Tutorial:

Web Information Retrieval

Monika Henzinger

http://www.henzinger.com/~monika

What is this talk about?

- Topic:
  - Algorithms for retrieving information on the Web
- Non-Topics:
  - Algorithmic issues in classic information retrieval (IR), e.g. stemming
  - String algorithms, e.g. approximate matching
  - Other algorithmic issues related to the Web:
    - networking & routing
    - cacheing
    - security
    - e-commerce

Preliminaries

- Each Web page has a unique URL

Page name

http://theory.stanford.edu/~focs98/tutorials.html

Access protocol

Host name = Domain name

Path

- (Hyper) link = pointer from one page to another, loads second page if clicked on
- In this talk: document = (Web) page

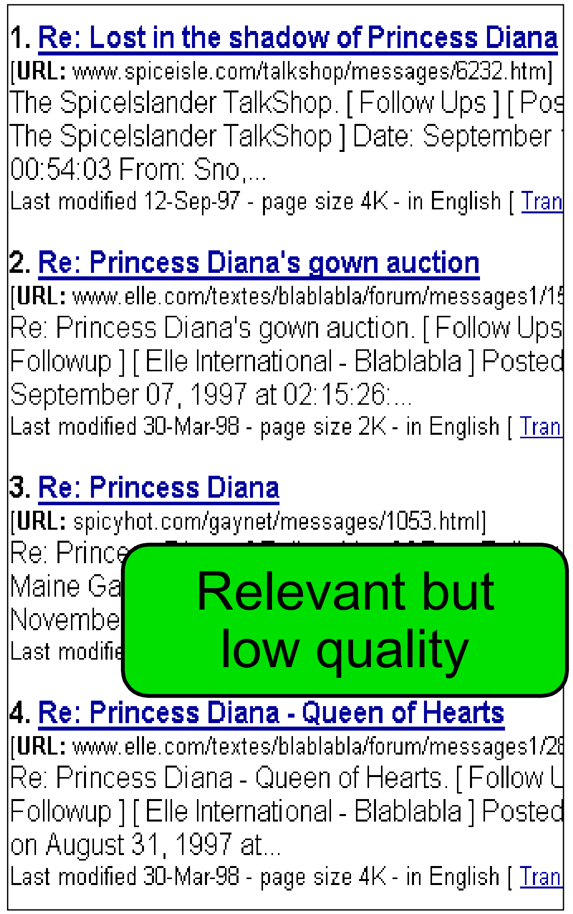
Example of a query

princess diana

Engine 1

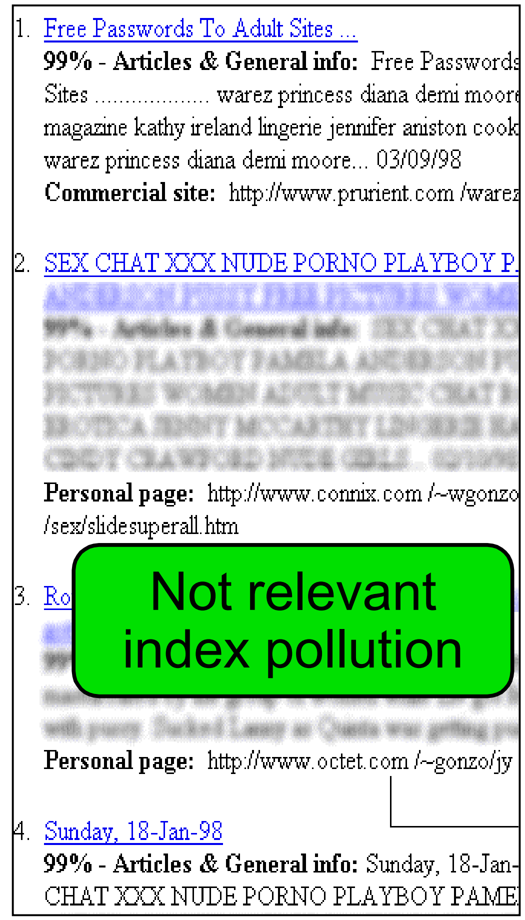
Relevant and high quality

Engine 2



Relevant but low quality

Engine 3



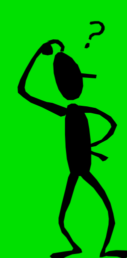
Not relevant index pollution

Outline

- Classic IR vs. Web IR
- Some IR tools specific to the Web

  For each type
  - Examples
  - Algorithmic issues
- Conclusions

Details on
- Ranking
- Duplicate elimination
- Search-by-example
- Measuring search engine index quality



Open problems

Classic IR

- Input: Document collection
- Goal: Retrieve documents or text with information content that is relevant to user's information need
- Two aspects:
  1. Processing the collection
  2. Processing queries (searching)
- Reference Texts: SW'97, BR'99

Determining query results

"model" = strategy for determining which documents to return

- Logical model: String matches plus AND, OR, NOT
- Vector model (Salton et al.):
  - Documents and query represented as vector of terms
  - Vector entry *i* = weight of term *i* = function of frequencies within document and within collection
  - Similarity of document & query = cosine of angle of their vectors
  - Query result: documents ordered by similarity
- Other models used in IR but not discussed here:
  - Probabilistic model, cognitive model, …

IR on the Web

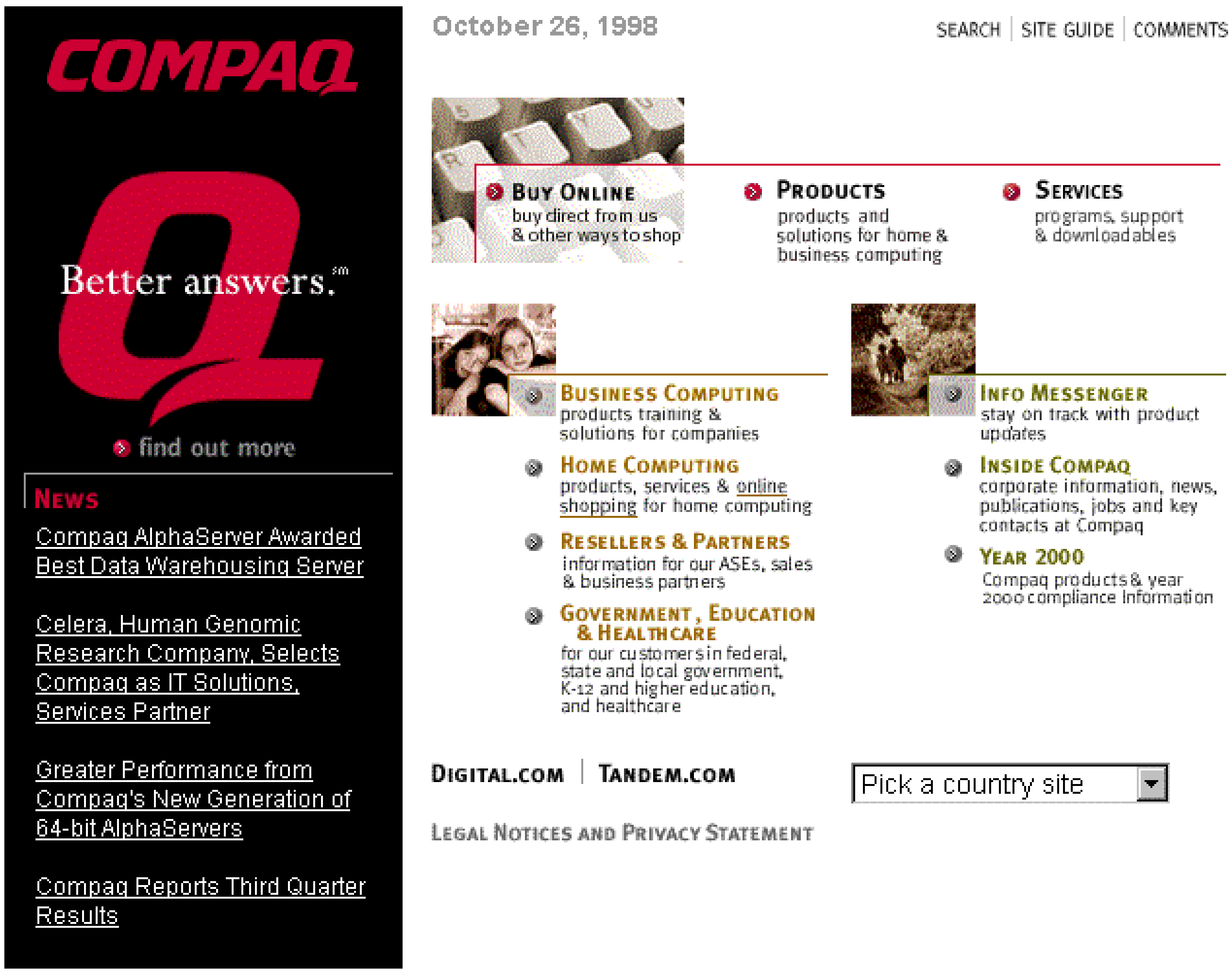
- Input: The publicly accessible Web
- Goal: Retrieve high quality pages that are relevant to user's need
  - Static (files: text, audio, …)
  - Dynamically generated on request: mostly data base access
- Two aspects:
  1. Processing and representing the collection
     - Gathering the static pages
     - "Learning" about the dynamic pages
  2. Processing queries (searching)

What's different about the Web?

(1) Pages:

- Bulk …………………… >1B (12/99) [GL'99]
- Lack of stability……….. Estimates: 23%/day, 38%/week [CG'99]
- Heterogeneity
  - Type of documents .. Text, pictures, audio, scripts,…
  - Quality ……………… From dreck to ICDE papers …
  - Language ………….. 100+
- Duplication
  - Syntactic……………. 30% (near) duplicates
  - Semantic……………. ??
- Non-running text……… many home pages, bookmarks, ...
- High linkage…………… ≥ 8 links/page in the average

Typical home page: non-running text



Typical home page: Non-running text

The big challenge

Meet the user needs
given
the heterogeneity of Web pages

What's different about the Web?

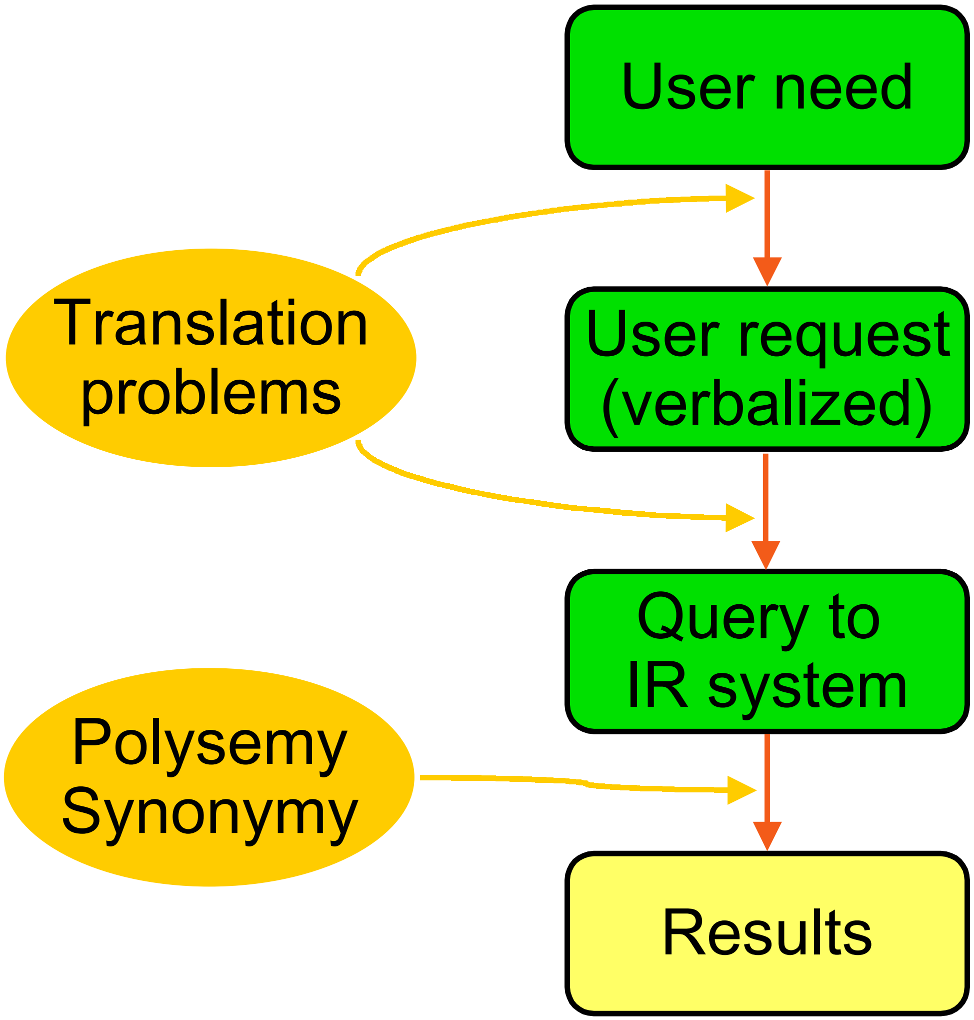
(2) Users:

- Make poor queries
  - Short (2.35 terms avg)
  - Imprecise terms
  - Sub-optimal syntax (80% queries without operator)
  - Low effort
- Wide variance in
  - Needs
  - Knowledge
  - Bandwidth
- Specific behavior
  - 85% look over one result screen only
  - 78% of queries are not modified
  - Follow links
  - See various user studies in CHI, Hypertext, SIGIR, etc.

The bigger challenge

Meet the user needs
given
the heterogeneity of Web pages
and
the poorly made queries.

Why don't the users get what they want?



Example

I need to get rid of mice in the basement

What's the best way to trap mice alive?

`mouse trap`

Software, toy cars, inventive products, etc

Google output: mouse trap

bigsun.wbs.net/homepages/m/o/u/mouse_trap
New! Try out GoogleScout

Doc Fizzix - Mousetrap Cars and Mouse Trap Powered Vehicles
...The best mousetrap cars& **mouse trap** cars site! **Mouse**...
www.docfizzix.com/ Cached (14k) New! Try out GoogleScout

Mouse Trap
... **Mouse Trap Mouse Trap** is a simple but effective...
...can also be configured to **trap** the **mouse** on system startup or at a...
www.homeonthewww.com/ryan/mousetrap.html Cached (5k) New! Try out GoogleScout

Tin Cat Repeating Mouse Trap
www.biconet.com/critter/tincat.html Cached (11k) New! Try out GoogleScout

J

Google output: trap mice

Smart Mouse Trap
www.biconet.com/critter/smt.html Cached (11k) New! Try out GoogleScout J

Tin Cat Repeating Mouse Trap
www.biconet.com/critter/tincat.html Cached (11k) New! Try out GoogleScout J

Horned Owl Inflatable Scarecrow
www.biconet.com/critter/owl.html Cached (10k) New! Try out GoogleScout

Rat Traps, mice traps ,glue boards, moth traps,pantry pest traps
...MULTIPLE TRAPS FOR **MICE** MOUSE MASTER A multiple catch **trap** for... J
...Single **Trap** #855 + Lure $22.06 SNAP TRAPS FOR RATS AND **MICE** RAT...
doyourownpestcontrol.com/traps.htm Cached (40k) New! Try out GoogleScout

Mice
...the wall (see our **trap** placement guide) since **mice** mostly navigate...
...those signs of **mice**? That's where you place the **trap**. **Mice**... J
www.unexco.com/mice.html Cached (15k) New! Try out GoogleScout

National Food Safety Database: Disaster Handbook
...traps are needed in a house to **trap mice** than rats. Rats and...
...week before they approach a **trap**. **Mice** are curious and will...
www.foodsafety.org/dh/dh044.htm Cached (21k) New! Try out GoogleScout

The bright side: Web advantages vs. classic IR

User

- Many tools available
- Personalization
- Interactivity (refine the query if needed)

Collection/tools

- Redundancy
- Hyperlinks
- Statistics
  - Easy to gather
  - Large sample sizes
- Interactivity (make the users explain what they want)

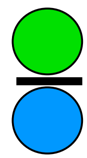
Quantifying the quality of results

- How to evaluate different strategies?
- How to compare different search engines?

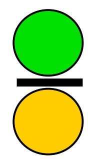
Classic evaluation of IR systems

We start from a **human made** relevance judgement for each (query, page) pair and compute:

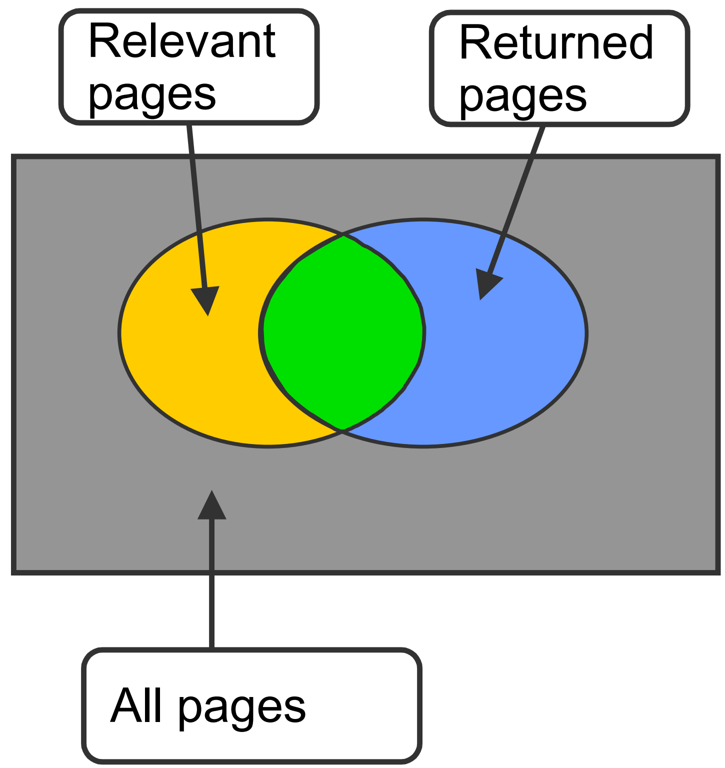
- Precision: % of returned pages that are relevant.



- Recall: % of relevant pages that are returned.



- Precision at (rank) 10: % of top 10 pages that are relevant
- Relative Recall: % of relevant pages *found by some means* that are returned



Evaluation in the Web context

- Quality of pages varies widely
- We need both relevance and high quality = **value** of page.
  - Precision at 10: % of top 10 pages that are **valuable** …

Web IR tools

- General-purpose search engines:
  - direct: AltaVista, Excite, Google, Infoseek, Lycos, ….
  - Indirect (Meta-search): MetaCrawler, DogPile, AskJeeves, InvisibleWeb, …
- Hierarchical directories: Yahoo!, all portals.
- Specialized search engines:
  - Home page finder: Ahoy
  - Shopping robots: Jango, Junglee,…
  - Applet finders

Database mostly built by hand

Web IR tools (cont...)

- Search-by-example: Alexa's "What's related", Excite's "More like this", Google's "Googlescout", etc.
- Collaborative filtering: Firefly, GAB, …
- …

- Meta-information:
  - Search Engine Comparisons
  - Query log statistics
  - …

General purpose search engines

- Search engines' components:
  - Spider = Crawler -- collects the documents
  - Indexer -- process and represents the data
  - Search interface -- answers queries

Algorithmic issues related to search engines

- Collecting documents
  - Priority
  - Load balancing
    - Internal
    - External
  - Trap avoidance
  - ...
- Processing and representing the data
  - Query-independent ranking
  - Graph representation
  - Index building
  - Duplicate elimination
  - Categorization
  - …
- Processing queries
  - Query-dependent ranking
  - Duplicate elimination
  - Query refinement
  - Clustering
  - ...

Ranking

- Goal: order the answers to a query in decreasing order of value
  - Query-independent: assign an intrinsic value to a document, regardless of the actual query
  - Query-dependent: value is determined only wrt a particular query.
  - Mixed: combination of both valuations.
- Examples
  - Query-independent: length, vocabulary, publication data, number of citations (indegree), etc.
  - Query-dependent: cosine measure

Some ranking criteria

- Content-based techniques (variant of term vector model or probabilistic model) – mostly query-dependent
- Ad-hoc factors (anti-porn heuristics, publication/location data, ...) – mostly query-independent
- Human annotations
- Connectivity-based techniques
  - Query-independent: PageRank [PBMW’98, BP’98], indegree [CK’97], …
  - Query-dependent: HITS [K’98], …

Connectivity analysis

- Idea: Mine hyperlink information of the Web
- Assumptions:
  – Links often connect related pages
  – A link between pages is a recommendation
- Classic IR work (citations = links) a.k.a. “Bibliometrics” [K’63, G’72, S’73,…]
- Socio-metrics [K’53, MMSM’86,…]
- Many Web related papers build on this idea [PPR’96, AMM’97, S’97, CK’97, K’98, BP’98,…]

Graph representation for the Web

- A node for each page u
- A directed edge (u,v) if page u contains a hyperlink to page v.

Query-independent ranking: Motivation for PageRank

- Assumption: A **link** from page A to page B is a **recommendation** of page B by the author of A (we say B is *successor* of A)
- Quality of a page is related to its in-degree

- Recursion: Quality of a page is related to
  – its in-degree, and to
  – the quality of pages linking to it
- **PageRank** [BP ‘98]

Definition of PageRank [BP'98]

- Consider the following infinite random walk (surf):
  - Initially the surfer is at a random page
  - At each step, the surfer proceeds
    - to a randomly chosen web page with probability d
    - to a randomly chosen successor of the current page with probability 1-d
- **The PageRank of a page p is the fraction of steps the surfer spends at p in the limit.**

PageRank (cont.)

Said differently:

- Transition probability matrix is

$$d \times U + (1-d) \times A$$

where U is the uniform distribution and A is adjacency matrix (normalized)

- PageRank = stationary probability for this Markov chain, i.e.

$$PageRank(u) = \frac{d}{n} + (1-d) \sum_{(v,u) \in E} PageRank(v) / outdegree(v)$$

where n is the total number of nodes in the graph

- Used as one of the ranking criteria in Google

Output from Google: princess diana

Diana, Princess of Wales
... **Diana, Princess** of Wales 1 July 1961 - 31 August 1997 The BBC Web...
www.royal.gov.uk/start.htm Cached (2k) New! Try out GoogleScout

J

www.royal.gov.uk/
New! Try out GoogleScout

J

Princess Diana: Remember Diana, Princess of Wales
...This ribbon is in memory of **Diana, Princess** of Wales. Please put...
...indefinitely as a tribute to **Diana, Princess** of Wales. I feel I...
www.gargaro.com/diana.html Cached (8k) New! Try out GoogleScout

J

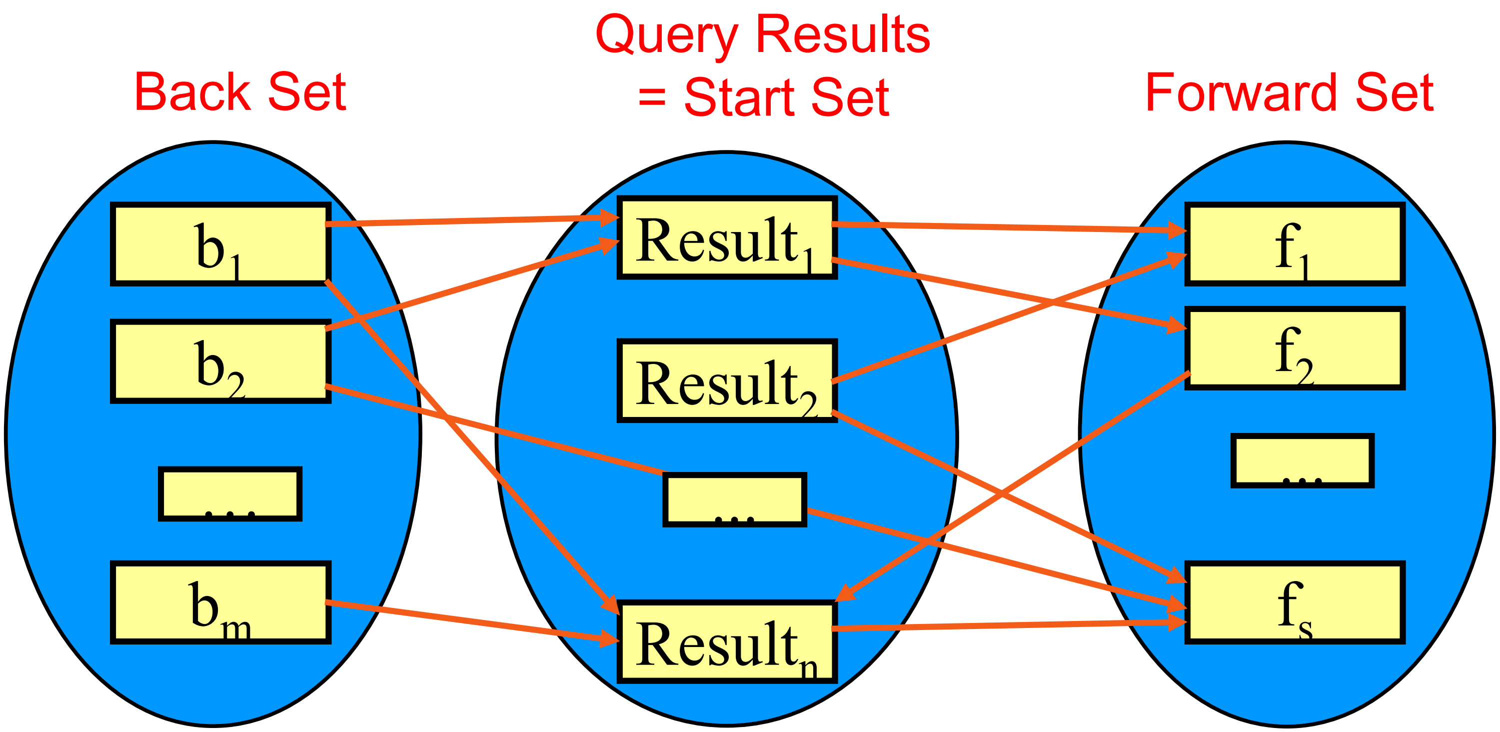
www.geocities.com/RainForest/Vines/1009/diana.htm
New! Try out GoogleScout

J

CNN - The Death of Princess Diana
...▪ The Burial: **Princess Diana**'s coffin is taken to family...
...service held in memory of **Princess Diana** - VXtreme streaming video...
www.cnn.com/WORLD/9708/31/diana.links/ Cached (12k) New! Try out GoogleScout

Query-dependent ranking: the neighborhood graph

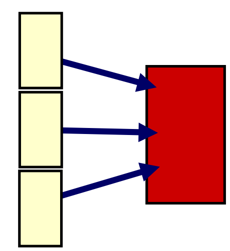
- Subgraph associated to each query



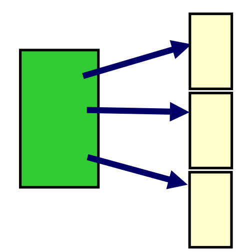
An edge for each hyperlink, but no edges within the same host

HITS [K'98]

- Goal: Given a query find:
  - Good sources of content (authorities)

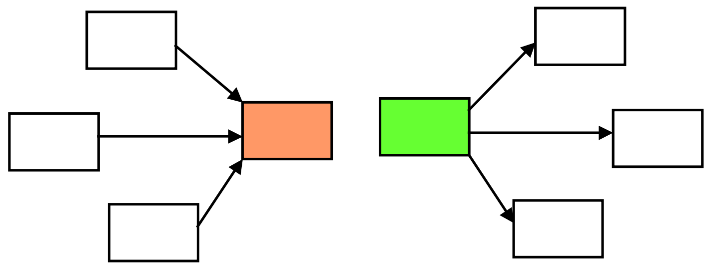


  - Good sources of links (hubs)

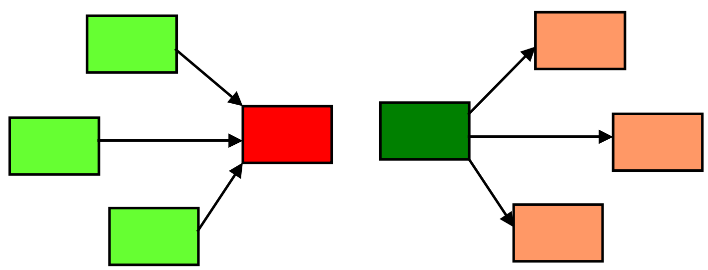


Intuition

- Authority **comes from** in-edges.
  Being a good hub **comes from** out-edges.



- Better authority **comes from** in-edges from good hubs.
  Being a better hub **comes from** out-edges to good authorities.



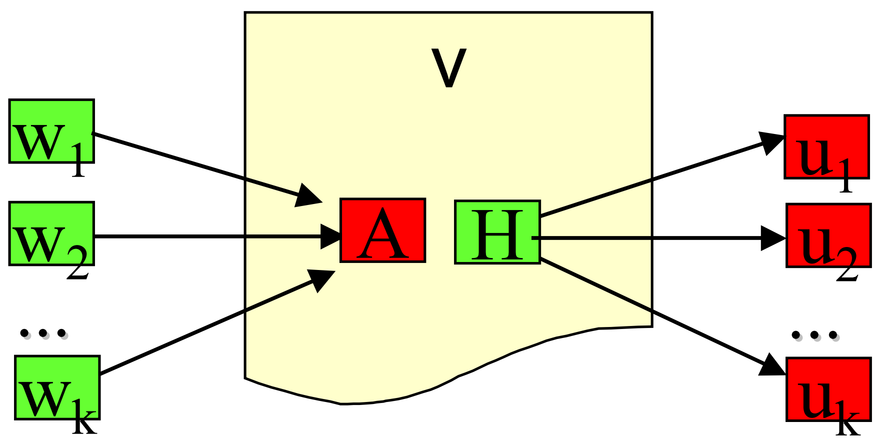
HITS details

Repeat until $\overrightarrow{HUB}$ and $\overrightarrow{AUTH}$ converge:

Normalize $\overrightarrow{HUB}$ and $\overrightarrow{AUTH}$

$HUB[v] := \Sigma\, AUTH[u_i]$ for all u_i with $Edge(v, u_i)$

$AUTH[v] := \Sigma\, HUB[w_i]$ for all w_i with $Edge(w_i, v)$



Output from HITS: jobs

1. www.ajb.dni.uk - British career website J
2. www.britnet.co.uk/jobs.htm
3. www.monster.com - US career website J
4. www.careermosaic.com - US career website J
5. plasma-gate.weizmann.ac.il/Job…
6. www. jobtrak.com - US career website J
7. www.occ.com - US career website J
8. www.jobserve.com - US career website J
9. www.allny.com/jobs.html - jobs in NYC
10. www.commarts.com/bin/… - US career website J

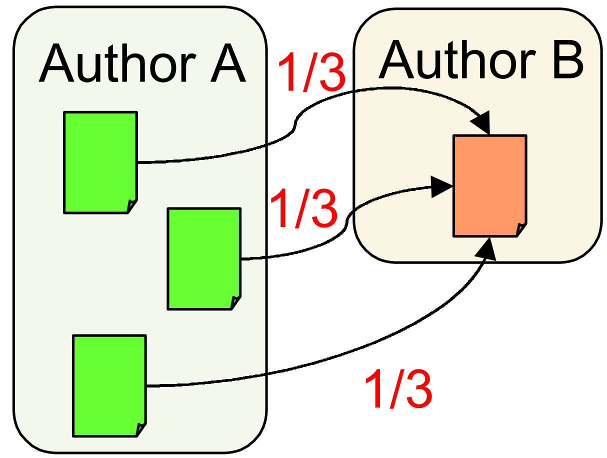
Output from HITS: +jaguar +car

1. www.toyota.com
2. www.chryslercars.com
3. www.vw.com
4. www.jaguravehicles.com J
5. www.dodge.com
6. www.usa.mercedes-benz.com
7. www.buick.com
8. www.acura.com
9. www.bmw.com
10. www.honda.com

Problems & solutions

- Some edges are "wrong" -- not a recommendation:
  - multiple edges from same author
  - automatically generated
  - spam, etc.

  Solution: Weight edges to limit influence



- Topic drift
  - Query: **`+jaguar +car`**
    Result: pages about **`cars`** in general

  Solution: Analyze content and assign topic scores to nodes

Modified HITS algorithms

Repeat until $\overrightarrow{\text{HUB}}$ and $\overrightarrow{\text{AUTH}}$ converge:

Normalize $\overrightarrow{\text{HUB}}$ and $\overrightarrow{\text{AUTH}}$

$\text{HUB}[v] := \Sigma\ \text{AUTH}[u_i]\ \text{TopicScore}[u_i]\ \text{weight}[v,u_i]$

for all u_i with Edge(v, u_i)

$\text{AUTH}[v] := \Sigma\ \text{HUB}[w_i]\ \text{TopicScore}[w_i]\ \text{weight}[w_i,v]$

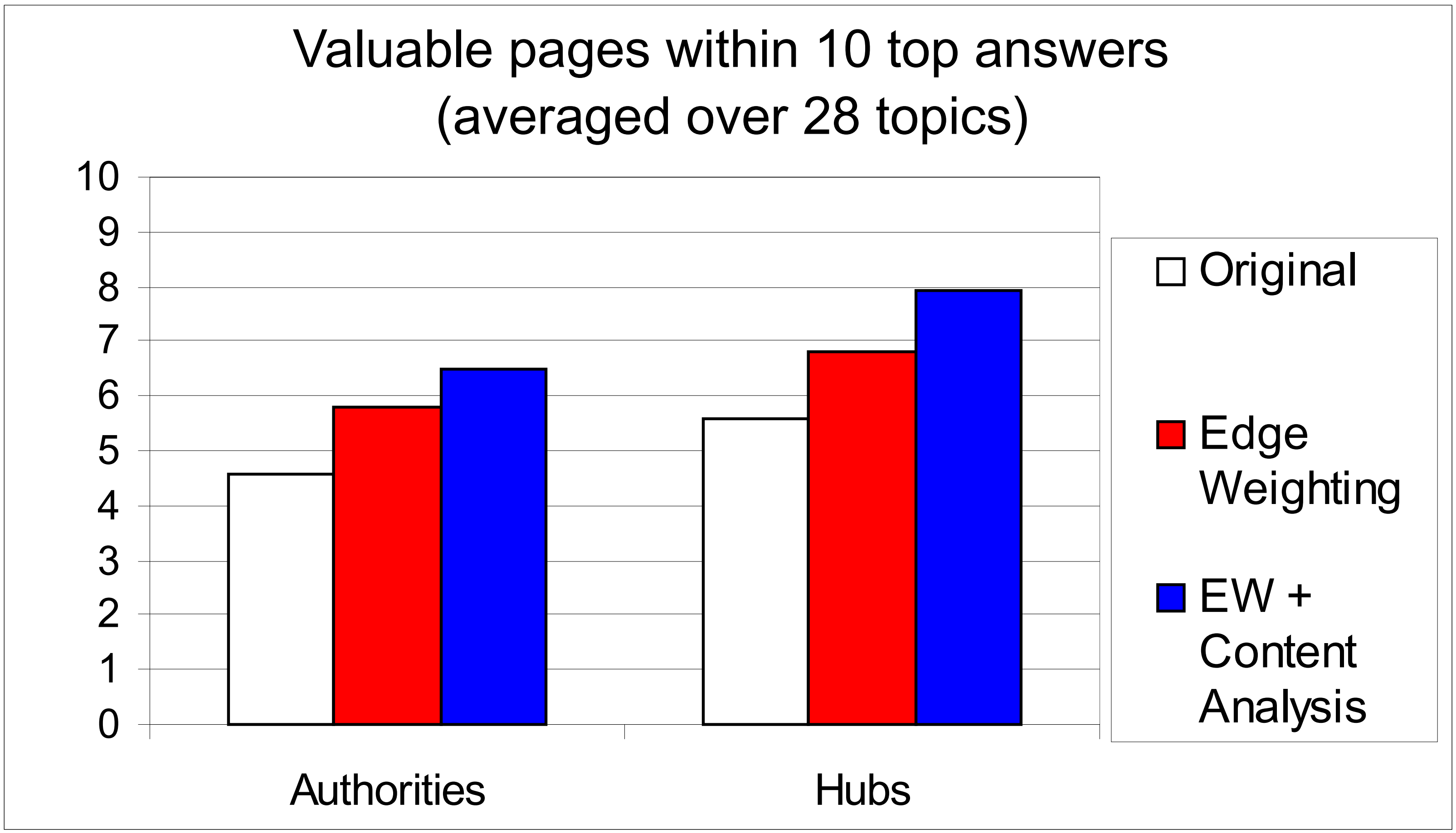
for all w_i with Edge(w_i, v)

[CDRRGK'98, BH'98, CDGKRRT'98]

Output from modified HITS: *+jaguar +car*

1. www.jaguarcars.com/ - official website of Jaguar cars J
2. www.collection.co.uk/ - official Jaguar accessories J
3. home.sn.no/.../jaguar.html - the Jaguar Enthusiast Place J
4. www.terrysjag.com/ - Jaguar Parts J
5. www.jaguarvehicles.com/ - official website of Jaguar cars J
6. www.jagweb.com/ - for companies specializing in Jags. J
7. jagweb.com/jdht/jdht.html - articles about Jaguars and Daimler
8. www.jags.org/ - Oldest Jaguar Club J
9. connection.se/jagsport/ - Sports car version of Jaguar MK II
10. users.aol.com/.../jane.htm -Jaguar Association of New England Ltd.

User study [BH'98]



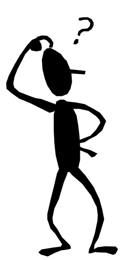
PageRank vs. HITS

- Computation:
  - Expensive
  - Once for all documents and queries (offline)
- Query-independent – requires combination with query-dependent criteria
- Hard to spam

- Computation:
  - Expensive
  - Requires computation for each query
- Query-dependent
- Relatively easy to spam
- Quality depends on quality of start set
- Gives hubs as well as authorities

Open problems

- Compare performance of query-dependent and query-independent connectivity analysis
- Exploit order of links on the page (see e.g. [CDGKRRT'98],[DH'99])
- Both Google and HITS compute principal eigenvector. What about non-principal eigenvector? ([K'98])



- Derive other graphs from the hyperlink structure …

Algorithmic issues related to search engines

- Collecting documents
  - Priority
  - Load balancing
    - Internal
    - External
  - Trap avoidance
  - ...
- Processing and representing the data
  - Query-independent ranking
  - Graph representation
  - Index building
  - Duplicate elimination
  - Categorization
  - …
- Processing queries
  - Query-dependent ranking
  - Duplicate elimination
  - Query refinement
  - Clustering
  - ...

More on graph representation

- Graphs derived from the hyperlink structure of the Web:
  - Node =page
  - Edge (u,v) iff pages u and v are related in a specific way (directed or not)
- Examples of edges:
  - iff u has hyperlink to v
  - iff there exists a page w pointing to both u and v
  - iff u is often retrieved within x seconds after v
  - …

Graph representation usage

- Ranking algorithms
  - PageRank
  - HITS
  - …
- Search-by-example [DH'99]
- Categorization of Web pages
  - [CDI'98]
- Visualization/Navigation
  - Mapuccino [MJSUZB'97]
  - WebCutter [MS'97]
  - …
- Structured Web query tools
  - WebSQL [AMM'97]
  - WebQuery [CK'97]
  - …

Example: SRC Connectivity Server [BBHKV'98]

Directed edges = Hyperlinks

- Goal: Support two basic operations for all URLs collected by AltaVista
  - InEdges(URL *u*, int *k*)
    - Return k URLs pointing to u
  - OutEdges(URL *u*, int *k*)
    - Return k URLs that u points to
- Difficulties:
  - Memory usage (~180 M nodes, 1B edges)
  - Preprocessing time (days …)
  - Query time (~ 0.0001s/result URL)

URL database

Sorted list of URLs is 8.7 GB (≈ 48 bytes/URL)

Delta encoding reduces it to 3.8 GB (≈ 21 bytes/URL)

Original text

www.foobar.com/
www.foobar.com/gandalf.htm
www.foograb.com/

Delta Encoding

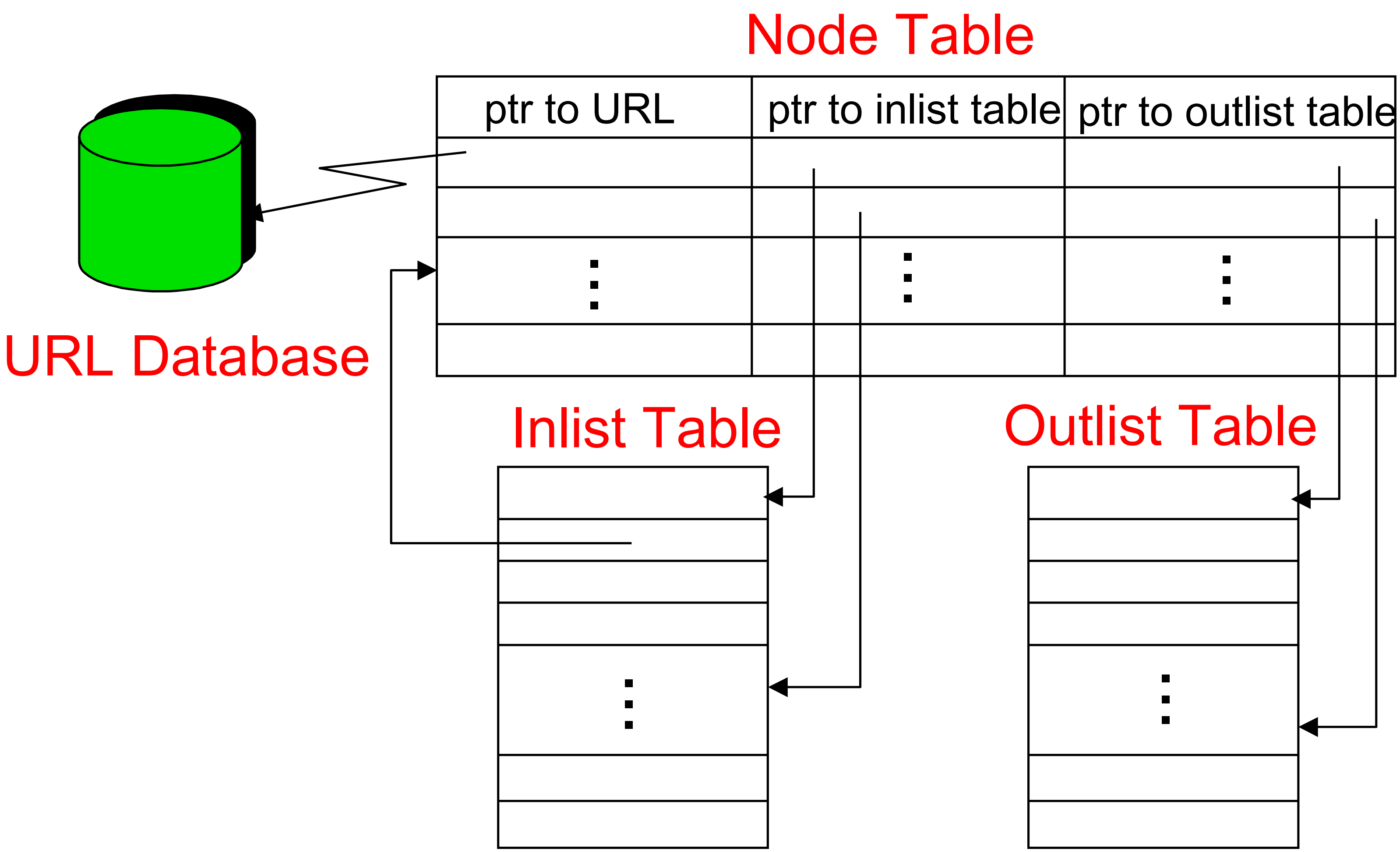
0 www.foobar.com/ 1
15 gandalf.htm 26
7 grab.com/ 41

15	gandalf.htm	26

size of shared prefix

Node ID

Graph data structure



Web graph factoid

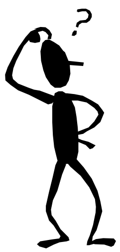
- >1B nodes (12/99), mean indegree is ~ 8
- *Zipfian Degree Distributions* [KRRT'99]:
  - $F_{in}(i)$ = fraction of pages with indegree *i*

$$F_{in}(i) \sim \frac{1}{i^{2.1}}$$

  - $F_{out}(i)$ = fraction of pages with outdegree *i*

$$F_{out}(i) \sim \frac{1}{i^{2.38}}$$

Google Open problems



- Graph compression: How much compression possible without significant run-time penalty?
  - Efficient algorithms to find frequently repeated small structures (e.g. wheels, $K_{2,2}$)
- External memory graph algorithms: How to assign the graph representation to pages so as to reduce paging? (see [NGV'96, AAMVV'98])
- Stringology: Less space for URL database? Faster algorithms for URL to node translation?
- Dynamic data structures: How to make updates efficient at the same space cost?

Algorithmic issues related to search engines

- Collecting documents
  - Priority
  - Load balancing
    - Internal
    - External
  - Trap avoidance
  - ...
- Processing and representing the data
  - Query-independent ranking
  - Graph representation
  - Index building
  - Duplicate elimination
  - Categorization
  - …
- Processing queries
  - Query-dependent ranking
  - Duplicate elimination
  - Query refinement
  - Clustering
  - ...

Index building

- ***Inverted index data structure***: Consider all documents concatenated into one huge document
  - For each word keep an ordered array of all positions in document, potentially compressed

Word 1	1st position	…	last position
⋮	⋮		⋮

- Allows efficient implementation of AND, OR, and AND NOT operations

Algorithmic issues related to search engines

- Collecting documents
  - Priority
  - Load balancing
    - Internal
    - External
  - Trap avoidance
  - ...
- Processing and representing the data
  - Query-independent ranking
  - Graph representation
  - Index building
  - Duplicate elimination
  - Categorization
  - …
- Processing queries
  - Query-dependent ranking
  - Duplicate elimination
  - Query refinement
  - Clustering
  - ...

Reasons for duplicates

- Proliferation of almost equal documents on the Web:
  - Legitimate: Mirrors, local copies, updates, etc.
  - Malicious: Spammers, spider traps, dynamic URLs
  - Mistaken: Spider errors
- Approximately 30% of the pages on the Web are (near) duplicates. [BGMZ'97,SG'98]

Uses of duplicate information

- Smarter crawlers
- Smarter web proxies
  - Better caching
  - Handling broken links
- Smarter search engines
  - no duplicate answers
  - smarter connectivity analysis
  - less RAM and disk

2 Types of duplicate filtering

- Fine-grain: Finding near-duplicate documents
- Coarse-grain: Finding near-duplicate hosts (*mirrors*)

Fine-grain: Basic mechanism

- Must filter both duplicate and near-duplicate documents
- Computing pair-wise edit distance would take forever
- Preferably to store only a short sketch for each document.

The basics of a solution

[B’97],[BGMZ’97]

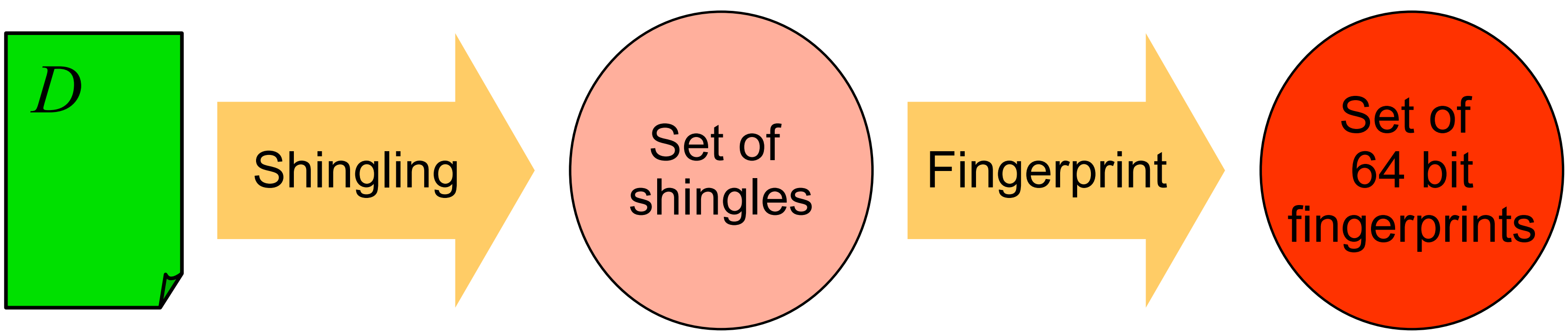
1. Reduce the problem to a set intersection problem

2. Estimate intersections by sampling minima.

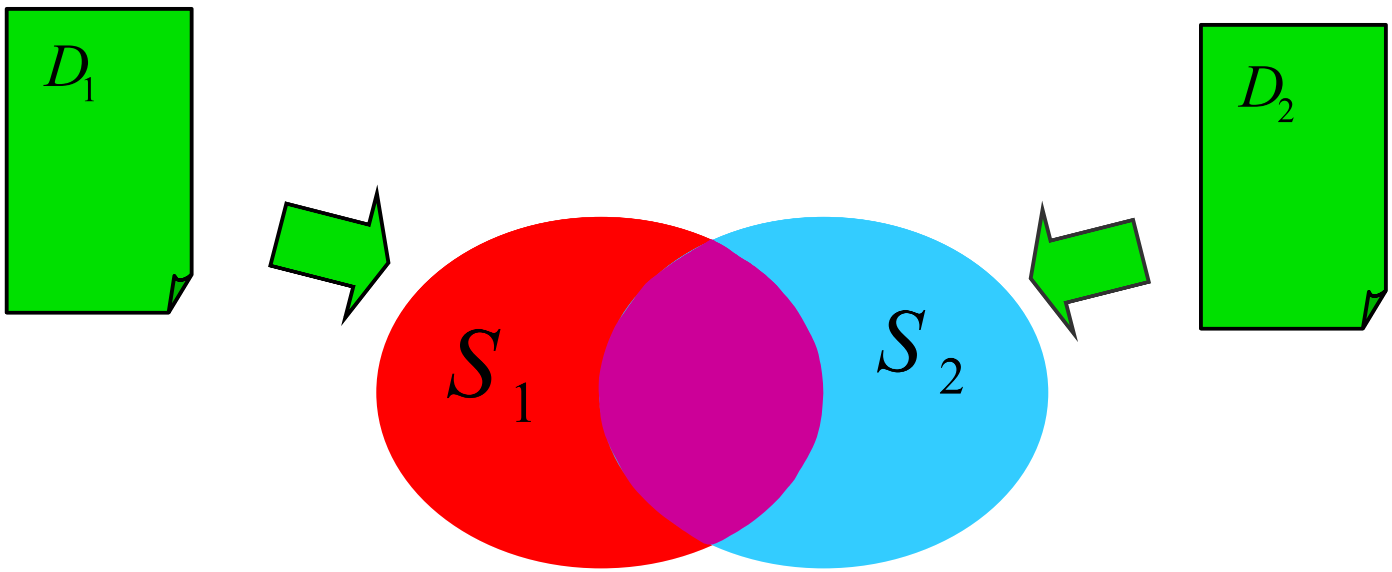
Shingling

- Shingle = Fixed size sequence of w contiguous words

```
a rose is a rose is a rose
a rose is a
  rose is a rose
       is a rose is
          a rose is a
            rose is a rose
```



Defining resemblance

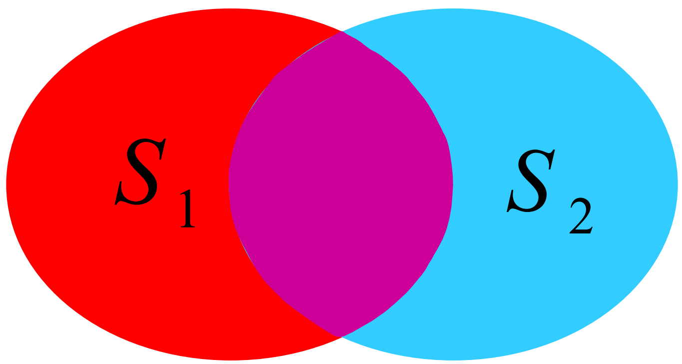


$$resemblance = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

Sampling minima

- Apply a random permutation σ to the set $[0..2^{64}]$
- Crucial fact

Let $\alpha = \min(\sigma(S_1)) \quad \beta = \min(\sigma(S_2))$



$$\Pr(\alpha = \beta) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

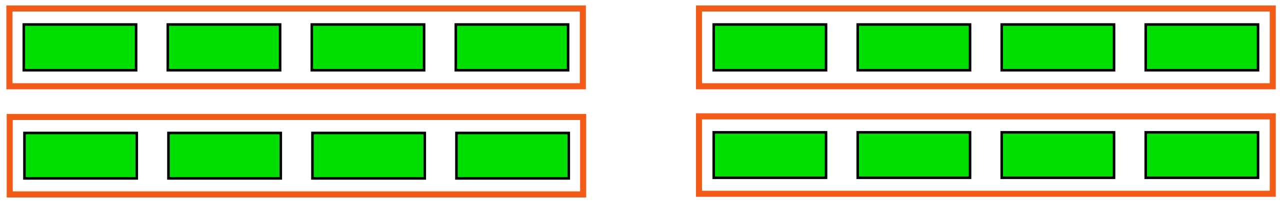
Implementation

- Choose a set of t random permutations of U
- For each document keep a sketch $S(D)$ consisting of t minima = samples
- Estimate resemblance of A and B by counting common samples
- The permutations should be from a min-wise independent family of permutations. See [BCFM'97] for the theory of mwi permutations.

If we need only high resemblance

Sketch 1:

Sketch 2:



- Divide sketch into k groups of s samples (t = k * s)
- Fingerprint each group ⇒ feature
- Two documents are fungible if they have at least r common features.
- Want
  Fungibility ⇔ Resemblance above fixed threshold ρ

Real implementation

- ρ = 90%. In a 1000 word page with shingle length = 8 this corresponds to
  - Delete a paragraph of about 50-60 words.
  - Change 5-6 random words.
- Sketch size $t = 84$, divide into $k = 6$ groups of $s = 14$ samples
- 8 bytes fingerprints $\rightarrow$ we store only $6 \times 8 =$

  48 bytes/document
- Threshold $r = 2$

Probability that two documents are deemed fungible

Two documents with resemblance ρ

- Using the full sketch

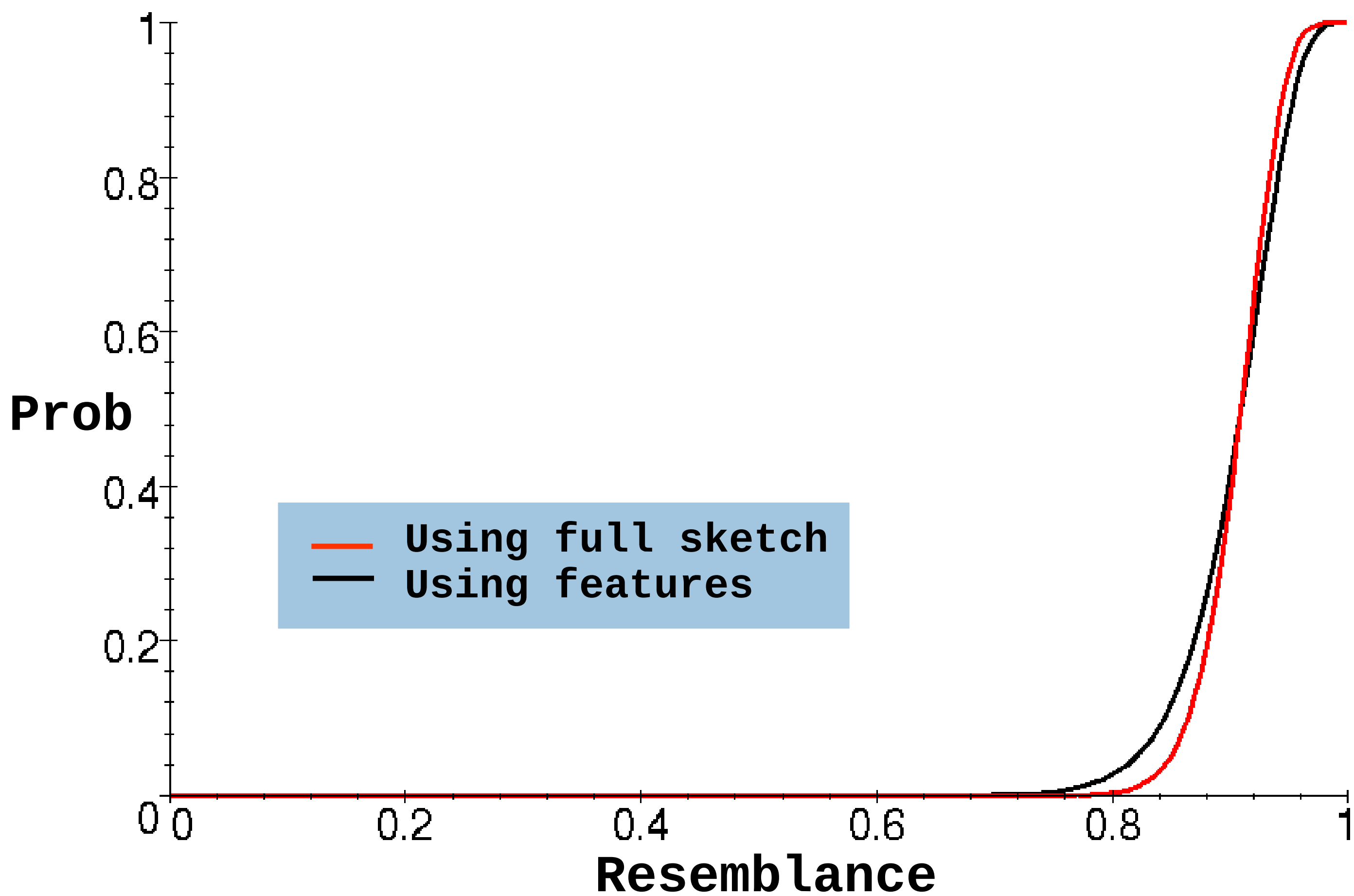
$$P = \sum_{i=r\cdot s}^{k\cdot s} \binom{k \cdot s}{i} \rho^{i} (1-\rho)^{k\cdot s-i}$$

- Using features

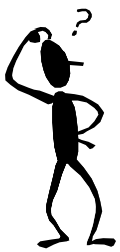
$$P = \sum_{i=r}^{k} \binom{k}{i} \rho^{s\cdot i} (1-\rho^{s})^{k-i}$$

Features vs. full sketch

Probability that two pages are deemed fungible



Fine-grain duplicate elimination: open problems and related work



- Best way of grouping samples for a given threshold and/or for multiple thresholds?
- Efficient ways to find in a data base pairs of records that share many attributes. Best approach?
- Min-wise independent permutations -- lots of open questions.
- Other applications possible (images, sounds, ...) -- need translation into set intersection problem.

- Related work: M'94, BDG'95, SG'95, H'96, FSGMU'98

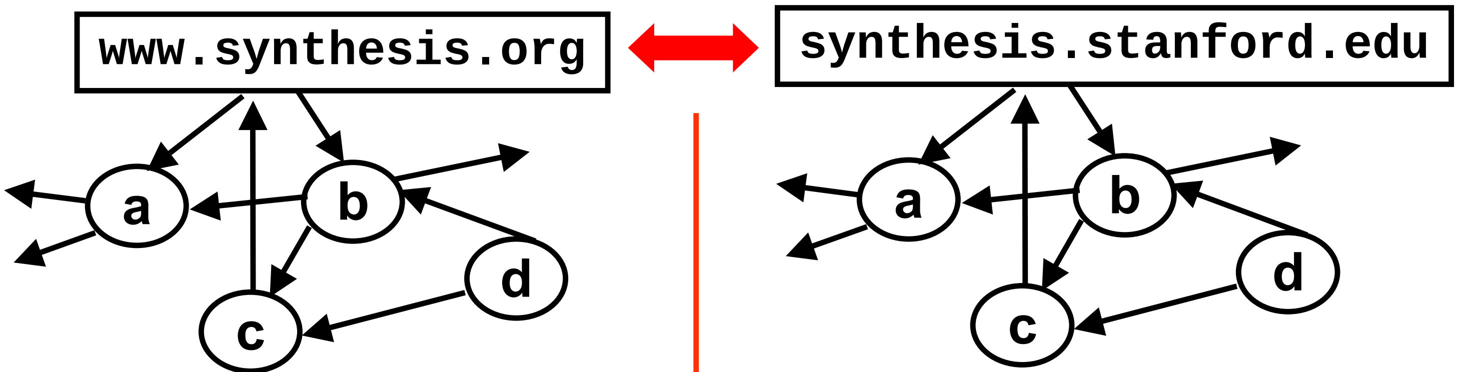
2 Types of duplicate filtering

- Fine-grain: Finding near-duplicate documents
- Coarse-grain: Finding near-duplicate hosts (*mirrors*)

Input: set of URLs

- Input:
  - Subset of URLs on various hosts, collected e.g. by search engine crawl or web proxy
  - No content of pages pointed to by URLs except each page is labeled with its out-links
- Goal: Find pairs of hosts that mirror content

Example



www.synthesis.org/Docs/ProjAbs/synsys/synalysis.html
www.synthesis.org/Docs/ProjAbs/synsys/visual-semi-quant.html
www.synthesis.org/Docs/annual.report96.final.html
www.synthesis.org/Docs/cicee-berlin-paper.html
www.synthesis.org/Docs/myr5
www.synthesis.org/Docs/myr5/cicee/bridge-gap.html
www.synthesis.org/Docs/myr5/cs/cs-meta.html
www.synthesis.org/Docs/myr5/mech/mech-intro-mechatron.html
www.synthesis.org/Docs/myr5/mech/mech-take-home.html
www.synthesis.org/Docs/myr5/synsys/experiential-learning.html
www.synthesis.org/Docs/myr5/synsys/mm-mech-dissec.html
www.synthesis.org/Docs/yr5ar
www.synthesis.org/Docs/yr5ar/assess
www.synthesis.org/Docs/yr5ar/cicee
www.synthesis.org/Docs/yr5ar/cicee/bridge-gap.html
www.synthesis.org/Docs/yr5ar/cicee/comp-integ-analysis.html

synthesis.stanford.edu/Docs/ProjAbs/deliv/high-tech-classroom.html
synthesis.stanford.edu/Docs/ProjAbs/mech/mech-enhanced-circ-anal.html
synthesis.stanford.edu/Docs/ProjAbs/mech/mech-intro-mechatron.html
synthesis.stanford.edu/Docs/ProjAbs/mech/mech-mm-case-studies.html
synthesis.stanford.edu/Docs/ProjAbs/synsys/quant-dev-new-teach.html
synthesis.stanford.edu/Docs/annual.report96.final.html
synthesis.stanford.edu/Docs/annual.report96.final_fn.html
synthesis.stanford.edu/Docs/myr5/assessment
synthesis.stanford.edu/Docs/myr5/assessment/assessment-main.html
synthesis.stanford.edu/Docs/myr5/assessment/mm-forum-kiosk-A6-E25.html
synthesis.stanford.edu/Docs/myr5/assessment/neato-ucb.html
synthesis.stanford.edu/Docs/myr5/assessment/not-available.html
synthesis.stanford.edu/Docs/myr5/cicee
synthesis.stanford.edu/Docs/myr5/cicee/bridge-gap.html
synthesis.stanford.edu/Docs/myr5/cicee/cicee-main.html
synthesis.stanford.edu/Docs/myr5/cicee/comp-integ-analysis.html

Coarse-grain: Basic mechanism

- Must filter both duplicate and near-duplicate mirrors
- Pair-wise testing would take forever
- Both high precision (not outputting wrong mirrors) and high recall (finding almost all mirrors) are important

A definition of mirroring

Host1 and Host2 are mirrors iff

For all paths p such that

`http://Host1/p`

is a web page,

`http://Host2/p`

exists with duplicate (or *near-duplicate*) content,

and vice versa.

The basics of a solution

[BBDH'99]

1. Pre-filter to create a small set of pairs of potential mirrors (*pre-filtering step*)

2. Test each pair of potential mirrors (*testing step*)

3. Use different pre-filtering algorithms to improve recall

Testing step

- Test root pages + *x* URLs from each host sample
- **If** one test returns "not near-duplicate"
  then hosts are *not mirrors*
- **If** root pages and > *c$ x* URLs from each host sample are near-identical
  then hosts are *mirrors*,
  else they are *not mirrors*

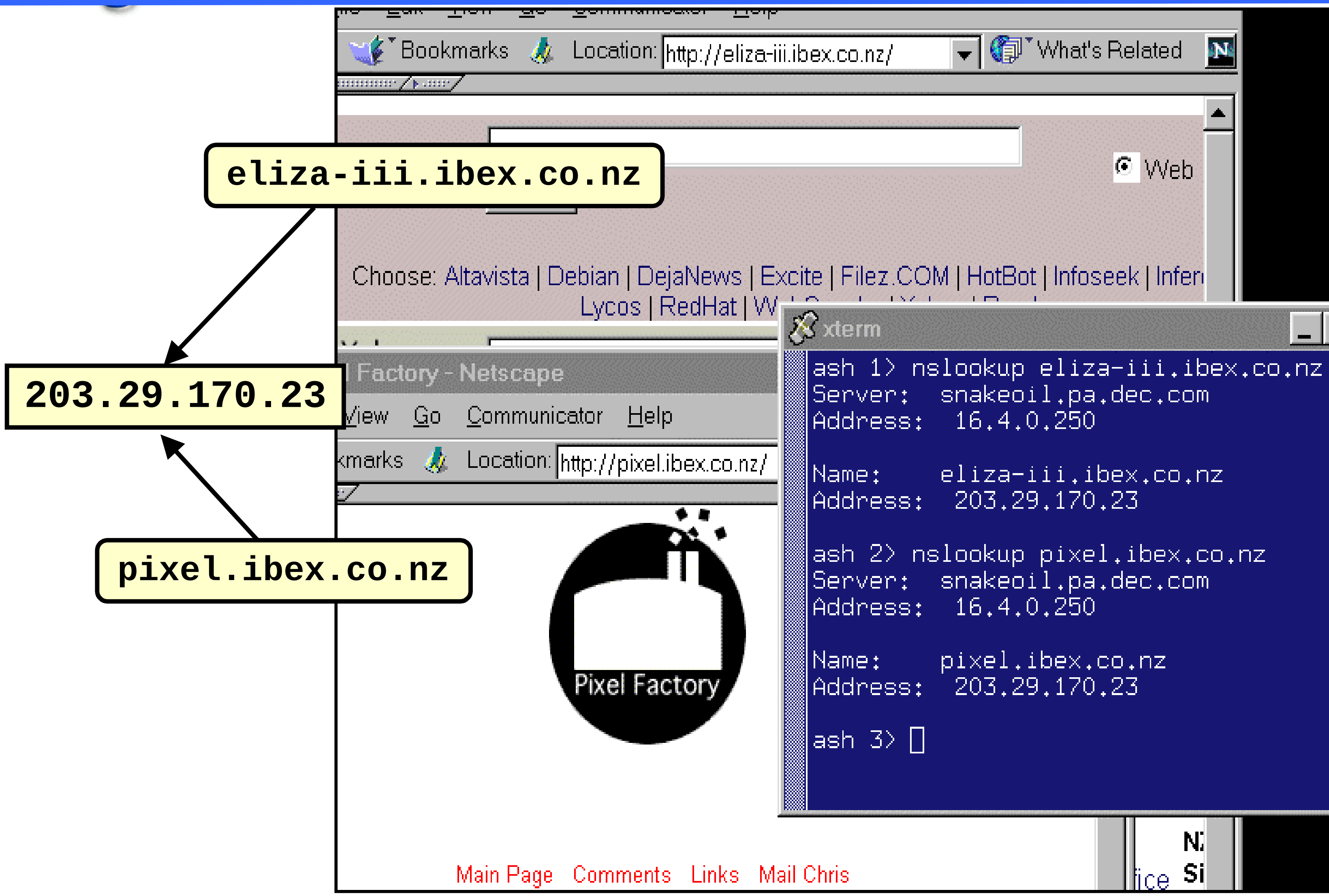
Pre-filtering step

- Goal: Output **quickly** list of pairs of potential mirrors containing
  - many true mirror pairs (high recall)
  - not many non-mirror pairs (high precision)
- Note: 2-sided error is allowed
  - Type-1: true mirror pairs might be missing in output
  - Type-2: non-mirror pair might be output
- Testing of host pairs will eliminate type-2 errors, but not type-1 errors

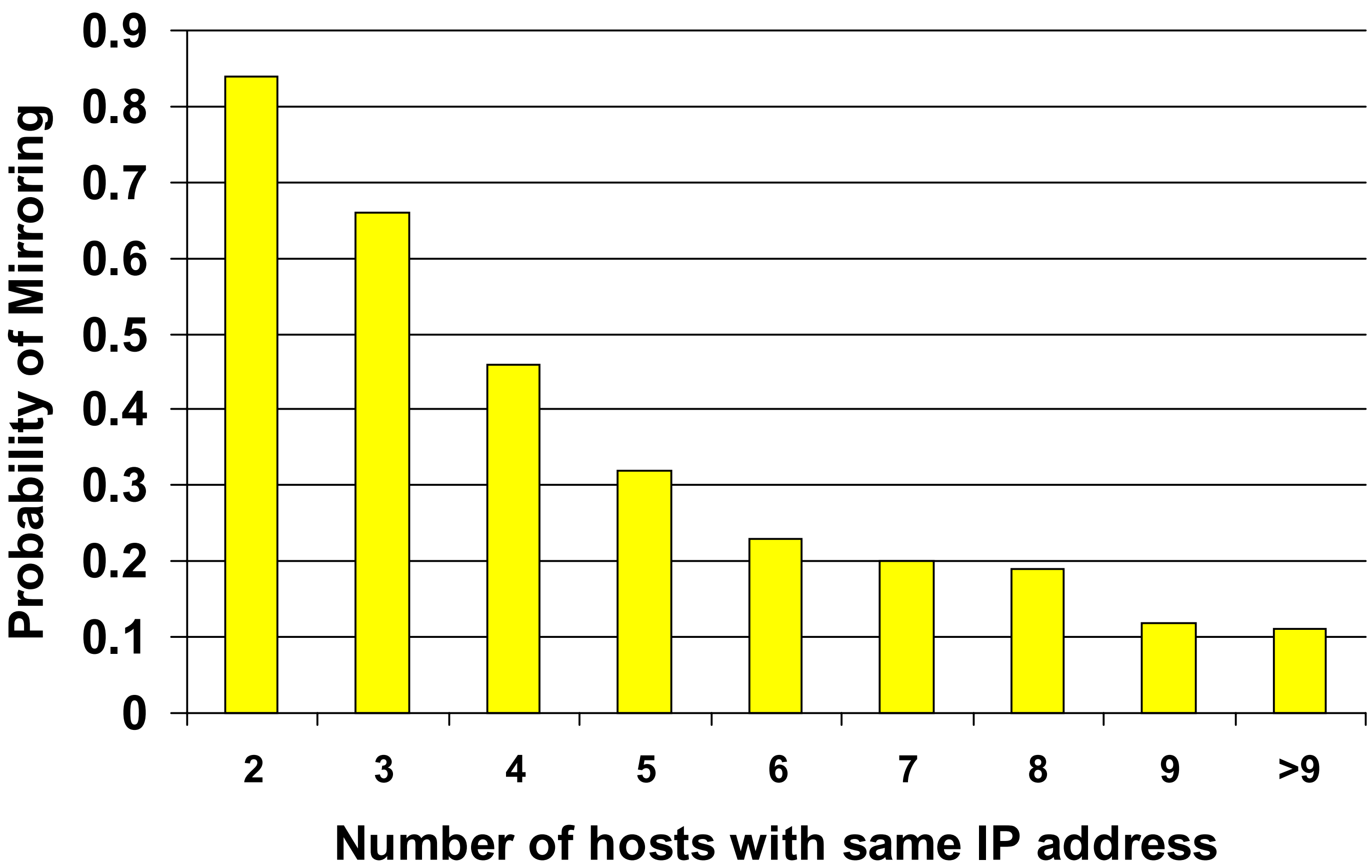
Different pre-filtering techniques

- IP-based
- URL-string based
- URL-string and hyperlink based
- Hostname and hyperlink based

Problem with IP addresses



Number of host with same IP address vs mirror probability



IP based pre-filtering algorithms

- *IP4:* Cluster hosts based on IP address
  - Enumerate pairs from clusters in increasing cluster size (max 200 pairs)
- *IP3:* Cluster hosts based on first 3 octets of their IP address
  - Enumerate pairs from clusters in increasing cluster size (max 5 pairs)

URL string based pre-filtering algorithms

Information extracted from URL strings:

- Similar hostnames: might belong to same organization
- Similar paths: might have replicated directories
- extract “features” for host from URL strings and test similarity

Similarity Testing Approach:

- *Feature vector* for each host similar to term vector for document:
  - Host corresponds to document
  - Feature corresponds to term
- Similarity of hosts = Cosine of angle of feature vectors

URL string based algorithms (cont.)

- *paths*: Features are paths: e.g.,
  /staff/homepages/~dilbert/foo
- *prefixes:* Features are prefixes: e.g.,
  /staff
  /staff/homepages
  /staff/homepages/~dilbert
  /staff/homepages/~dilbert/foo
- Other variants: *hosts* and *shingles*

Paths + connectivity (*conn*)

- Take output from *paths* and filter thus:
  - Consider 10 common paths in sample with highest outdegree
  - Paths are **equivalent** if 90% of their combined out-edges are common to both
  - Keep host-pair if 75% of the paths are equivalent

Hostname connectivity

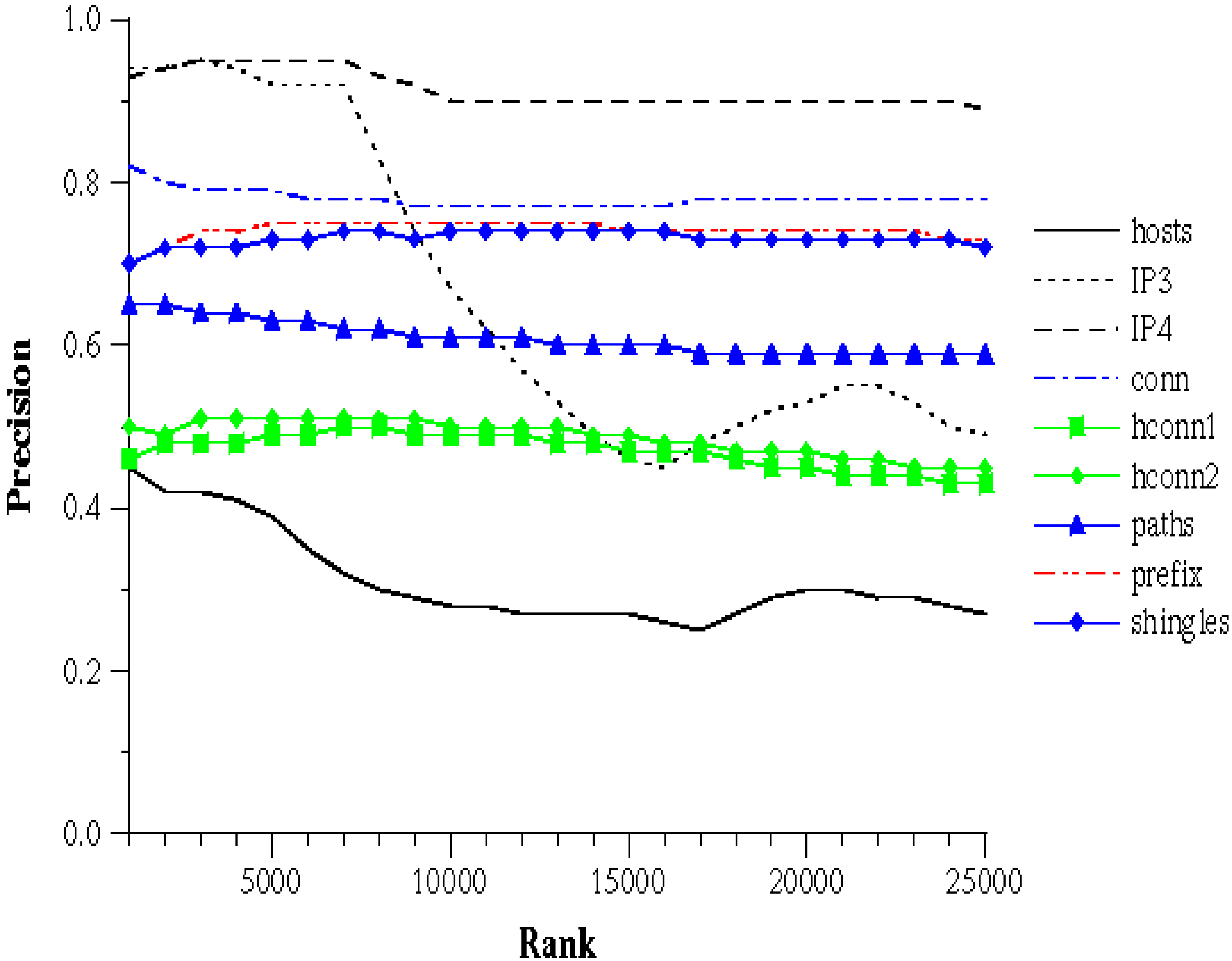
- Idea: Mirrors point to similar set of other hosts
- Feature vector approach to test similarity:
  - features are hosts that are pointed to
  - 2 different ways of feature weighting:
    - *hconn1*
    - *hconn2*

Experiments

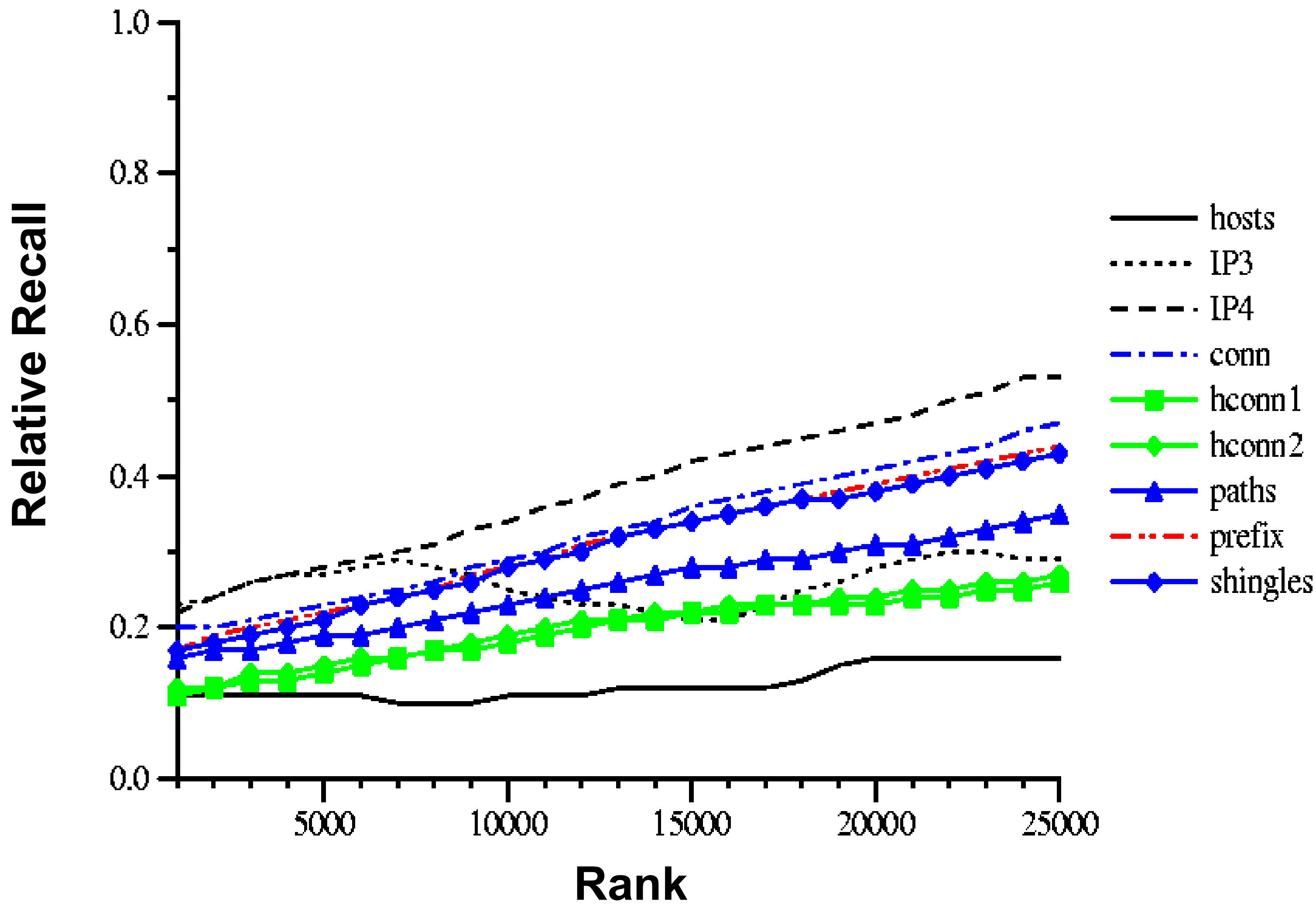
- Input: 140 million URLs on 233,035 hosts + out-edges
  - Original 179 million URLs reduced by considering only hosts with at least 100 URLs in set
- For each of the above pre-filtering algorithms:
  - Compute list of 25,000 (100,000) ranked pairs of potential mirrors
  - Test each pair of potential mirrors (testing step) and output list of mirrors

  Determine precision and *relative* recall

Precision up to rank 25,000



Relative recall up to rank 25,000



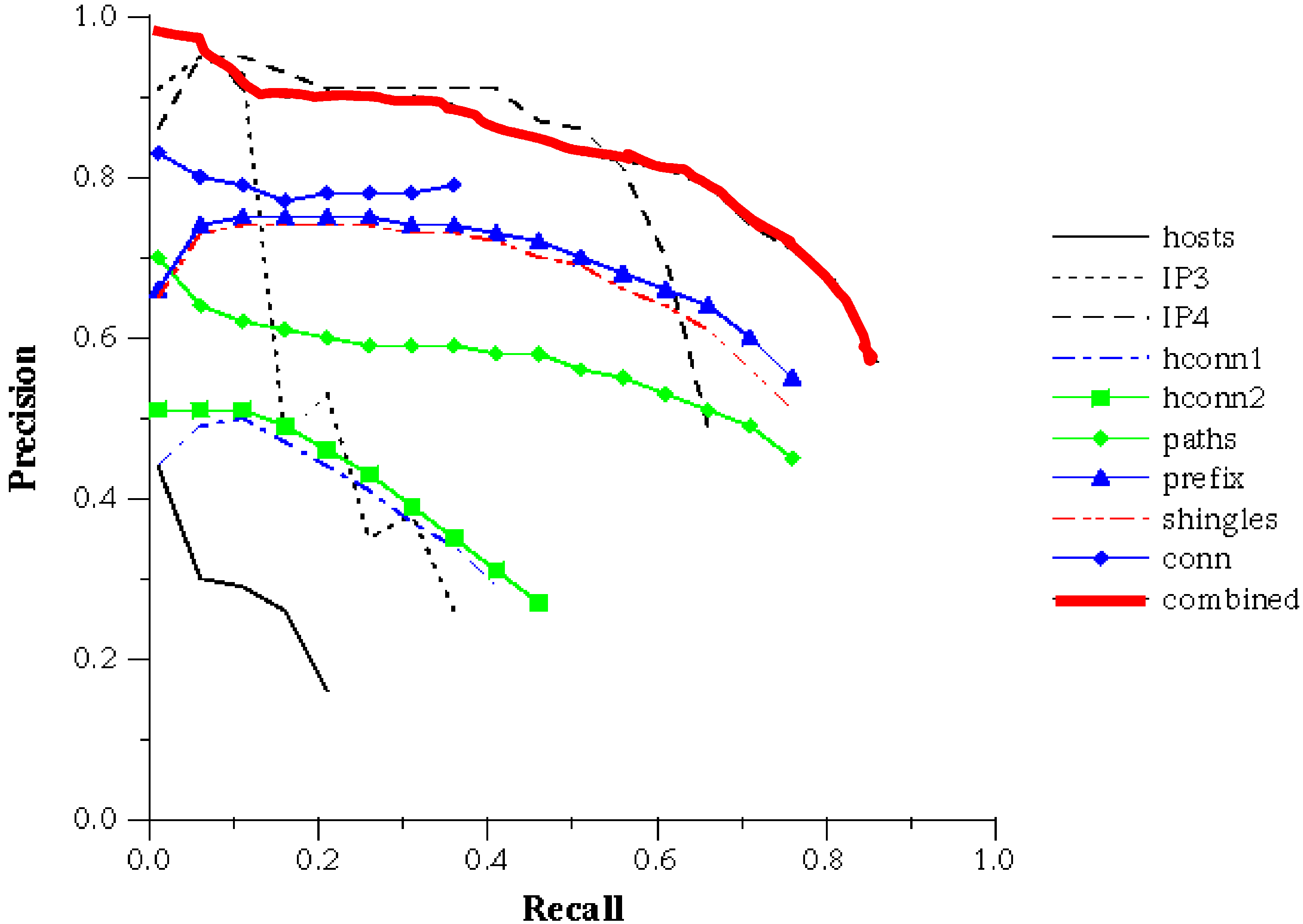
Relative recall at 25,000 for combined output

	hosts	IP3	IP4	conn	hconn1	hconn2	paths	prefix	shingles
hosts	17%								
IP3	39%	30%							
IP4	61%	58%	54%						
conn	58%	66%	80%	47%					
hconn1	40%	51%	69%	59%	26%				
hconn2	41%	52%	70%	60%	29%	27%			
paths	48%	59%	78%	55%	51%	52%	36%		
prefix	54%	61%	75%	65%	58%	58%	57%	44%	
shingles	53%	61%	75%	64%	57%	58%	57%	48%	44%

Combined approach (*combined*)

- Combines top 100,000 results from *hosts, IP4, paths, prefix,* and *hconn1*.
- Sort host pairs by:
  - Number of algorithms that return the host pair
  - Use best rank for any algorithm to break ties between host pairs
- At rank 100,000: relative recall of 86%, precision of 57%

Precision vs relative recall



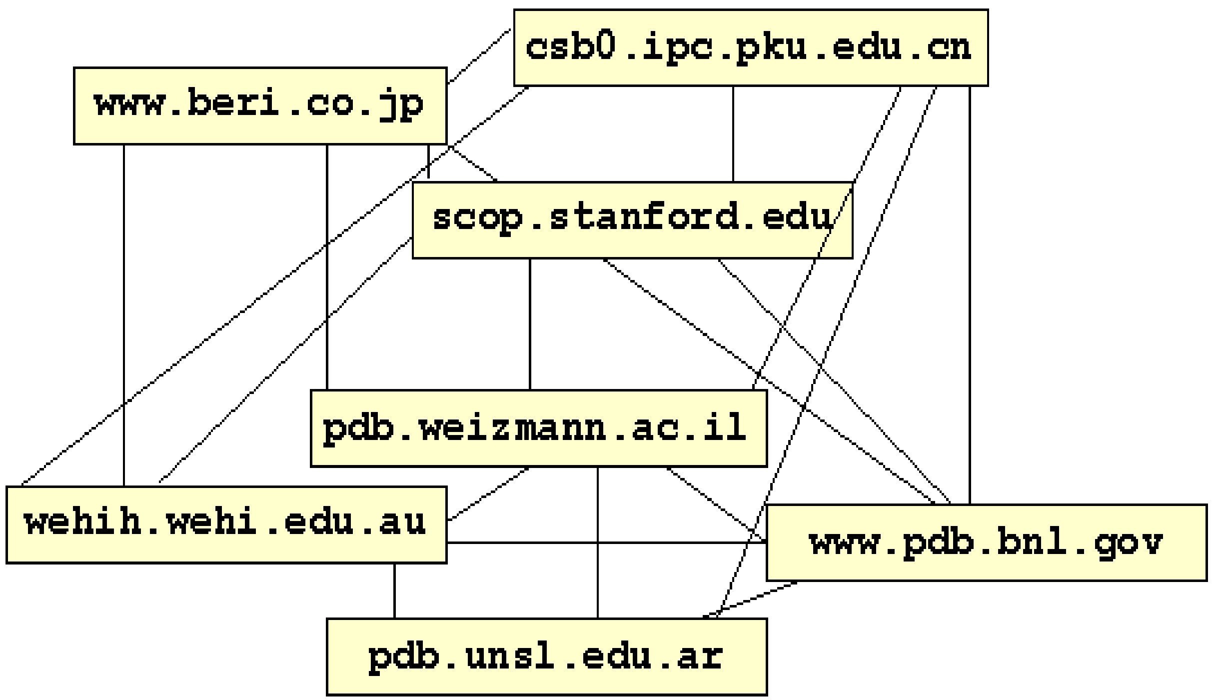
Web host graph

- A node for each host *h*
- An undirected edge (*h,h'*) if *h* and *h'* are output as mirrors

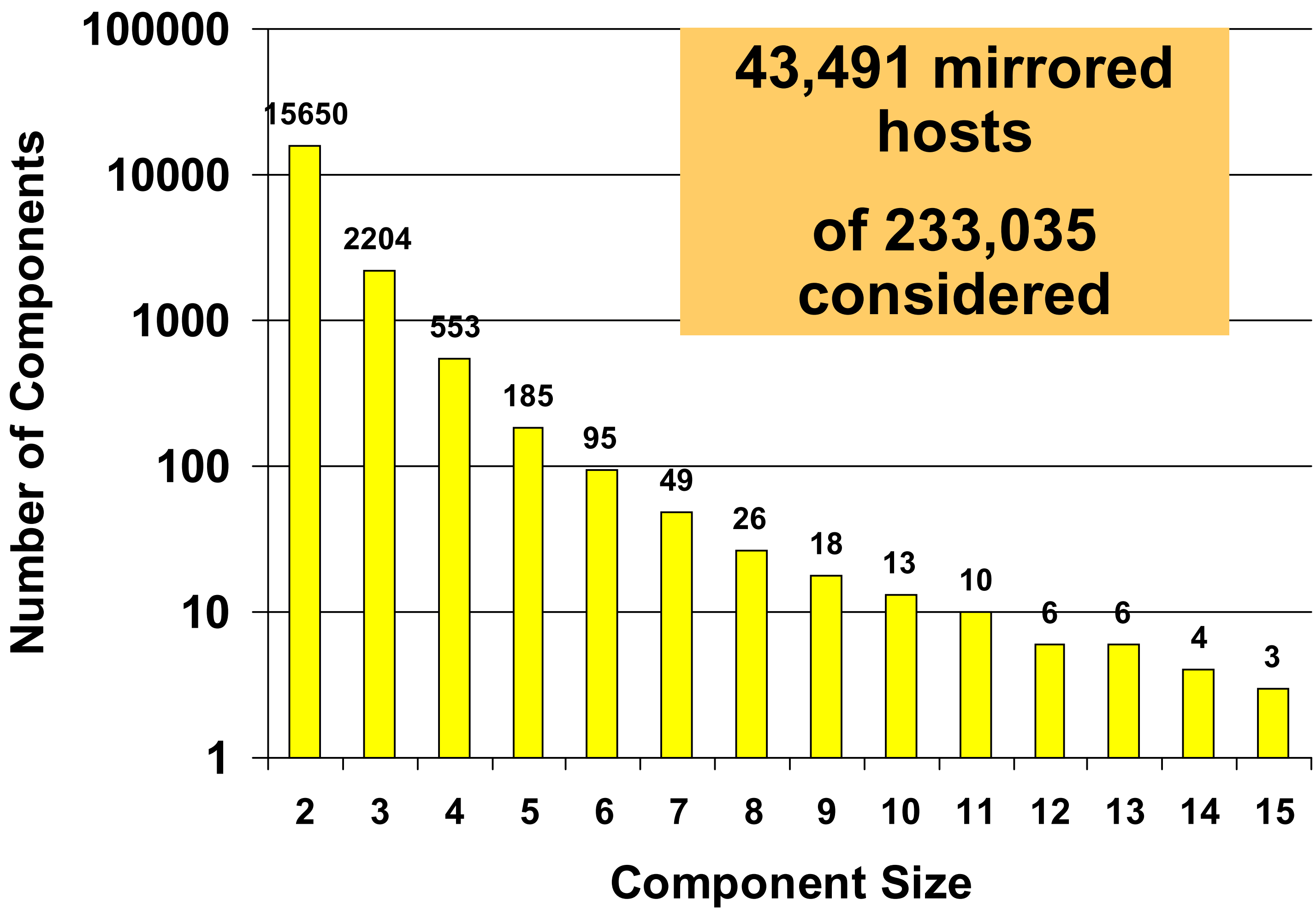
- Each (connected) component gives a set of mirrors

Example of a component

Protein Data Bank



Component size distribution



Coarse-grain duplicate filtering: Summary and open problems

- Mirroring is common (43,491 mirrored hosts out of 233,035 considered hosts)
  - Load balancing, franchises/branding, virtual hosting, spam
- Mirror detection based on non-content attributes is feasible.
- [CSG'00] use page content similarity based approach. Open Problem: Compare and combine content and non-content techniques.
- Open Problem: Assume you can choose which URLs to visit at a host. Determine best technique.

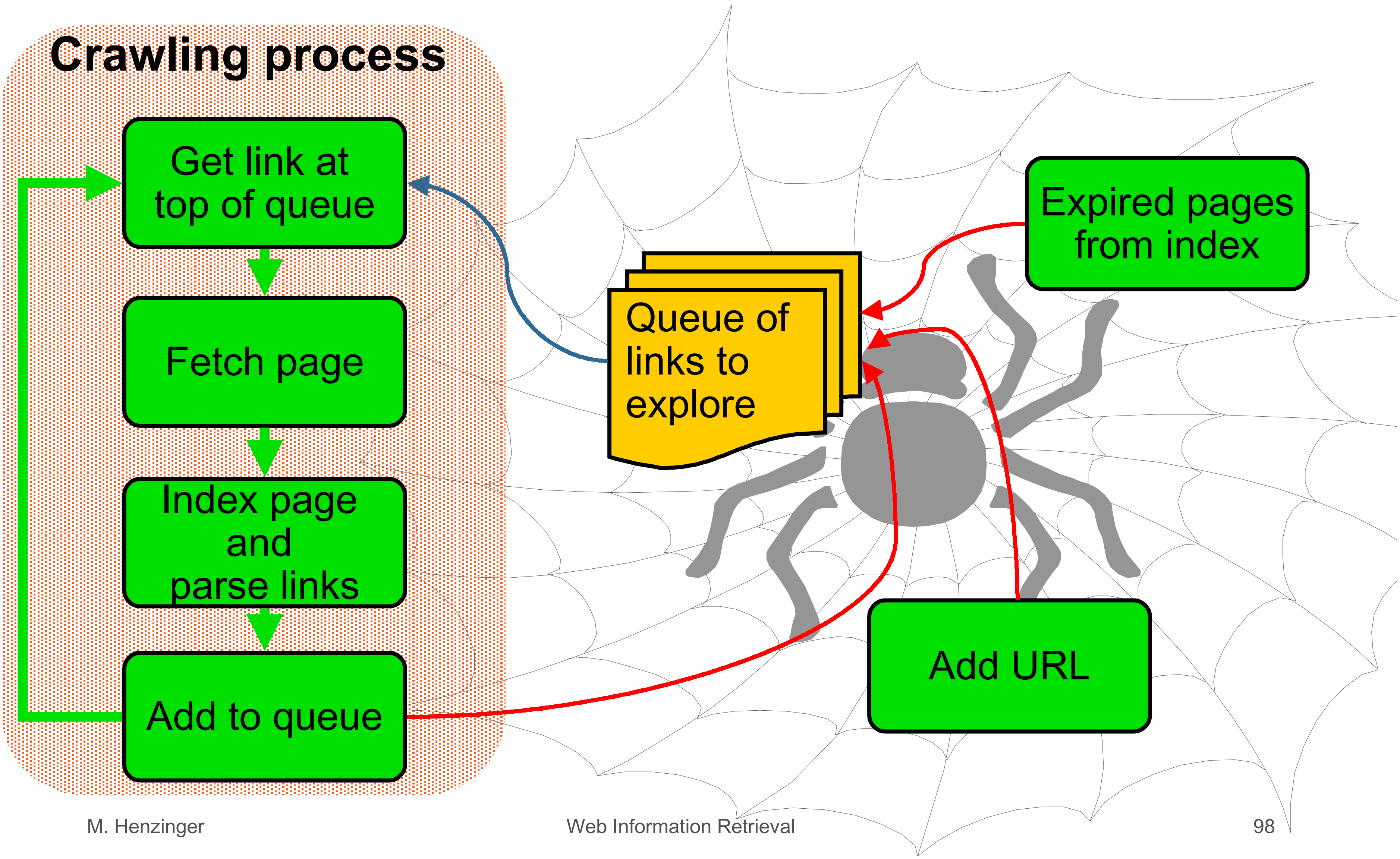
Algorithmic issues related to search engines

- Collecting documents
  - Priority
  - Load balancing
    - Internal
    - External
  - Trap avoidance
  - ...

- Processing and representing the data
  - Query-independent ranking
  - Graph representation
  - Index building
  - Duplicate elimination
  - Categorization
  - …

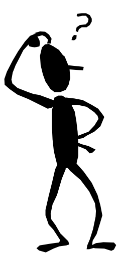
- Processing queries
  - Query-dependent ranking
  - Duplicate elimination
  - Query refinement
  - Clustering
  - ...

Adding pages to the index



Queuing discipline

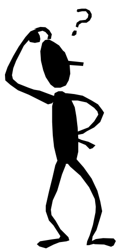
- Standard graph exploration:
  - Random
  - BFS
  - DFS (+ depth limits)
- **Goal:** get “best” pages for a given index size
  - Priority based on query-independent ranking:
    - highest indegree [M’95]
    - highest potential PageRank [CGP’98]



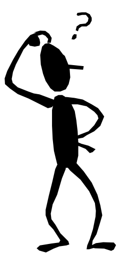
- **Goal:** keep index fresh
  - Priority based on rate of change [CLW’97]

Google Load balancing

- Internal -- can not handle too much retrieved data simultaneously, but
  - Response time is unpredictable
  - Size of answers is unpredictable
  - There are additional system constraints (# threads, # open connections, etc.)



- External
  - Should not overload any server or connection
  - A well-connected crawler can saturate the entire outside bandwidth of some small countries
  - Any queuing discipline must be acceptable to the community



Web IR Tools

- General-purpose search engines
- Hierarchical directories
- Specialized search engines
  (dealing with heterogeneous data sources)
- Search-by-example
- Collaborative filtering
- Meta-information

Hierarchical directories

Building of hierarchical directories:

- Manual: Yahoo!, LookSmart, Open Directory
- Automatic:
  - Populating of hierarchy [CDRRGK'98]: For each node in the hierarchy formulate fine-tuned query and run modified HITS algorithm
  - Categorization: For each document find "best" placement in the hierarchy. Techniques are connectivity and/or text based [CDI'98, …]

Web IR Tools

- General-purpose search engines
- Hierarchical directories
- Specialized search engines
(dealing with heterogeneous data sources)
  - Shopping robots
  - Home page finder [SLE'97]
  - Applet finders
  - …
- Search-by-example
- Collaborative filtering
- Meta-information

Dealing with heterogeneous sources

- Modern life problem:

> Given information sources with various capabilities, query all of them and combine the output.

- Examples
  - Inter-business e-commerce e.g. **`www.industry.net`**
  - Meta search engines
  - Shopping robots
- Issues
  - Determining relevant sources -- the "identification" problem
  - Merging the results -- the "fusion" problem

Example: a shopping robot

- Input: A product description in some form
  Find: Merchants for that product on the Web
- Jango [DEW'97]
  - preprocessing: Store vendor URLs in database; learn for each vendor:
    - the URL of the search form
    - how to fill in the search form and
    - how the answer is returned
  - request processing: fill out form at every vendor and test whether the result is a success
  - range of products is predetermined

Jango input example

Jango output

Excite Product Finder

powered by Jango

Wine - Products — Get Reviews · New Wine Search

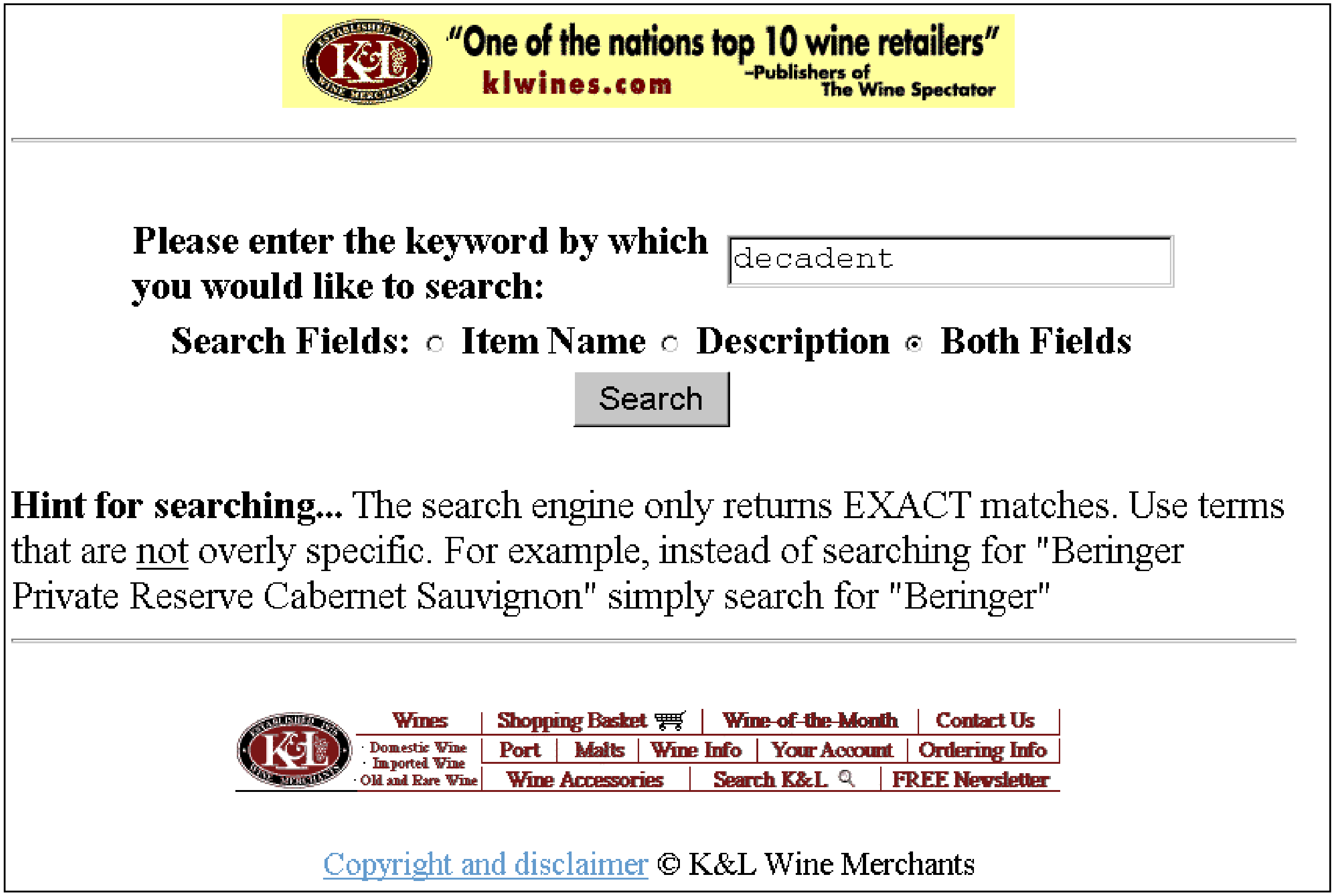
Your Search: Variety = "viognier"

Instructions: **Click a column title to sort results** by the information in that column. For more details on a particular wine, click on a link in the Name column.

Search Results: 15 items have been located. Click here for a search summary.

Winery	Variety	Name	Year	Quantity	Store	Price	
	Viognier	Alban Vineyards Estate Viognier	97	1 bottle	K&L Wine Merchants	$22.95	Buy!
Arrowood Vineyards & Winery	Viognier	Arrowood Viognier Saralee's Vineyard	96	1 bottle	California Wine	$28.00	Buy!
Calera	Viognier	Calera Mt. Harlan	95	1 bottle	Taylor & Norton	$26.99	Buy!
Calera	Viognier	Calera Mt. Harlan	97	1 bottle	Taylor & Norton	$27.49	Buy!
	Viognier	Chance Creek Viognier	97	1 bottle	K&L Wine Merchants	$13.99	Buy!
	Viognier	Gregory Graham	97	1 bottle	Taylor & Norton	$19.99	Buy!

Direct query to K&L Wine



What price decadence?

1983 Pichon Lalande **Qnty. Avail.:** 3 **Bottle:** $129.00

94 points from Parker… 'Consistently one of the great wines of the 1983 vintage, as well as one of my personal favorites, this beautiful wine has been gorgeous to drink since bottling. It displays no signs of evolution, although it remains undeniably rich, seductive, and compelling. Deep dark ruby-colored, with a huge nose of Asian spices, blackcurrants, plums, and flowers, this super-concentrated, velvety-textured wine reveals gobs of rich, creamy fruit. It can be drunk now or cellared for 15-20 years. It is Pauillac at its most **DECADENT** and seductive! '

1961 Palmer, Margaux **Qnty. Avail.:** 5 **Bottle:** $699.00

99 points from Robert Parker… 'The 1961 Palmer has long been considered to be a legend from this vintage, and its reputation is well-deserved. The wine is at its apogee, with an extraordinary, sweet, complex nose with aromas of flowers, cassis, toast, and minerals. It is intensely concentrated, offering a cascade of lavishly ripe, full-bodied, opulent fruit, soft tannins, and a voluptuous finish. This is a **DECADENT** Palmer, unparalleled since in quality with the exception of 1983 and 1989.'

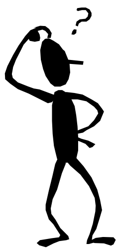
1996 Charmes-Chambertin, Jean Raphet **Bottle:** $79.95

Top Pick! The obvious standout in a super lineup of Raphet wines. Very tasty and very limited. 95 points from the Wine Advocate (Pierre Rovanni)… 'Extraordinary. Medium-to-dark rub-colored, its awesome aromatics reveal fruit cake, cinnamon, spicy chutney , and assorted red fruits. This is a sexy, intense, **DECADENT**, and full-bodied gem crammed with loads of sweet cherries, perfume, flowers and spices… the mouth0watering flavors continue to coat the palate for what seems like minutes.'

1993 Haut-Marbuzet, St-Estephe **Bottle:** $19.95

Ripe cherry-berry scents in the nose are followed by a delicious, lush, elegant wine that gently flows onto the palate. Hints of cedar and tobacco. Robert Parker... 'Haut-Marbuzet is one of the oldest estates in St.-Estephe, but its fame can be traced only to 1952, when it was purchased by the father of the current proprietor, Henri Duboscq. No one argues with the success proprietor Duboscq has enjoyed. His wine is a Bordeaux that behaves more like a **DECADENT** Burgundy or Rhone.'

Open problems



- Going beyond the lowest common capability
- Learning problem: automatic understanding of new interfaces
- Efficient ways to determine which DB is relevant to a particular query
- “Partial knowledge indexing”: indexer has limited access to the full DB

Web IR Tools

General-purpose search engines

Hierarchical directories

Specialized search engines
(dealing with heterogeneous data sources)

- Search-by-example
- Collaborative filtering
- Meta-information

Search-by-example

- Given: set S of URLs
- Find: URLs of similar pages
- Various Approaches:
  - Connectivity-based
  - Usage based: related pages are pages visited frequently after S
  - Query refinement
- Related Work:G'72, G'79, K'98, PP'97,S'97, CDI'98

Output from Google: related:www.ebay.com

Fannie Mae's Homepath.com - Your On-Line Path to a Home of Your Own
Fannie Mae's consumer Web site provides comprehensive information on buying and refinancing a home. Homebuyers can find -- Welcome to HomePath &re; -- a site that will help yo...
www.homepath.com/ Cached (7k) New! Try out GoogleScout

Today's Mortgage Information from HSH Associates, Financial Publishers
HSH Associates, the world's leading publisher of mortgage and consumer loan information, surveys current loan rates from 2,500 to 3,000 lenders throughout the US. We offer dai...
www.hsh.com/ Cached (15k) New! Try out GoogleScout

Countrywide Home Loans
The nation's largest independent mortgage lender. Pre-qualify for your potential maximum loan amount based on current interest rates and loan products with the Home Loan ...
www.countrywide.com/ Cached (7k) New! Try out GoogleScout

Keystroke Loans
Home loans at the best interest rates from the Web's leading mortgage loan broker - Keystroke.com -- July 8, 1999 Check out our Mortgage Rates for purchase and refinance quote...
www.keystrokenet.com/ Cached (8k) New! Try out GoogleScout

www.iqualify.com/
New! Try out GoogleScout

Output from Alexa: www.ebay.com

Netscape
Netcenter

You are here: Home > What's Related

What's Related

by Alexa

...to **http://www.eloan.com/**

1. Online Mortgage
2. Countrywide Home Loans
3. American Finance On Line
4. Keystroke Loans
5. Capital Mortgage Services, Inc.
6. Business Week
7. Chase Manhattan Mortgage Corporation
8. Home Loans
9. HomeByNet Home Page
10. 1003 LOAN APPLICATION - APPLICATION FORMS - Mortgage broker, loan, interest
11. Learn About Smart Browsing...

Connectivity based solutions

[DH'99]

- Algorithm Companion

- Algorithm Co-citation

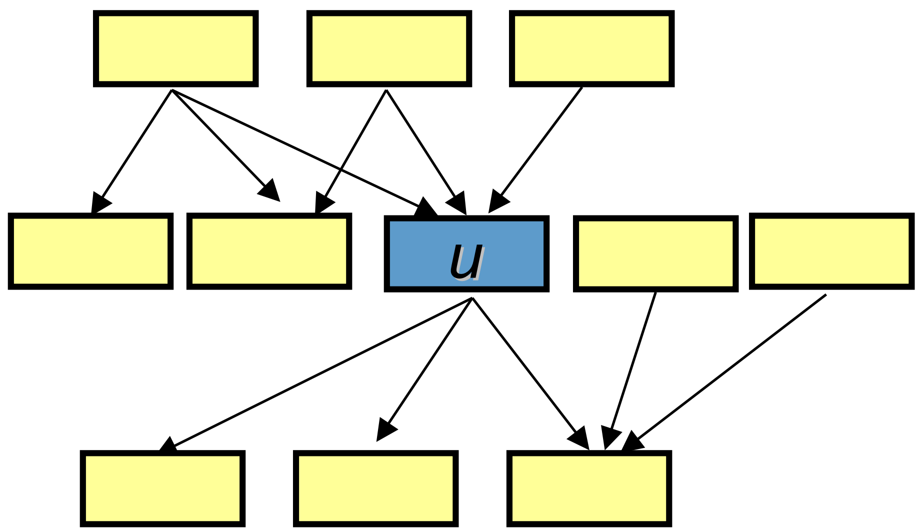
Algorithm Companion

- Build modified neighborhood graph *N*.
- Run modified HITS algorithm on *N*.

Major Question: How to form neighborhood graph s.t. top returned pages are useful related pages

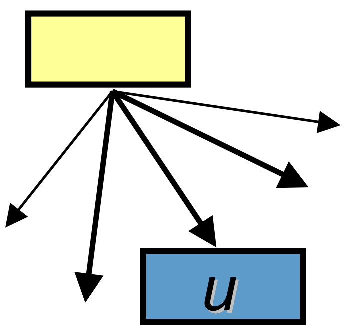
Building neighborhood graph *N*

- Node set: From URL *u* go 'back', 'forward', 'back-forward', and 'forward-back'
- Edge set: Directed edge if there is a hyperlink between 2 nodes
- Apply refinements to *N*



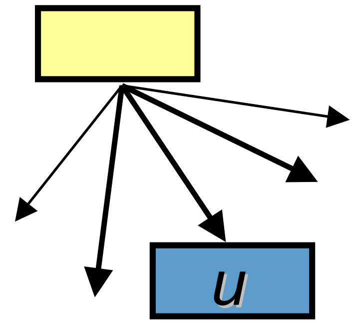
Refinement 1: Limit out-degree

- Motivation: Some nodes have high out-degree => graph would become too large
- Limit out-degree when going "forward"
  - Going forward from *u* : choose first 50 out-links on page
  - Going forward from other nodes: choose 8 out-links surrounding the in-link traversed to reach *u*



Co-citation algorithm

- Determine 2000 arbitrary back-nodes of *u.*
- Add to set *S* of siblings of *u*:
  For each back node 8 forward-nodes
  surrounding the link to *u*



- If there is enough co-citation with *u* then
  – return nodes in S in decreasing order of co-citations

  else

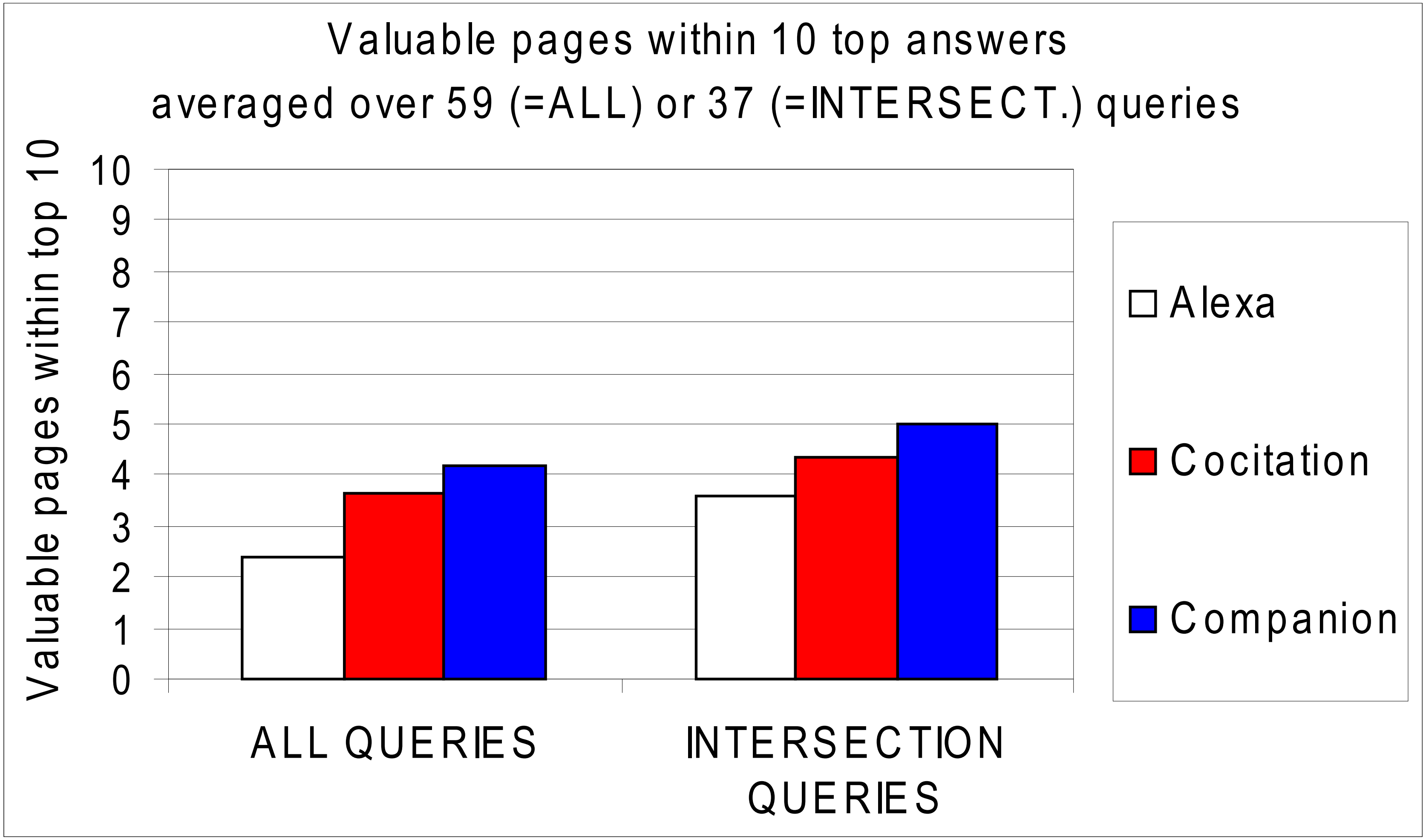
  – restart algorithm with one path element removed (http://…/X/Y/ -> http://…/X/)

Alexa's "What's Related"

- Uses:
  - Document Content
  - Usage Data
  - Connectivity

- Removes path elements if no answer for *u* is found

User study



Valuable pages within 10 top answers
averaged over 59 (=ALL) or 37 (=INTERSECT.) queries
Valuable pages within top 10
10
9
8
7
6
5
4
3
2
1
0
ALL QUERIES
INTERSECTION QUERIES
Alexa
Cocitation
Companion

Web IR Tools

General-purpose search engines:

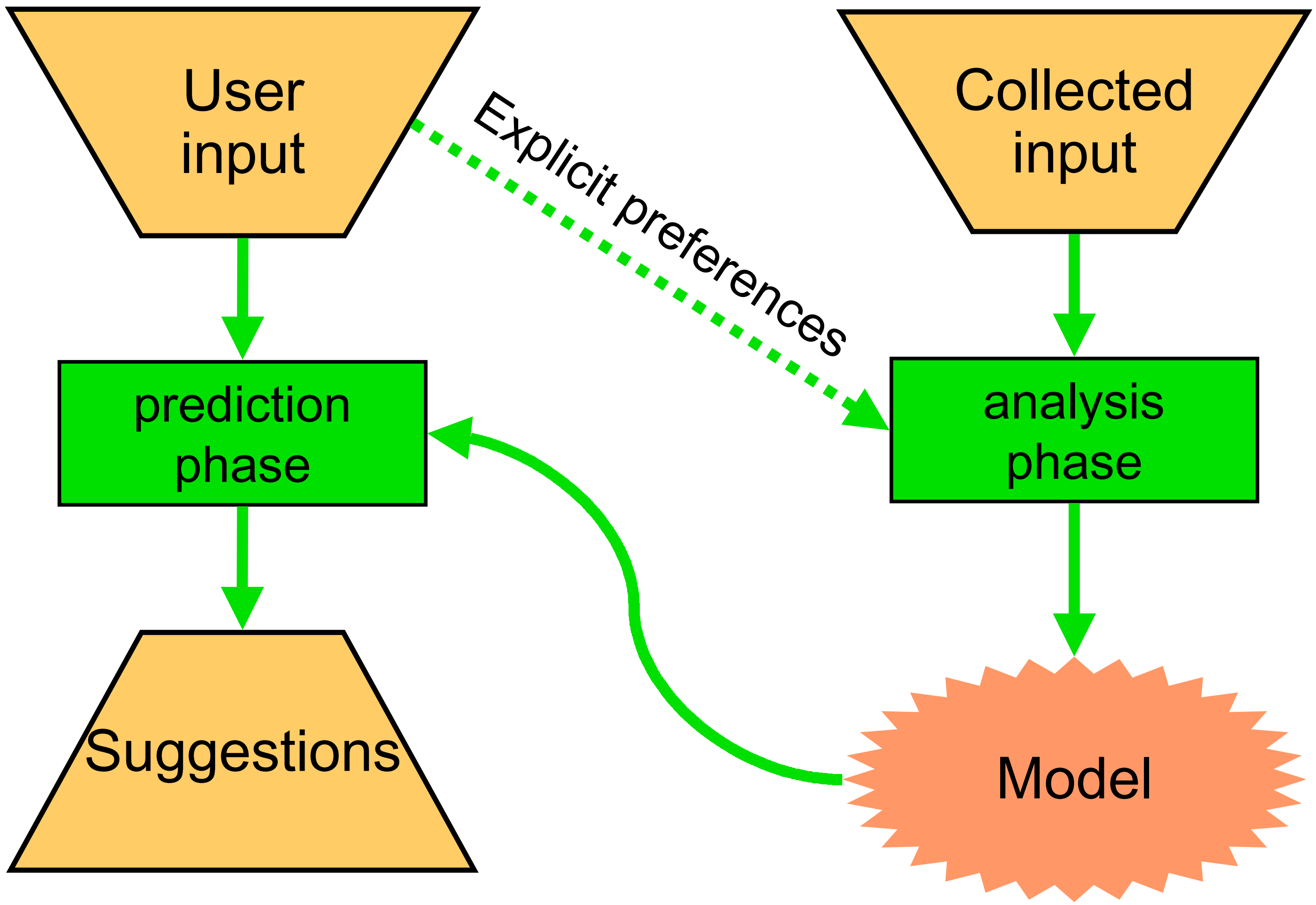
Hierarchical directories

Specialized search engines:

Search-by-example

- Collaborative filtering
- Meta-information

Collaborative filtering



Lots of projects

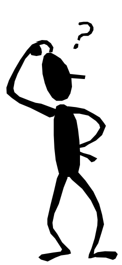
- Collaborative filtering seldom used in classic IR, big revival on the Web. Projects:
  - PHOAKS -- ATT labs → Web pages recommendation based on Usenet postings
  - GAB -- Bellcore → Web browsing
  - Grouplens -- U. Minnesota → Usenet newsgroups
  - EachToEach -- Compaq SRC → rating movies
  - …

See **`http://sims.berkeley.edu/resources/collab/`**

Google

Why do we care?

- The ranking schemes that we discussed are also a form of collaborative ranking!
  - Connectivity = people vote with their links
  - Usage = people vote with their clicks



- These schemes are used only for a global model building. Can it be combined with per-user data?
  Ideas:
  - Consider the graph induced by the user's bookmarks.
  - Profile the user -- see **`www.globalbrain.net`**
  - Deal with privacy concerns!

Web IR Tools

General-purpose search engines:

Hierarchical directories

Specialized search engines:

Search-by-example

Collaborative filtering

- Meta-information
  - Comparison of search engines
  - Log statistics

Comparison of search engines

Difficulty of independent measurement;

Usefulness for Comparison

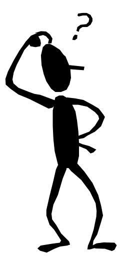
- Ideal measure: User satisfaction
- Number of user requests
- Quality of search engine index
- Size of search engine index

Comparing Search Engine Sizes

- Naïve Approaches
  - Get a list of URLs from each search engine and compare
    - Not practical or reliable.
  - Result Set Size Comparison
    - Reported sizes are approximate.
  - …
- Better Approach
  - Statistical Sampling

URL sampling

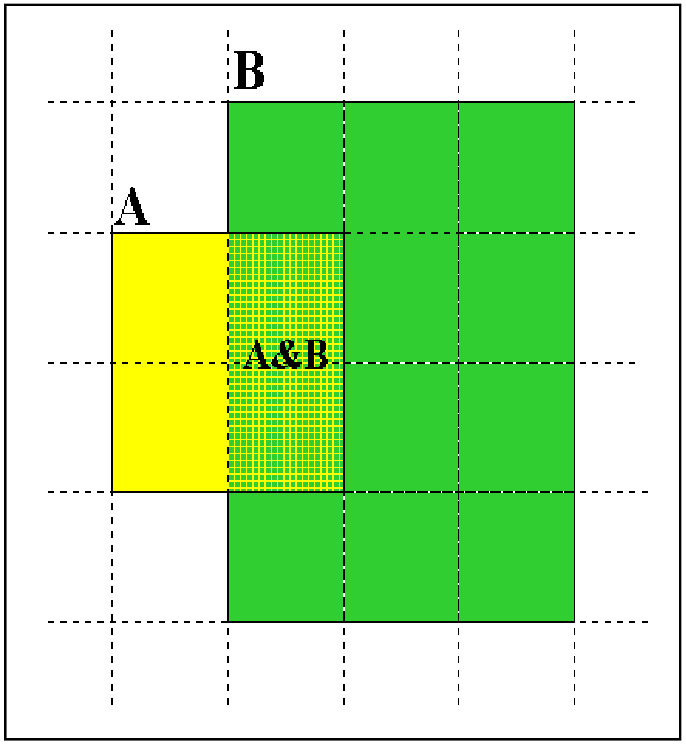
- Ideal strategy: Generate a random URL and check for containment in each index.
- Random URLs are hard to generate:
  - Random walks methods
    - Graph is directed
    - Stationary distribution is non-uniform
    - Must prove rapid mixing.
  - Pages in cache, query logs [LG'98a], etc.
    - Correlated to the interests of a particular group of users.
- A simple way: collect all pages on the Web and pick one at random.



Sampling via queries [BB'98]

- Search engines have the best crawlers -- why not exploit them?
- Method:
  - Sample from each engine in turn
  - Estimate the relative sizes of two search engines
  - Compute absolute sizes from a reference point

Estimate relative sizes



Select pages randomly from A (resp. B)

Check if page contained in B (resp. A)

$|A \cap B| \approx (1/2) * |A|$

$|A \cap B| \approx (1/6) * |B|$

$\therefore \quad |B| \approx 3 * |A|$

Two steps: (i) Selecting (ii) Checking

Selecting a random page

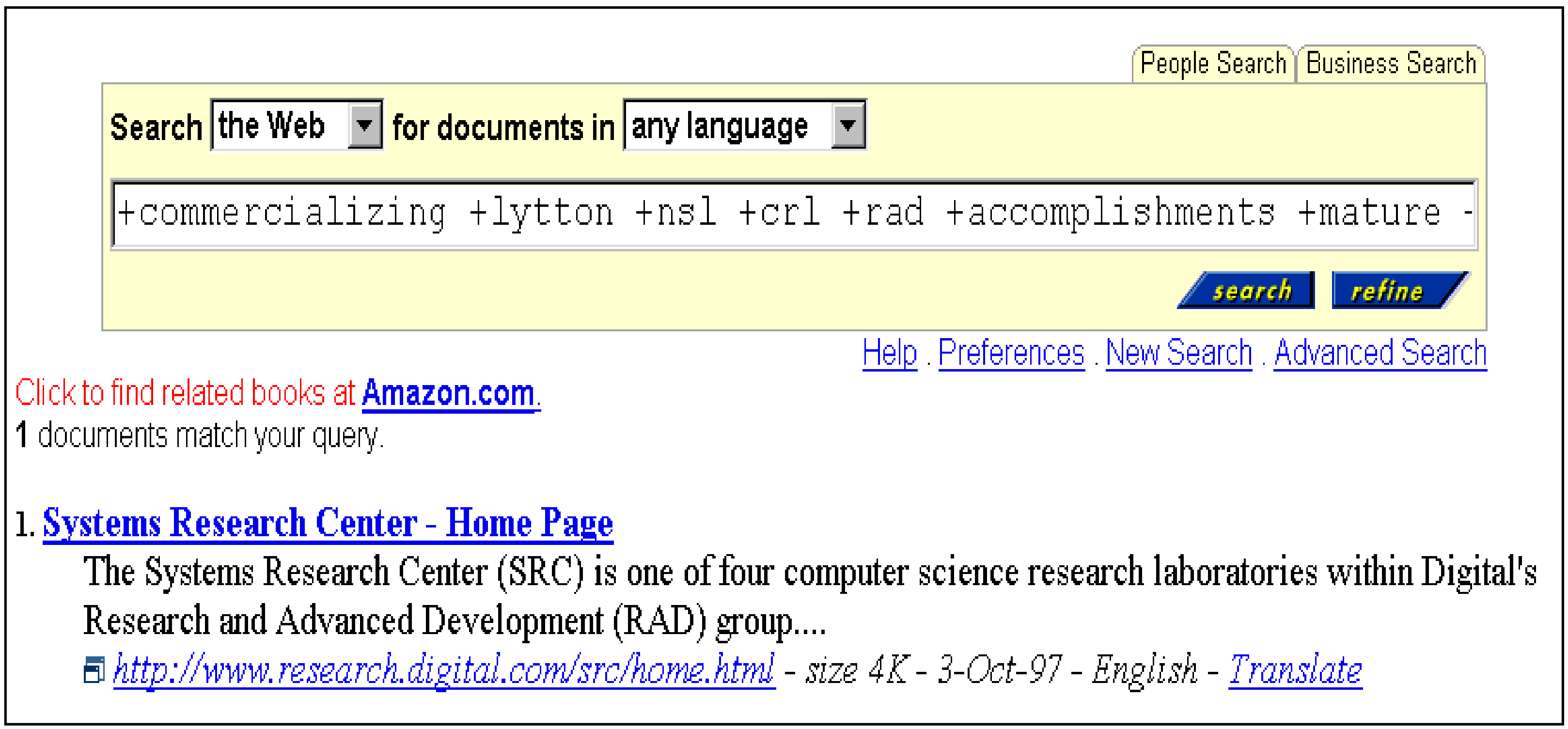
- Generate random query
  - Build a lexicon of words that occur on the Web
  - Combine random words from lexicon to form queries
- Get the first 100 query results from engine A
- Select a random page out of the set
- Distribution is biased -- the conjecture is that

$$\frac{\sum_{D \in A \cap B} p(D)}{\sum_{D \in A} p(D)} \sim \frac{|A \cap B|}{|A|}$$

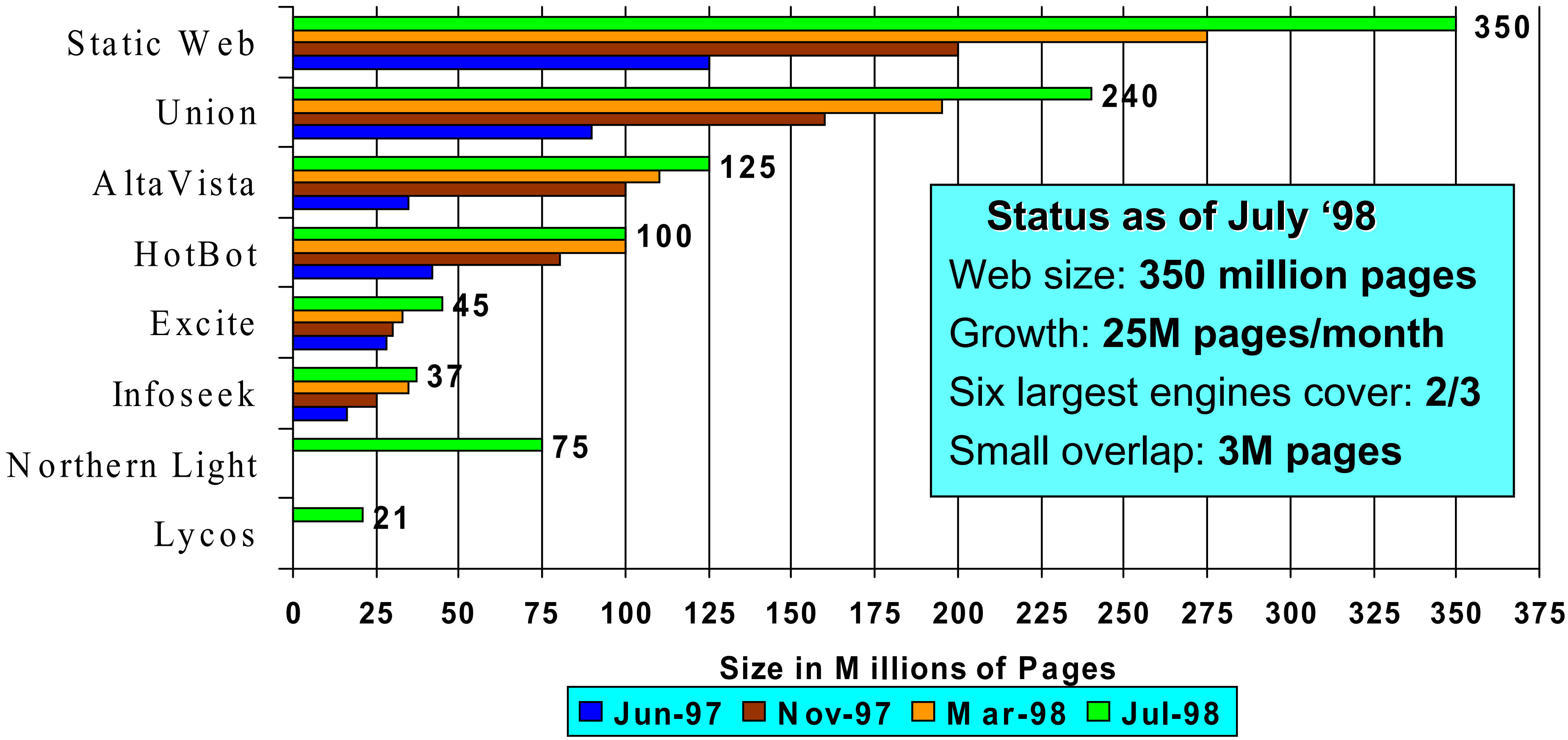
where $p(D)$ is the probability that D is picked by this scheme

Checking if an engine has a page

- Create a "unique query" for the page:
  - Use 8 rare words.
  - E.g., for the Digital Systems Research Center Home Page:



Results of the BB'98 study



Status as of July '98

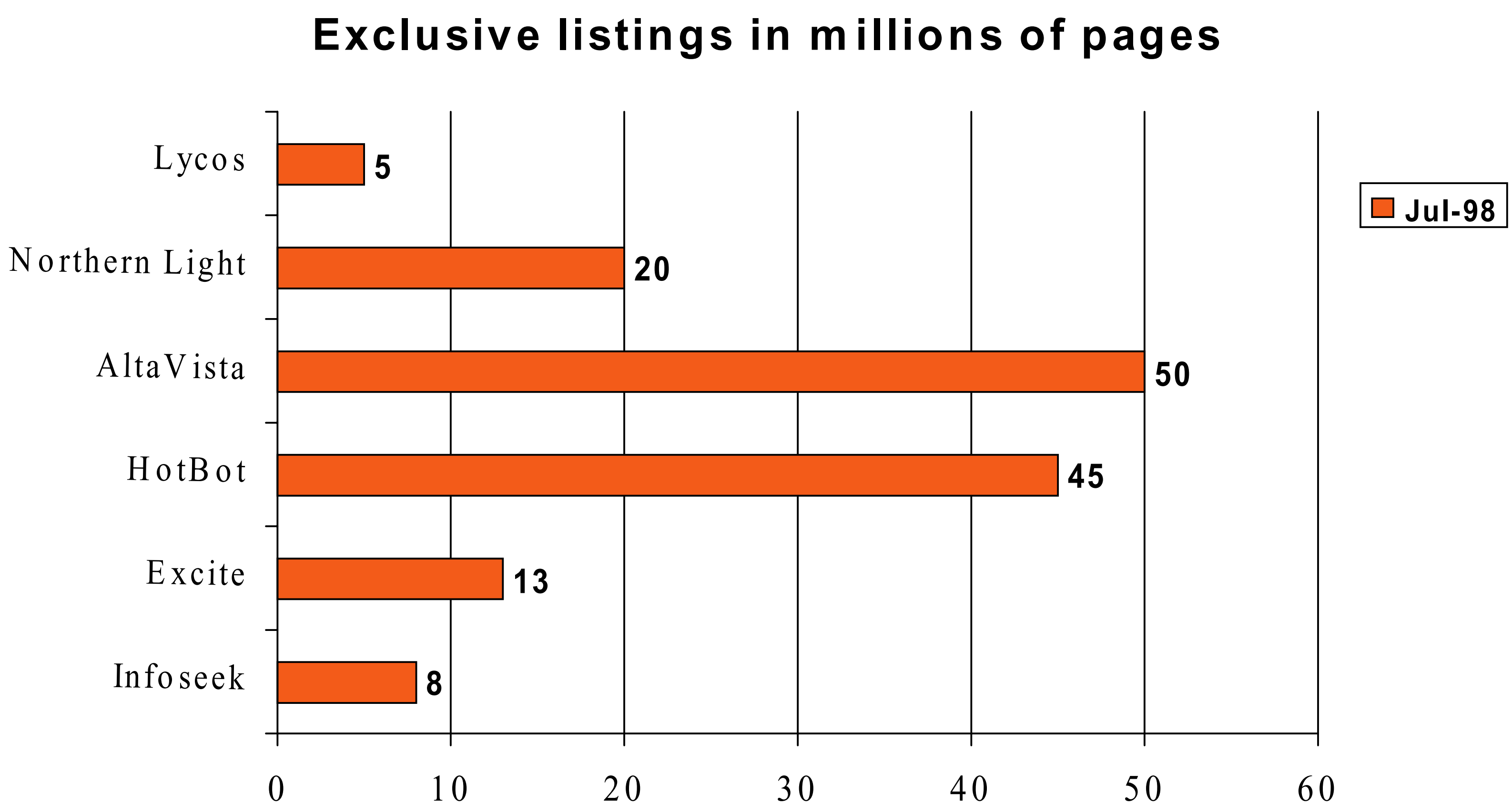
Web size: **350 million pages**

Growth: **25M pages/month**

Six largest engines cover: **2/3**

Small overlap: **3M pages**

Crawling strategies are different!



Comparison of search engines

Difficulty of independent measurement;

Usefulness for Comparison

- Ideal measure: User satisfaction
- Number of user requests
- Quality of search engine index
- Size of search engine index

Quality: A general definition [HHMN'99]

- Assign each page p a weight $w(p)$ such that

$$\sum_{all\ p} w(p) = 1$$

- Can be thought of as probability distribution on pages
- Quality of a search engine index S is $w(S) = \sum_{p \in S} w(p)$
- Example:
  - If w is same for all pages, weight is proportional to total size (in pages).
- Average page quality in index S is $w(S)/|S|$.
- We use: weight $w(p)$ of a page p is its PageRank

Estimating quality by sampling

Suppose we can choose random pages according to w (so that page p appears with probability $w(p)$)

- Choose a sample of pages $p_1, p_2, p_3 \ldots p_n$
- Check if the pages are in search engine index S
- Estimate for quality of index S is the percentage of sampled pages that are in S, i.e.

$$\overline{w}(S) = \frac{1}{n} \sum_j I[p_j \in S]$$

where $I[p_j \text{ in } S]$ = 1 if p_j is in S and 0 otherwise

Missing pieces

- Sample pages according to the PageRank distribution.
- Test whether page *p* is in search engine index *S.*
  - → same methodology as [BB'98]

Sampling pages (almost) according to PageRank

- Perform a random walk and select *n* random pages from it.
- Problems:
  - Starting state bias: finite walk only approximates PageRank.
  - Can't jump to a random page; instead, jump to a random page on a random host seen so far.
- Sampling pages according to a distribution that behaves similarly to PageRank, but it not identical to PageRank

Experiments

- Performed two long random walks with d=1/7 starting at www.yahoo.com

	Walk 1	Walk2
length	18 hours	54 hours
attempted downloads	2,867,466	6,219,704
HTML pages successfully downloaded	1,393,265	2,940,794
unique HTML pages	509,279	1,002,745
sampled pages	1,025	1,100

Random walk effectiveness

- Pages (or hosts) that are “highly-reachable” are visited often by the random walks
- Initial bias for www.yahoo.com is reduced in longer walk
- Results are consistent over the 2 walks
- The average indegree of pages with indegree <= 1000 is high:
  - 53 in walk 1
  - 60 in walk 2

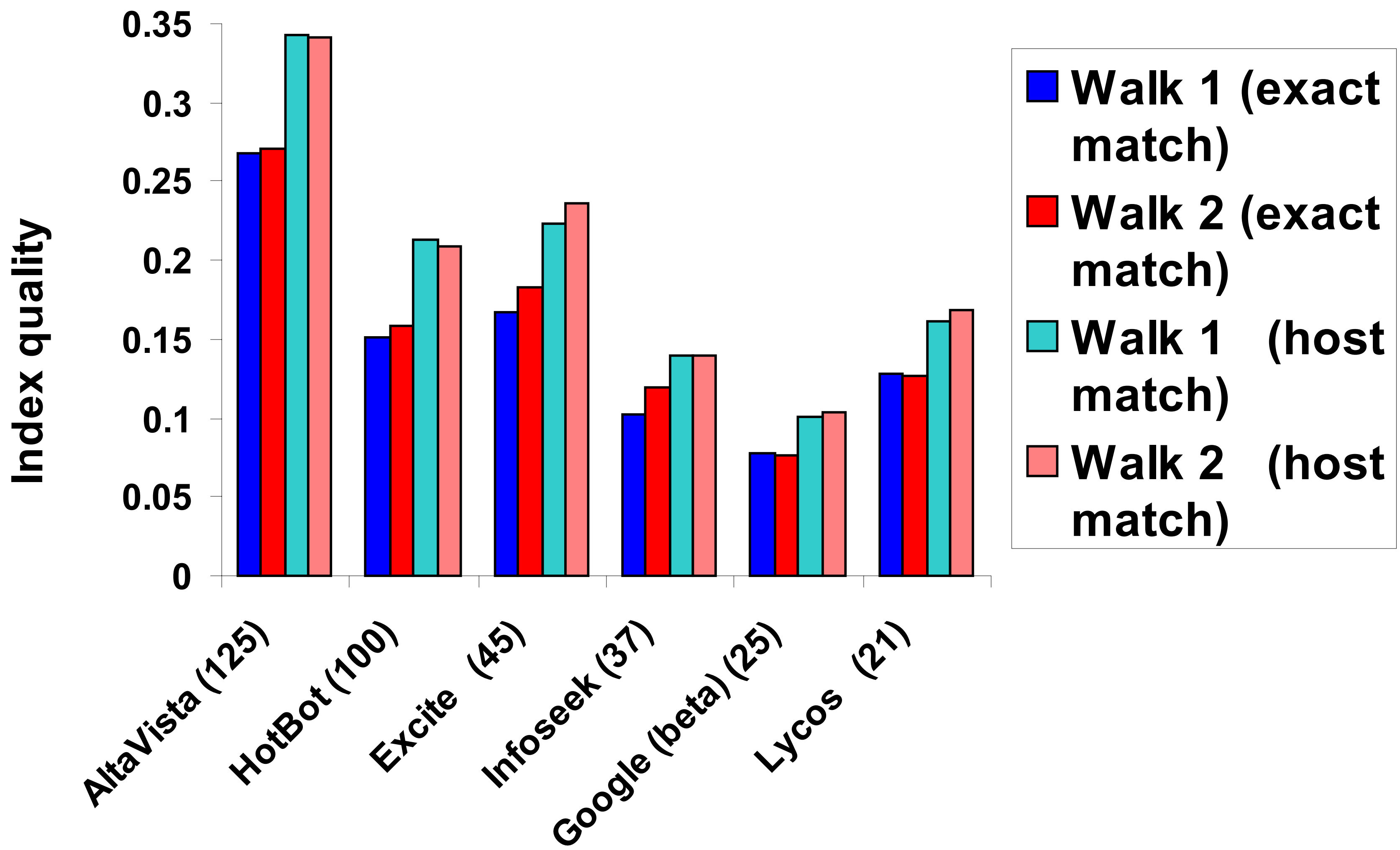
Most frequently visited pages

Page	Freq. Walk2	Freq. Walk1	Rank Walk1
www.microsoft.com/	3172	1600	1
www.microsoft.com/windows/ie/default.htm	2064	1045	3
www.netscape.com/	1991	876	6
www.microsoft.com/ie/	1982	1017	4
www.microsoft.com/windows/ie/download/	1915	943	5
www.microsoft.com/windows/ie/download/all.htm	1696	830	7
www.adobe.com/prodindex/acrobat/readstep.htm	1634	780	8
home.netscape.com/	1581	695	10
www.linkexchange.com/	1574	763	9
www.yahoo.com/	1527	1132	2

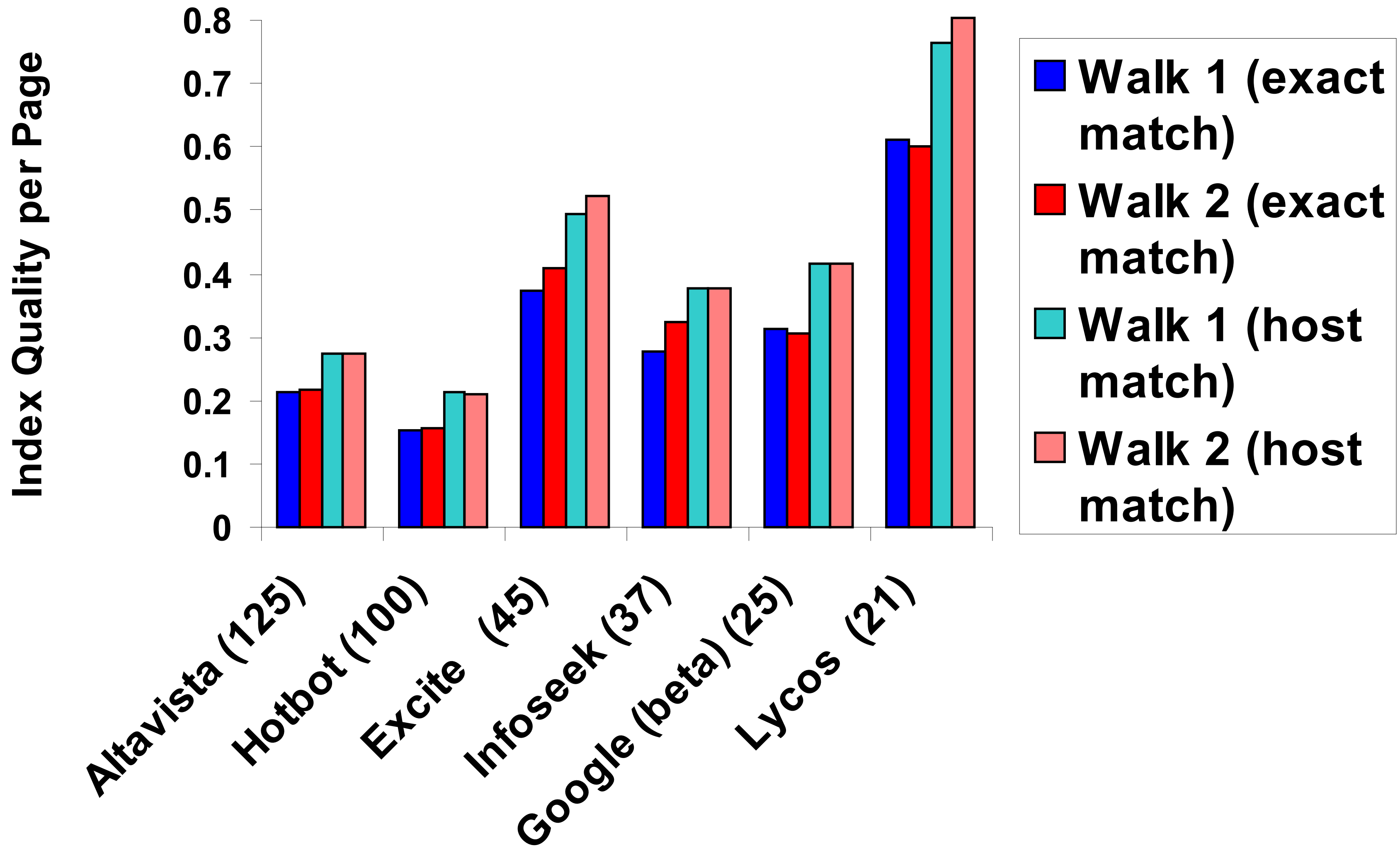
Most frequently visited hosts

Site	Frequency	Frequency	Rank
	Walk 2	Walk 1	Walk 1
www.microsoft.com	32452	16917	1
home.netscape.com	23329	11084	2
www.adobe.com	10884	5539	3
www.amazon.com	10146	5182	4
www.netscape.com	4862	2307	10
excite.netscape.com	4714	2372	9
www.real.com	4494	2777	5
www.lycos.com	4448	2645	6
www.zdnet.com	4038	2562	8
www.linkexchange.com	3738	1940	12
www.yahoo.com	3461	2595	7

Results for index quality



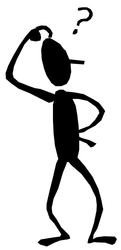
Results for index quality/page



Insights from the data

- Our approach appears consistent over repeated tests
  - Random walks are a useful tool
- Quality is different from size for search engine indices
  - Some search engines are apparently trying to index high quality pages

Open problems



- Random page generation via random walks
- Cryptography based approach: want random pages from each engine but no cheating! (page should be chosen u.a.r. from the actual index)
  - Each search engine can commit to the set of pages it has without revealing it
  - Need to ensure that this set is the same as the set actually indexed
  - Need efficient oblivious protocol to obtain random page from search engine
  - See [NP‘98] for possible solution

Web IR Tools

- General-purpose search engines:
- Hierarchical directories
- Specialized search engines:
- Search-by-example
- Collaborative filtering
- Meta-information
  - Comparison of search engines
  - Log statistics

How often do people view a page?

- Problems:
  - Web caches interfere with click counting
  - cheating pays (advertisers pay by the click)
- Solutions:
  - naïve: forces caches to re-fetch for every click.
    - Lots of traffic, annoyed Web users
  - extend HTML with counters [ML'97]
    - requires compliance, down caches falsify results.
  - use sampling [P'97]
    - force refetches on random days
    - force refetches for random users and IP addresses
  - cryptographic audit bureaus [NP'98a]
- Commercial providers: 100hot, Media Matrix, Relevant Knowledge, …

Query log statistics [SHMM'98]

request = new query or new result screen of old query

session = a series of requests by one user close together in time

- analyzed ~1B AltaVista requests consisting of:
  - ~840 M non-empty requests
  - ~575 M non-empty queries
    - ➪1.5 requests per query in the average
  - ~153 M unique non-empty queries
    - ➪ query is repeated 3.8 times in the average, but 64% of queries occur only once
  - ~285 M user sessions
    - ➪2.9 requests and 2.0 queries per session in the average

Lots of things we didn't even touch...

- Clustering = group similar items (documents or queries) together ↔ unsupervised learning
- Categorization = assign items to predefined categories ↔ supervised learning
- Classic IR issues that are not substantially different in the Web context:
  - Latent semantic indexing -- associate "concepts" to queries and documents and match on concepts
  - Summarization: abstract the most important parts of text content. (See [TS'98] for the Web context)
- Many others …

Final conclusions

- We talked mostly about IR methods and tools that
  - take advantage of the Web particularities
  - mitigate some of the difficulties
- Web IR offers plenty of interesting problems…

  … but not on a silver platter
- Almost every area of algorithms research is relevant
- Great need for good algorithm engineering!

Acknowledgements

- An earlier version of this talk was created in collaboration with Andrei Broder and was presented at the 39th IEEE Symposium on Foundations in Computer Science 1998.
- Thanks for useful comments to Krishna Bharat, Moses Charikar, Edith Cohen, Jeff Dean,Uri Feige, Puneet Kumar, Hannes Marais, Michael Mitzenmacher, Prabhakar Raghavan, and Lyle Ramshaw.