sparse link structure.
 - The underlying assumption is that if there exists an authority p for Q , it is likely that some page in the $RS(Q)$ is pointing to it.

Subject Directory

- ▶ Manual classification and indexing of pages' content to **hierarchically structured subject categories** by (domain experts) editors.
- ▶ No full text but **hand-picked URLs** per category.
- ▶ External submission of links for certain category; payment.
- ▶ **Manual annotation of index entries** (reviews, rating, summary) by editors.
- ▶ One-time query-answering.



Suite101.com
Real People Helping Real People

looksmart

dmoz open directory project

INFOMINE
Scholarly Internet Resource Collections
<http://infomine.ucr.edu>

SCIRUS
for scientific information only

<http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/BeyondWeb.html#Directories>

Klusch

Information Agents, C 6132, 2004

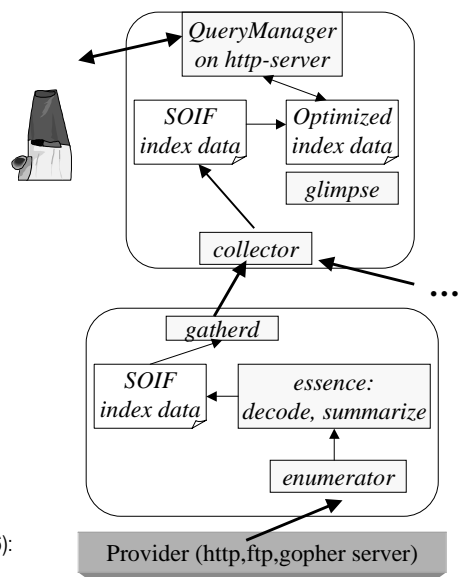
19

Classical Web Index Harvest

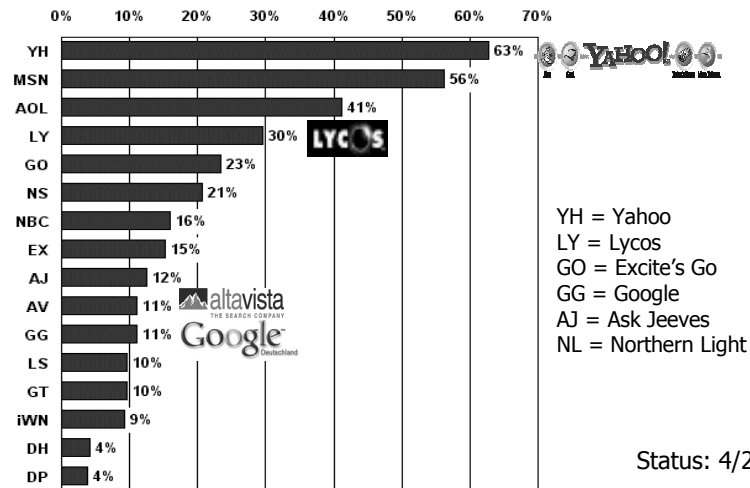
- ▶ **Users maintain/update the index**
- ▶ **Hierarchical control:** broker(s), gatherer(s)
- ▶ Recursive gathering
- ▶ Multiple summarizers
- ▶ **Summary data in SOIF** (summary object interchange format)
- ▶ Optimized index (glimpse)
- ▶ Efficient reuse of information provided by gatherers

- Insufficient user query support
- Proprietary, pre-given summarizers
- Complex configuration by user

- Harvest (Bowman et al. 1995):
<http://www.tardis.ed.ac.uk/harvest/>
- WHOIS++ Web Index Service (Weider et al. 1996):
<ftp://ds.internic.net/rfc/rfc1913.txt>
- Pharos (Dolin et al. 1997)



Subject Directories Are More Popular Than Search Engines ...



Status: 4/2001

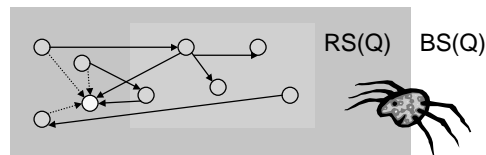
Klusch

Information Agents, C 6132, 2004

21

How To Build A Query-Oriented Hyperlink Graph? (2)

- ◆ Expand the $RS(Q)$ to a Base Set $BS(Q)$ along
 - Links that enter (in-links):
Adding limited number of pages that point to pages in $RS(Q)$
 - Links that leave (out-links):
Adding limited number of pages that are pointed to from pages in $RS(Q)$



Hyperlink graph $HLG(Q)$ for Q is $HLG[BS(Q)] = (BS(Q), E')$

Klusch

Information Agents, C 6132, 2004

22

Computing of the Base Set BS(Q)

Let

- ◆ Q query string, BS(Q) base set of pages relevant to Q
- ◆ E text-based search engine; t, d natural numbers;
- ◆ RS(Q) root set of t top-ranked results of E on Q.

BS(Q) := RS(Q);

for each p in RS(Q) **do**

{

IN(p) = set of all pages pointing to p;

OUT(p) = set of all pages p points to;

BS(Q) := BS(Q) + OUT(p);

if |IN(p)| ≤ d **then** Add all pages in IN(p) to BS(Q)
else Add arbitrary set of d pages from IN(p) to BS(Q);

}

return BS(Q)



Basic Limits of Computing BS(Q)

- To limit computational expense: Limit the number of
 - root pages in RS(Q) to the top 200 pages
- To eliminate purely navigational links:
 - Eliminate links between two pages on the same host
- To eliminate "non-authority-conveying" links:
 - Allow only m ($m \cong 4-8$) pages from a given host as pointers to any individual page.

Recall: Query-Oriented Hyperlink Analysis

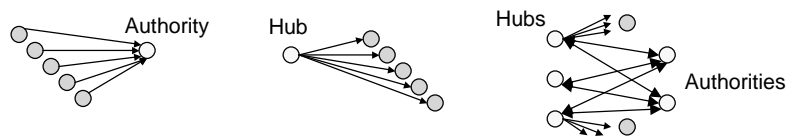
- ◆ Given a query expression Q by the user
 - Transformed to an internal query format of the search engine
- ◆ Create a hyperlink graph $HLG(Q)$ for Q
 - The $HLG(Q)$ consists of pages that may be relevant to Q
- ◆ Return an answer set $S(Q)$ of page links such that
 - $S(Q)$ is reasonably small and determined w/ limited computational efforts on $HLG(Q)$
 - $S(Q)$ is rich in pages that are relevant to Q
 - $S(Q)$ contains all (most) authoritative pages for Q

Authoritative Pages

- ◆ What is an authoritative page p for given query Q ?
- ◆ Page p is an authority page for Q
 - iff many pages that are relevant to Q link to p ,
means p has a high in-degree with respect to the $HLG(Q)$.
- ◆ High in-degree of p indicates its authority by popularity
 - The more pages that link to p in $HLG(Q)$, the greater is p 's importance to the topics requested in Q (cf. HLA assumption 3)
 - Authors of pages in the $HLG(Q)$ recommend p as to be relevant to the same topic of Q by linking their page to p .
 - But: What is meant to be a "good authority" ?

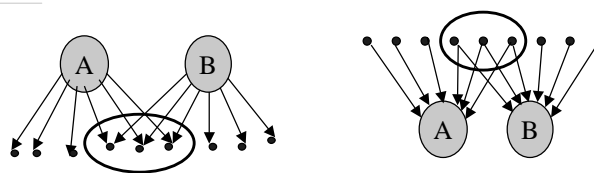
Authorities and Hubs

- ◆ A good authority page p for query Q
 - is linked to by important pages, means “hub pages” for Q , and
 - links to major hub pages for Q
- ◆ Hub page p for query Q
 - links to many relevant pages (high out-degree of p wrt. Q)



Warning: Universal authority and hub pages may not be relevant to Q !
 E.g., a link to popular subject directory altavista.com is usually not relevant to query topic “Java”, thus may cause distraction from relevant links (topic drift).

Alternative View: Citations Vs. Links



- ◆ Documents A and B are **similar** to the extent of their
 - **Bibliographic coupling** (Kessler, 1963) = The number of documents that are cited by both A and B
 - **Co-Citation** (Small, 1973) = The number of documents that cite both
- ◆ Hyperlinks are different than citations
 - Citation to relevant literature/content is *enforced* by a human user peer-review
 - Many links are purely navigational, hence do not represent any citation
 - Company websites usually don't „cite“ (point to) their competitors

The PageRank Algorithm (Page & Brin, 1998)

... is the core HLA algorithm of the search engine

Google™



Larry Page, Sergey Brin

Basic idea of PageRank

- Perform a random walk on the HLG(Q) to simulate the query driven search of any user for pages relevant to given query Q.
- Break link cycles in HLG(Q) randomly.
- Measure the importance of visited page p in HLG(Q)
 - ◆ „The more pages that link to p have themselves a high authority score, the greater is p’s importance PageRank(p).”
- **Return top-ranked authority pages p for query Q**

Web Page Importance Measure PageRank

Let be IN(p) set of all pages pointing to page p in BS(Q) Base Set for HLG(Q),
s real-valued random number

$$PageRank(p) = \frac{s}{|BS(Q)|} + (1-s) \cdot \sum_{q \in IN(p)} \frac{PageRank(q)}{outdegree(q)}$$

Real-valued function *PageRank(p)* ...

- models probability distribution of selecting page p via random navigation
 - User jumps *to a random page* - with probability s
 - User follows *link in current page p* - with probability (1 - s)
 - PageRanks of all pages p in BS(Q) are normalized to 1

PageRank (2)

IN(p) set of all pages pointing to page p in BS(Q) Base Set for HLG(Q).

$$PageRank(p) = \frac{s}{|BS(Q)|} + (1-s) \cdot \sum_{q \in IN(p)} \frac{PageRank(q)}{outdegree(q)}$$

- ... computes the authority score of p

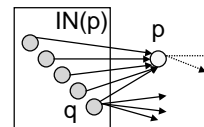
Rank all pages q that are pointing to p

How many pages point to p as an "authority"?

How important are those pages that point to p?

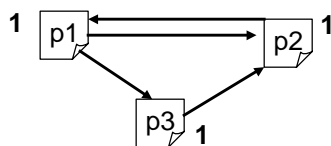
Rank of q is normalized by its out-degree

Are these pages pure authorities rather than hubs?



Example: Computing Authority Scores with PageRank

HLG(Q):



Initialize

Set $PR(p1) = PR(p2) = PR(p3) = 1$;

$outdeg(p1) = 2$

$outdeg(p2) = 1$

$outdeg(p3) = 1$

Iteration

For all pages p in the HLG(Q) do ...

- **Measure** importance $PR(p)$ of p by measuring PR of all pages from incoming links;

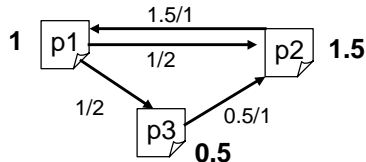
- **Propagate** $PR(p)$ to other pages **via outgoing links** of p.

... until convergence.

$$\sum_{q \in IN(p)} \frac{PageRank(q)}{outdegree(q)}$$

Example: PageRank (2)

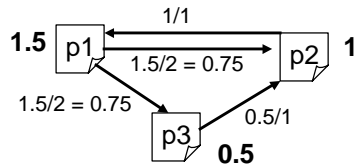
Iteration 1:



Propagate:

$$\begin{aligned} PR(p1) &= PR(p2) / \text{outdeg}(p2) &= 1 \\ PR(p2) &= PR(p1) / \text{outdeg}(p1) + PR(p3) / \text{outdeg}(p3) &= 1.5 \\ PR(p3) &= PR(p1) / \text{out}(p1) &= 0.5 \end{aligned}$$

Iteration 2:

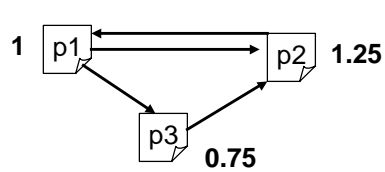


Propagate:

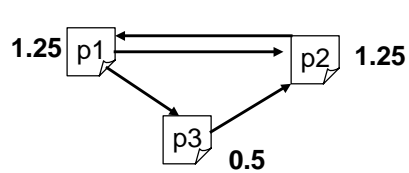
$$\begin{aligned} PR(p1) &= PR(p2) / \text{out}(p2) &= 1.5 \\ PR(p2) &= PR(p1) / \text{out}(p1) + PR(p3) / \text{out}(p3) &= 1 \\ PR(p3) &= PR(p1) / \text{out}(p1) &= 0.5 \end{aligned}$$

Example: PageRank (3)

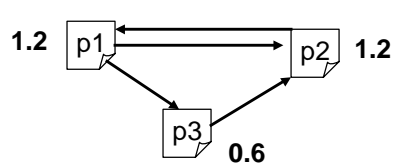
Iteration 3:



Iteration 4:



.... Result:



Pages **p1** and **p2** are **authoritative**

- p2: all pages link to p2
- p1: authority p2 links to p1

Page p3 is less authoritative

Convergence of PageRank

Basic Idea: Consider computation of PageRank for all pages as the computation of the Eigenvector of the adjacency matrix of HLG(BS(Q)).

Let be

- $n = |BS(Q)|$, $G = HLG(BS(Q))$
- $A =$ quadratic ($n \times n$) adjacency matrix of hyperlink graph G
 - $A(p,q) = 1/\text{outdeg}(p)$ iff $(p,q) \in E(G)$, $A(p,q) = 0$ else
- $x =$ real-valued n -dimensional ranking vector $(PR(p_1), \dots, PR(p_n))$

It can be proven that the PageRank formula is equivalent to $x = cAx$ ($Ax = cx$; $(A-cx) = 0$), means x is Eigenvector and c Eigenvalue of A .

Final PR-ranking x of all pages can be computed by iterative application of adjacency matrix A to initial PR-vector x .

Computing PageRank Revisited

Compute PageRank as the Eigenvector of the adjacency matrix of the HLG(Q)

Let

- A Adjacency matrix of HLG(Q)
- I Identity matrix
- c, d, t Real values
- t Upper convergence threshold

Init $x := (1, \dots, 1)$

repeat

$x' := A*x$

$c := ||x|| - ||x'||$

$x' := x' + c*I$

$d := ||x' - x||$

until $d \leq t$

Output x

PageRank Based Web Search

```
PageRank(topic, starting_urls) {
  foreach link (starting_urls) {
    enqueue(frontier, link);
  }
  while (#frontier > 0 and visited < MAX_PAGES) {
    if (multiplies_25(visited)) {
      foreach link (frontier) {
        PR(link) := recompute_PR;
      }
    }
    link := dequeue_link_with_max_PR(frontier);
    doc := fetch_new_document(link);
    score := sim(topic, doc);
    if (#buffered_pages >= MAX_BUFFER) {
      dequeue_page_with_min_score(buffered_pages);
    }
    enqueue(buffered_pages, doc);
    foreach outlink (extract_new_links(doc)) {
      if (#frontier >= MAX_BUFFER) {
        dequeue_link_with_min_PR(frontier);
      }
      enqueue(frontier, outlink);
    }
  }
}
```

The HITS Algorithm (Kleinberg, 1999)



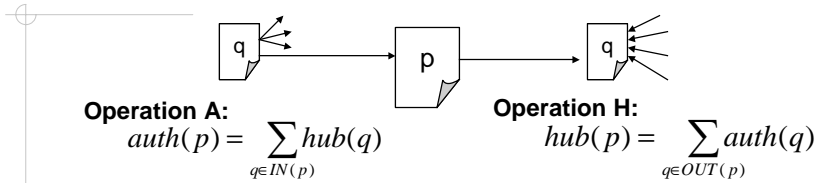
Jon Kleinberg

- ◆ Hyperlink Induced Topic Search (IBM)
- ◆ Identification of authorities *and* hubs in HLG(Q)
- ◆ Experiments showed that
 - Ranking by use of HITS is faster than PageRank on same set of pages.
 - The initial values of authority and hub scores will not affect the final computation results.

Basic idea of HITS

- Hubs and authorities exhibit a mutually reinforcing relationship!
- For each page *p* compute its reinforcing authority *and* hub scores.
- Normalize the scores such that their squares sum to 1.
- Select pages with top high authority and high hub scores.

HITS: Computing of Authority and Hub Scores



Iterate computation of mutually reinforcing authority and hub scores until convergence (fix point) is reached.

Iterate(BS(Q), k):

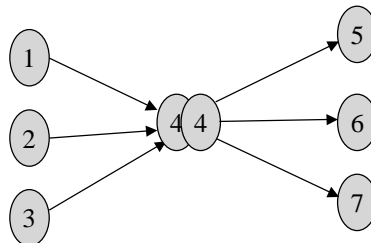
- BS(Q) set of n pages, k natural number (of iterations)
 - *auth*, *hub* n-dim vector of authority (hub) scores for all n pages in G
- for** i = 1 .. K **do**
 for j = 1 .. n **do**
 { $auth(i)[j] = A(hub(i-1), p_j)$; $hub(i)[j] = H(auth(i-1), p_j)$; }
return (*auth*, *hub*)

HITS: Computing of Authority and Hub Scores (2)

◆ HITS update rules

$$auth(p) = \sum_{q \in IN(p)} hub(q)$$

$$hub(p) = \sum_{q \in OUT(p)} auth(q)$$



$$auth(4) = hub(1) + hub(2) + hub(3)$$

$$hub(4) = auth(5) + auth(6) + auth(7)$$

The HITS Algorithm

HITS(BS(Q), k, t)

- BS(Q) base set of n linked pages
- k natural number (of iterations)
- t natural number

(*auth*, *hub*) = **Iterate**(BS(Q), k)

TopAuthorities = Set of k pages with largest values in *auth*

TopHubs = Set of k pages with largest values in *hub*

return (TopAuthorities, TopHubs)

Theorem 1: **Iterate** procedure converges (to fix points *auth** and *hub**) as k increases arbitrarily.

Experiments confirmed that **Iterate** converges quickly (k = 20).

Theorem 2: Given adjacency matrix **A** of HLG(Q) then *auth**, *hub** are the Eigenvectors of $A^T \cdot A$, and $A \cdot A^T$, respectively.

Other HLA Ranking Scheme: SALSA

(Lempel & Moran, 2000)

Perform a random walk on BS(Q) by alternately going to one of the pages

which links to the current page (in-degree) = step (a)

which is linked to by the current page (out-degree) = step (b)

Compute the

- authority scores of pages over their access probability distribution in this 2-step Markov process chain by doing **first step (a) then step (b)**

$$auth(p) = |\{p' : p' \rightarrow p\}| / \left| \bigcup_{p'' \in BS} \{p^* : p^* \rightarrow p''\} \right|$$

- hub scores of pages over their access probability distribution in this 2-step Markov process chain by doing **first step (b) then step (a)**

$$hub(p) = |\{p' : p \rightarrow p'\}| / \left| \bigcup_{p'' \in BS} \{p^* : p'' \rightarrow p^*\} \right|$$

SALSA (2)

$in(p)$ = set of pages that point to p

$out(p)$ = set of pages that p is pointing to

The transition probability of authorities in the random walk

$$P_a(p, p') = \sum_{k: k \in in(p) \cap in(p')} \frac{1}{|in(p)|} \cdot \frac{1}{|out(k)|}$$

The transition probability of hubs in the random walk

$$P_h(p, p') = \sum_{k: k \in out(p) \cap out(p')} \frac{1}{|out(p)|} \cdot \frac{1}{|in(k)|}$$

SALSA assumes no mutually reinforcing structure:

The relative authorities is determined solely from local links but

not from the whole graph like HITS does. That may avoid a „topic shift“.

Spiders for Topic-Oriented Crawling ...




◆ **Monitor** links and keep track of the in-degree and out-degree of each page encountered, and

◆ **Sort** their link queues Q

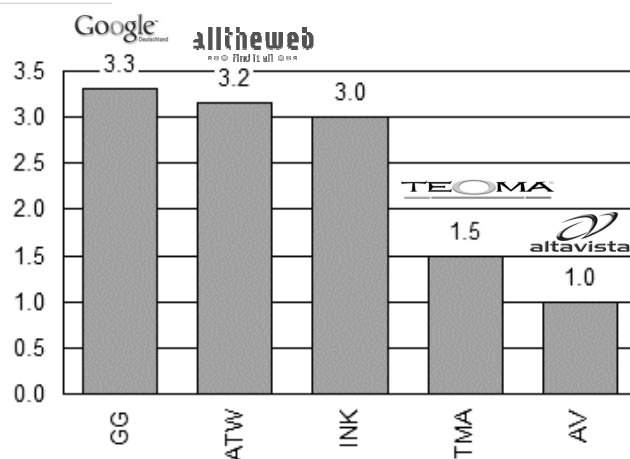
- to prefer popular pages with many in-coming links
= *authority pages*
- to prefer popular summary pages with many out-going links
= *hub pages*

... continue spidering algorithm loop

Summary: Query-Oriented Web Retrieval By Search Engines

- ◆ Given user query Q
- ◆ Build HLG(Q): Produce initial RS(Q) by querying multiple subject directories (and own index), BS(Q) by limited breadth-first crawling. 
- ◆ Download and index pages of HLG(Q) (= update index)
- ◆ Search for relevant pages by
 - Content-based querying of the updated index (IR model)
 - Present these pages, or (perform further link-directed filtering:)
 - Present only those of which are also top-ranked authorities in HLG(Q)
 - Ranking of (hub and) authority pages in HLG(Q) (HLA algorithm)
 - Present these pages, or (perform further content-based filtering:)
 - Present those of which are also highly similar to Q using content-based querying of the index

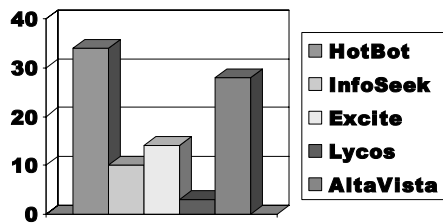
Recall: Actual Index Size of Search Engines



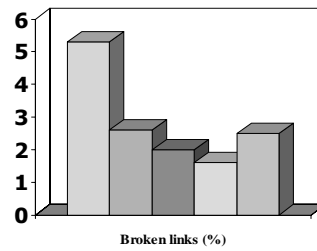
#Billions of indexed pages (2.9.2003) - Source: searchenginewatch.com

Limitations of Search Engines

Insufficient coverage (%)



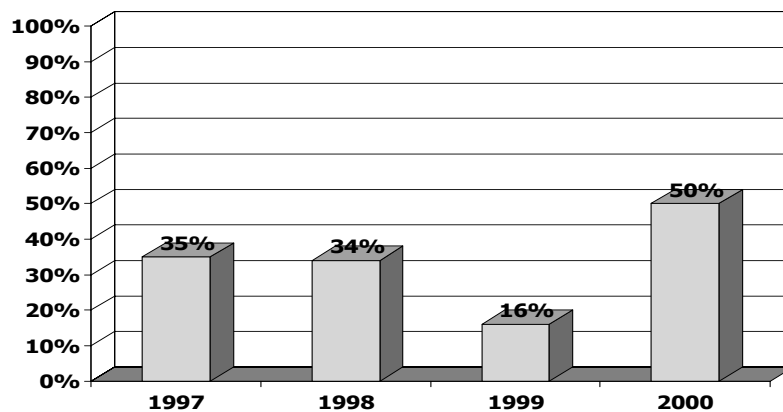
Insufficiently updated index (%)



S. Lawrence, C.L. Giles:

- Searching the world wide Web. Science, 280, 1998
- Accessibility of information on the Web. Nature, 400 (6/740), 1999

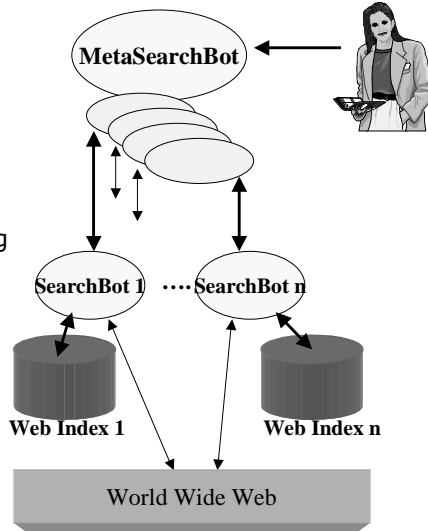
Evolution of Web Coverage by Search Engines



Source: F Menczer (Indiana U, USA), 2004

Meta-Search Engines

- ▶ Query multiple search bots and directories in parallel
- ▶ Collate ranked results
- ▶ (Cluster result pages)
- ▶ Provide uniform user interface
- ▶ Perform one time query answering



Klusch

Information Agents, C 6132, 2004

49

Benefits and Limitations of Meta-Search Engines

Significant enlargement of the search space in the Web per query

But: Enlargement is still restricted to the union of those defined by used search engines, and subject directories.

Automated translation of user query into each of the queried search bots' syntax.

But: Translation is *hard to evaluate* for user wrt. loss of information and equivalence of results.

Aggregation of results into one list of ranked hits provides uniform view on the results.

But: Rankings are *difficult to compare* due to different proprietary methods of collating results in categories and eliminating duplicates.

Common search engines do not exhibit any *pro-activity*, do not preserve data privacy (local tracking of user queries)



Klusch

Information Agents, C 6132, 2004

50

What About The So-Called „Invisible Web“ ?

- "The Invisible Web" is comprised of resources that are inaccessible to the user through any of the general-purpose search engines and directories
 - Main reasons for pages not being indexed are due to
 - Individual site policy: Robots exclusion (robots.txt)
 - Password-protected; Dynamically generated content
 - Economics: Limited search due to costs of updating comprehensive index
 - Ethics: Unethical content of Web pages, or pages mainly including spam
- Content-rich **resources of the invisible Web are searchable databases** from universities, libraries, associations, businesses, government agencies with own access portals (charge-free vs. fee-based access)

Selected examples of searchable databases such as

CiteSeer.IST
Scientific Literature Digital Library

at <http://www.invisible-web.net/>

Spiders and Search Engines: Relevant Literature

- Overview at <http://www.lib.berkeley.edu/Help/search.html>
- M Henzinger: Hyperlink analysis for the Web. IEEE Internet Computing, 2001
- S Brin, L Page: The anatomy of a large-scale hypertextual web search engine. Proc. 10th Intern. Conference on the World Wide Web, 1998
- V.N. Gudivada: Information Retrieval on the worldwide Web. IEEE Internet Computing, 1(5), 1997
- J. Kleinberg, A. Tomkins: Applications of linear algebra in information retrieval and hypertext analysis. Proc. ACM SIGMOD PODS, 1999
- J. Kleinberg: Authoritative sources in a hyperlinked environment. J ACM, 46(5): 604-632, 1999
- F Menczer et al.: Evaluating topic-driven Web crawlers. Proc. ACM Conference SIGIR, 2001
- Proceedings of international conference series on the WWW

Spiders and Search Engines: Selected Resources

- ◆ PageRank java class:
jung.sourceforge.net/api/1.3.0/edu/uci/ics/jung/algorithms/importance/PageRank.html
- ◆ HITS java class:
jung.sourceforge.net/api/1.3.0/edu/uci/ics/jung/algorithms/importance/HITS.html
- ◆ Open source crawlers/search engines
 - Nutch: <http://www.nutch.org/>
 - Jakarta Lucene: <http://jakarta.apache.org/lucene/>
- ◆ Open source topical crawlers, Best-First-N (Java)
 - <http://informatics.indiana.edu/fil/IS/JavaCrawlers/>
- ◆ LWP (Library for WWW in Perl)
 - <http://www.oreilly.com/catalog/perlwp/> (doc)
 - <http://search.cpan.org/~gaas/libwww-perl-5.79/> (Perl code)
- ◆ Evaluation framework for topical crawlers (Perl)
 - <http://informatics.indiana.edu/fil/IS/Framework/>

Alternative: Local and Structured Web Search

Basic idea

- Represent Web content by means of a structured data model
 - Allows a more structured (e.g. SQL-like) Web search
- Focus your Web search on particular site, or its „neighbors“ in a given range.
 - Avoids missing local site hits that are caused by global search engines

Selected Examples:

- WebSphinx crawler www-2.cs.cmu.edu/~rcm/websphinx



- WebGlimpse crawler webglimpse.net



• **Web Query Languages**

- W3QS (Shmueli+,1995), WebSQL (Mendelzon+, 1997), WebLog, WQL
- STRUQL (Levy+,1997), FLORID (Lausen+,1997)
- WebOQL (Mendelzon+,1998)

WebSQL: Graph-based Structured Web Search

Arocena, Levy, Milo & Mendelzon, 1997; DB Group at U Toronto

(1) Represent Web sites in a relational database scheme

document(url, title, text, type, length, modif)

anchor(base, label, href)

(2) Execute SQL-like queries over relational schema and HLG(Q)

- Keyword-based (via pattern matching on attributes of relations)
- **Structured** (via use of regular path expression R applied to HLG)
 - R consists of concatenation |, repetition * of hypertext-link-symbols:
 - **local** link -> refers to docs located on the same (local) server
 - **global** link => refers to docs located on different servers; use index server/spider to explore external links
 - R-driven search in the HLG is limited by number n of, or fewer local links: $R (->) \leq n$

Example: WebSQL Query Classes

• Global query type

Get all documents x in **some sites**, and find docs y in sites which are linked to those of x as defined by path expression R, and x contains keyword „k“

Select x.url, y.url **from** document x, document y
such that x mentions “k” **and** xRy

• Local query type

Get all documents x in **site i** or one-step neighbour site of i, or sites that are linked to i as defined by path expression R, and x contains keyword „k“.

Select x.url **from** document x **such that** iRx
where x.text **contains** “k”

Example: Structured Web Search By Use of WebSQL Queries

Select x.url **from** document x **such that** www.dfki.de =|->|->->x
where x.title contains "agents"

Meaning: Start from www.dfki.de page, find all docs x which contain "agents" in title and are not more than 2 links away on the same **(local)** server.

Select x.url; y.url **from** document x **such that**
x.text **contains** "agents", document y **such that** x=|->|->->y;
where y.text contains "agents"

Meaning: Find all docs x mentioning "agents" and all docs y linked to them via paths of length smaller than 2 and also mentioning "agents" in their text body. **(bounded)**

Select x.url; x.title **from** document x
where x.text **contains** "agents"

Meaning: Find all docs about "agents" that are reachable. **(global)**