Information Retrieval

Lecture 10

Recap

- Last lecture
  - HITS algorithm
  - using anchor text
  - topic-specific pagerank

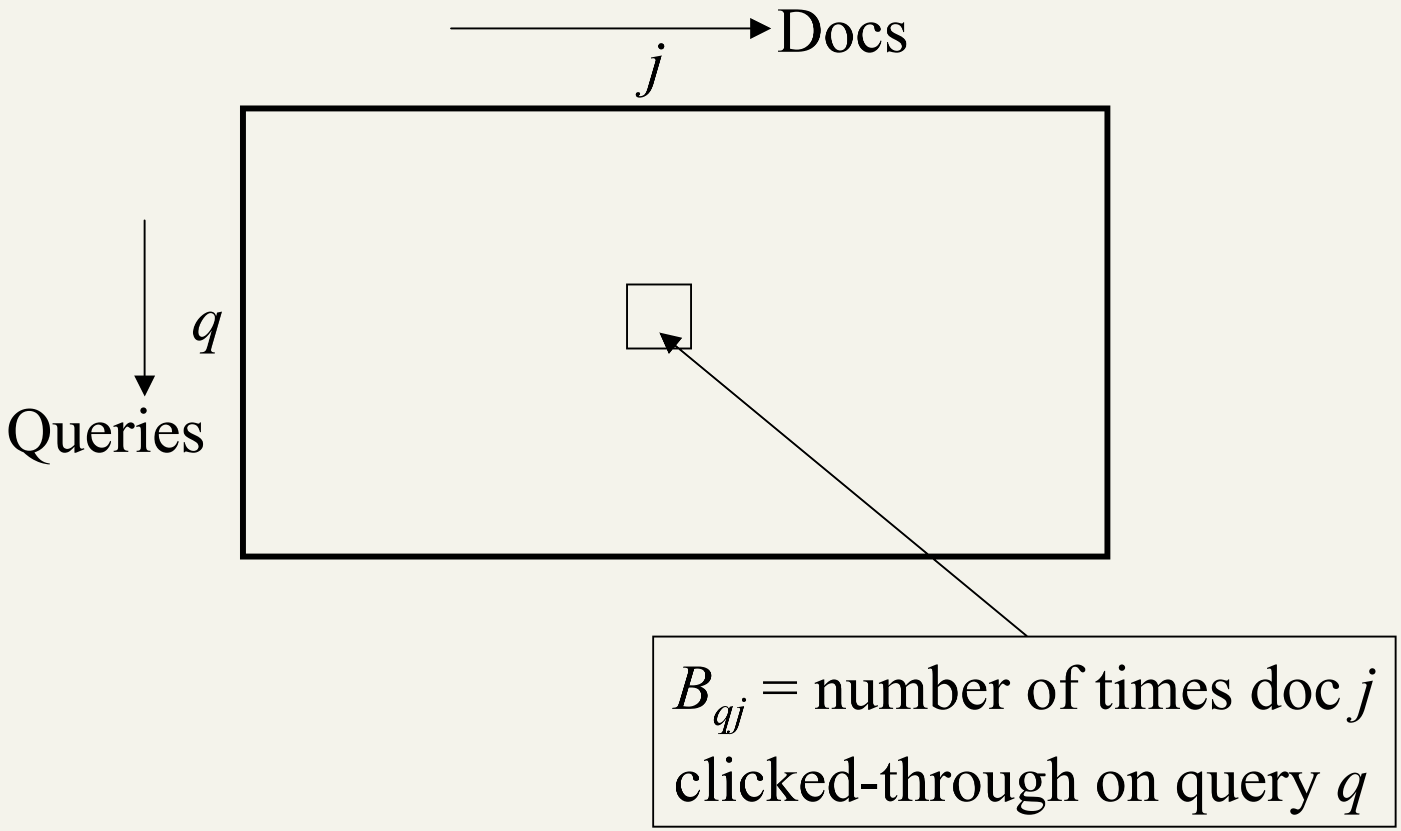
Today’s Topics

- Behavior-based ranking
- Crawling and corpus construction
- Algorithms for (near)duplicate detection
- Search engine / WebIR infrastructure

Behavior-based ranking

- For each query *Q*, keep track of which docs in the results are clicked on
- On subsequent requests for *Q*, re-order docs in results based on click-throughs
- First due to DirectHit →AskJeeves
- Relevance assessment based on
  - Behavior/usage
  - vs. content

Query-doc popularity matrix B



B_{qj} = number of times doc j clicked-through on query q

When query q issued again, order docs by B_{qj} values.

Issues to consider

- Weighing/combining text- and click-based scores.
- What identifies a query?
  - Ferrari Mondial
  - Ferrari   Mondial
  - Ferrari mondial
  - ferrari mondial
  - “Ferrari Mondial”
- Can use heuristics, but search parsing slowed.

Vector space implementation

- Maintain a term-doc popularity matrix C
  - as opposed to query-doc popularity
  - initialized to all zeros
- Each column represents a doc *j*
  - If doc *j* clicked on for query q, update $C_j \leftarrow C_j + \varepsilon q$ (here q is viewed as a vector).
- On a query q', compute its cosine proximity to C_j for all *j*.
- Combine this with the regular text score.

Issues

- Normalization of C_j after updating
- Assumption of query compositionality
  - “white house” document popularity derived from “white” and “house”
- Updating - live or batch?

Basic Assumption

- Relevance can be directly measured by number of click throughs
- Valid?

Validity of Basic Assumption

- Click through to docs that turn out to be non-relevant: what does a click mean?
- Self-perpetuating ranking
- Spam
- All votes count the same

Variants

- Time spent viewing page
  - Difficult session management
  - Inconclusive modeling so far
- Does user back out of page?
- Does user stop searching?
- Does user transact?

Crawling and Corpus Construction

- Crawl order
- Filtering duplicates
- Mirror detection

Crawling Issues

- How to crawl?
  - *Quality*: “Best” pages first
  - *Efficiency*: Avoid duplication (or near duplication)
  - *Etiquette*: Robots.txt, Server load concerns

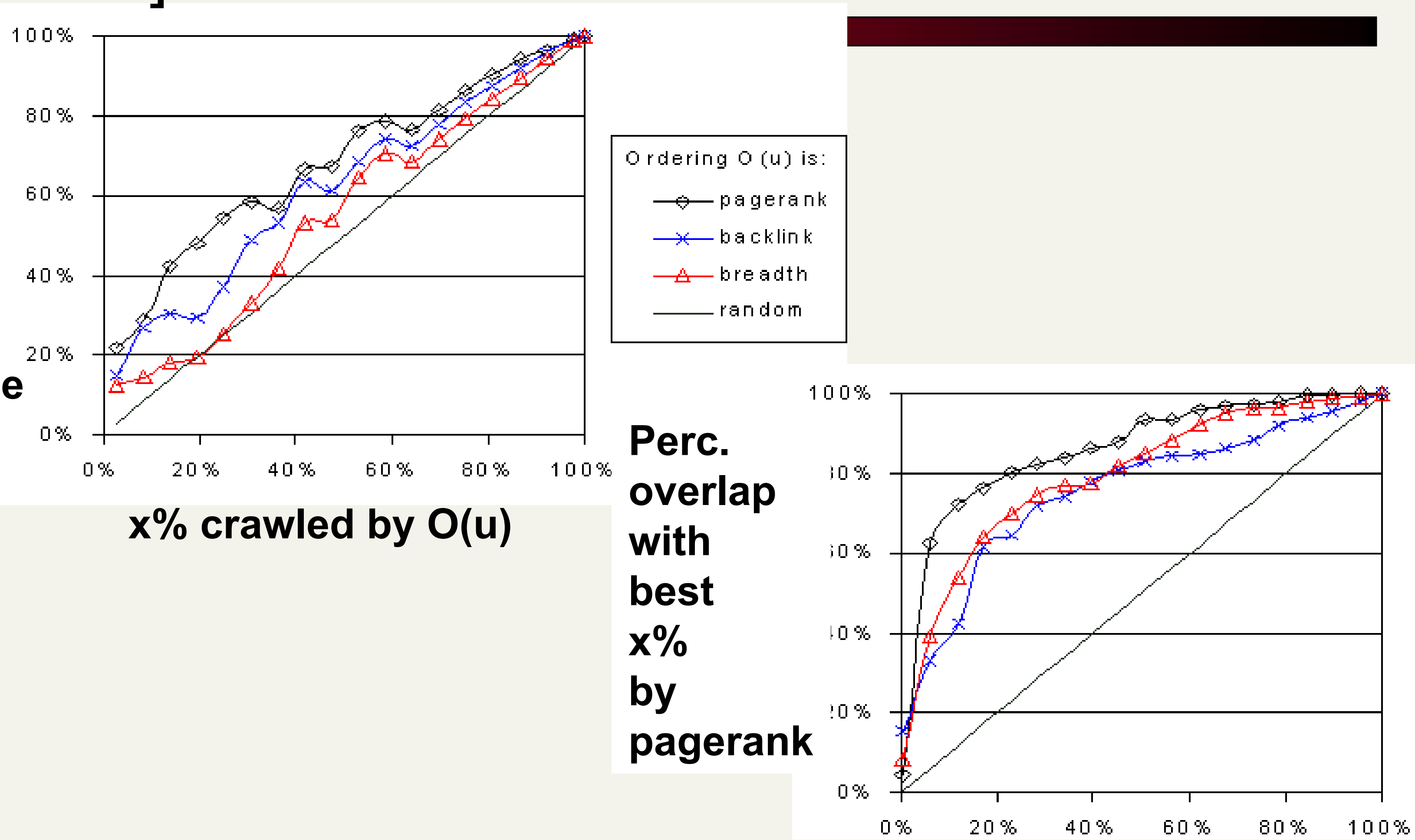
- How much to crawl? How much to index?
  - *Coverage*: How big is the Web? How much do we cover?
  - *Relative Coverage:* How much do competitors have?

- How often to crawl?
  - *Freshness*: How much has changed?
  - How much has really changed? (why is this a different question?)

Crawl Order

- Best pages first
  - Potential quality measures:
    - Final Indegree
    - Final Pagerank
  - Crawl heuristic:
    - BFS
    - Partial Indegree
    - Partial Pagerank
    - Random walk

Stanford Web Base (179K, 1998) [Cho98]



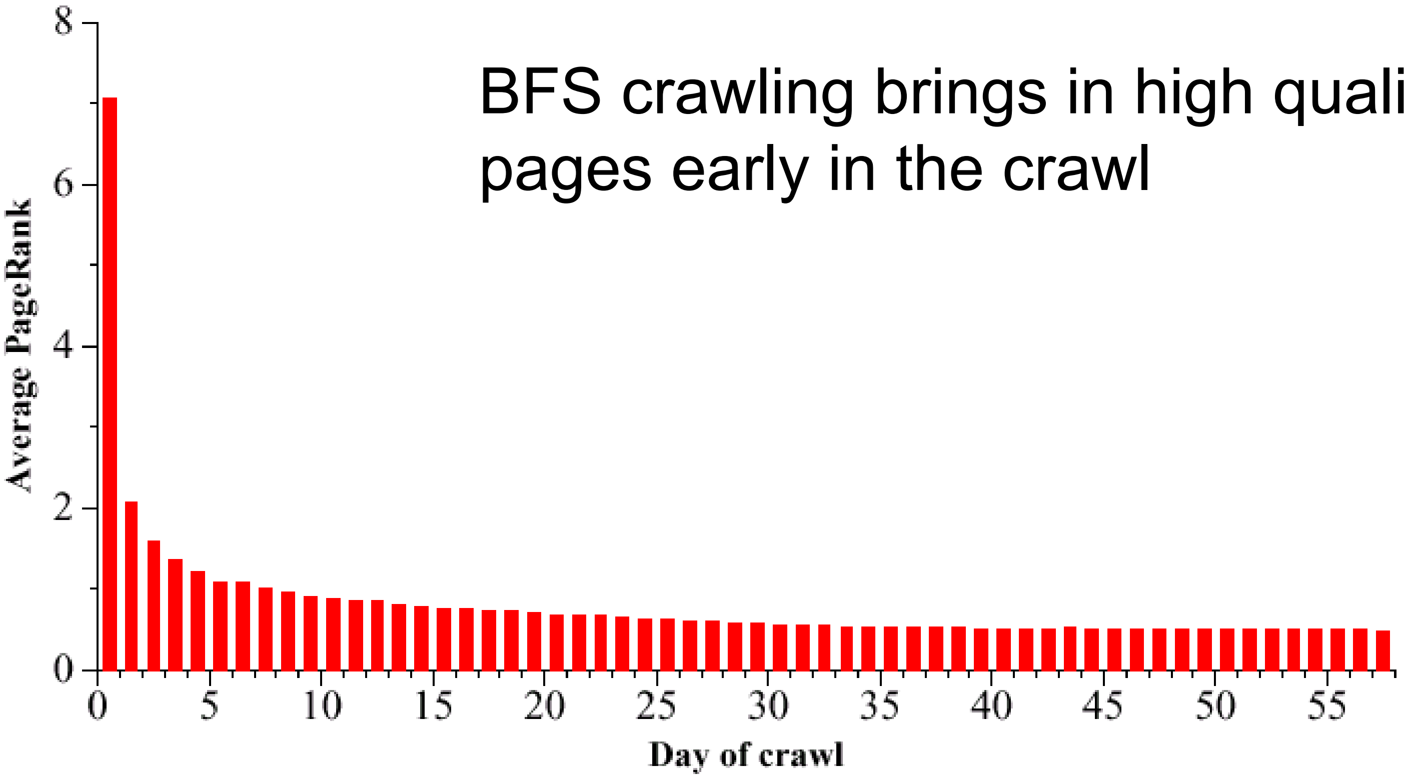
Perc. overlap with best x% by indegree

x% crawled by O(u)

Perc. overlap with best x% by pagerank

x% crawled by O(u)

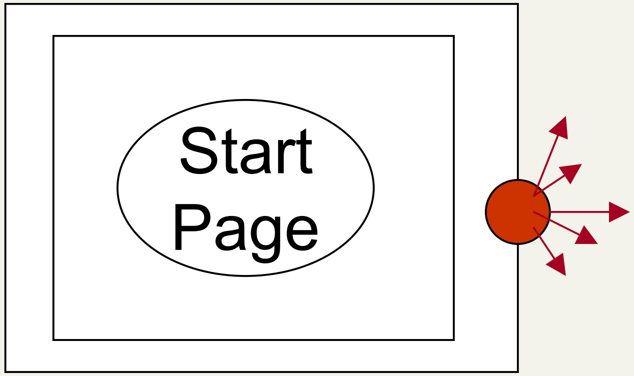
Web Wide Crawl (328M pages, 2000) [Najo01]



BFS crawling brings in high quality pages early in the crawl

Average PageRank score by day of crawl

BFS & Spam (Worst case scenario)

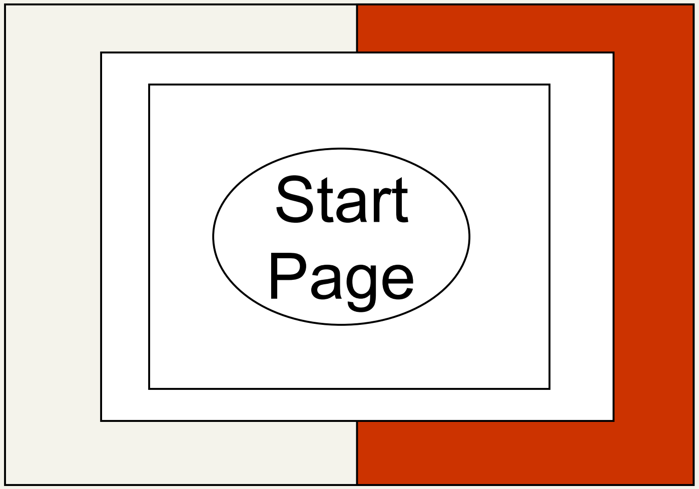


BFS depth = 2

Normal avg outdegree = 10

100 URLs on the queue
including a spam page.

Assume the spammer is able
to generate dynamic pages
with 1000 outlinks



BFS depth = 3
2000 URLs on the queue
50% belong to the spammer

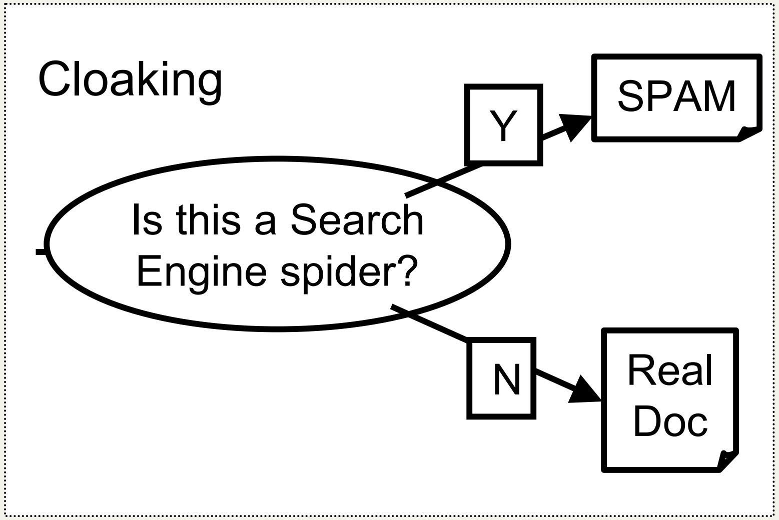
BFS depth = 4
1.01 million URLs on the
queue
99% belong to the spammer

Adversarial IR (Spam)

- Motives
    - Commercial, political, religious, lobbies
    - Promotion funded by advertising budget
- Operators
    - Contractors (Search Engine Optimizers) for lobbies, companies
    - Web masters
    - Hosting services
- Forum
    - Web master world (www.webmasterworld.com)
        - Search engine specific tricks
        - Discussions about academic papers ☺

A few spam technologies

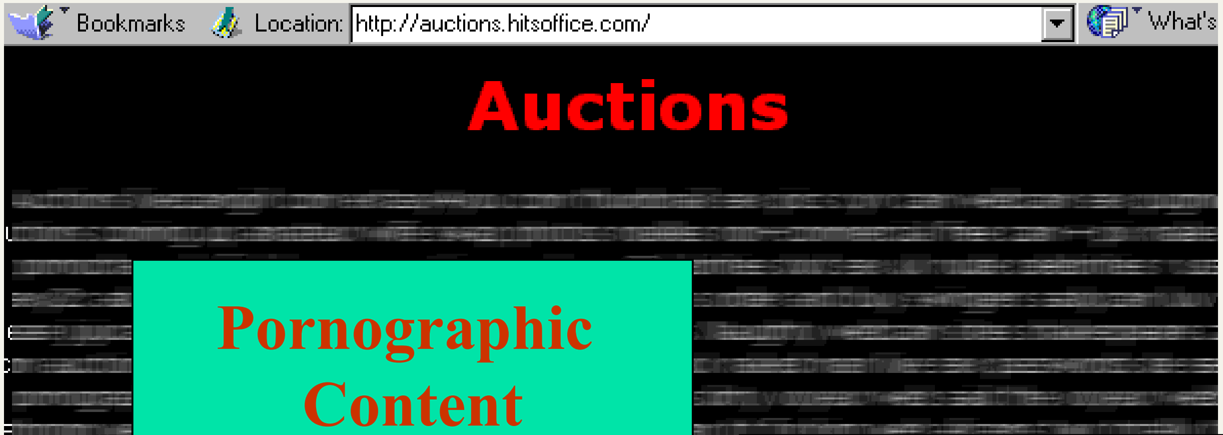
- Cloaking
  - Serve fake content to search engine robot
  - *DNS cloaking:* Switch IP address. Impersonate
- Doorway pages
  - Pages optimized for a single keyword that re-direct to the real target page
- Keyword Spam
  - Misleading meta-keywords, excessive repetition of a term, fake “anchor text”
  - Hidden text with colors, CSS tricks, etc.
- Link spamming
  - Mutual admiration societies, hidden links, awards
  - *Domain flooding:* numerous domains that point or re-direct to a target page
- Robots
  - Fake click stream
  - Fake query stream
  - Millions of submissions via Add-Url



Meta-Keywords =
“… London hotels, hotel, holiday inn, hilton, discount, booking, reservation, sex, mp3, britney spears, viagra, …”

Can you trust words on the page?

auctions.hitsoffice.com/

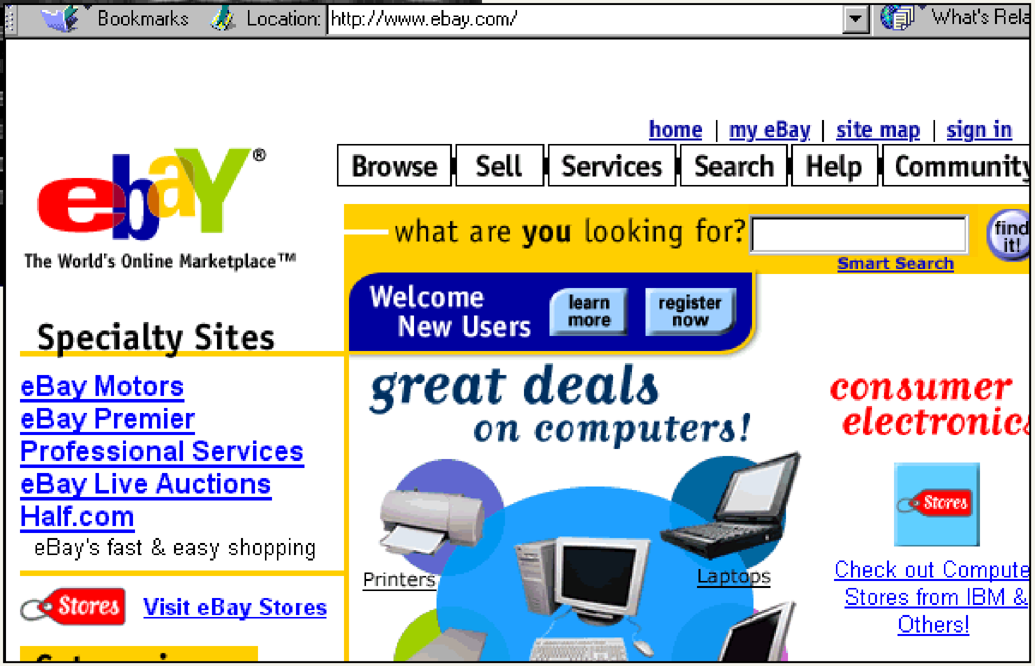


Bookmarks Location: http://auctions.hitsoffice.com/ What's

Auctions

Pornographic Content

www.ebay.com/



Bookmarks Location: http://www.ebay.com/ What's Rela

home | my eBay | site map | sign in

Browse | Sell | Services | Search | Help | Community

ebay®

The World's Online Marketplace™

what are **you** looking for? find it!

Smart Search

Welcome New Users learn more register now

Specialty Sites

- eBay Motors
- eBay Premier
- Professional Services
- eBay Live Auctions
- Half.com

eBay's fast & easy shopping

Stores Visit eBay Stores

great deals on computers!

Printers Laptops

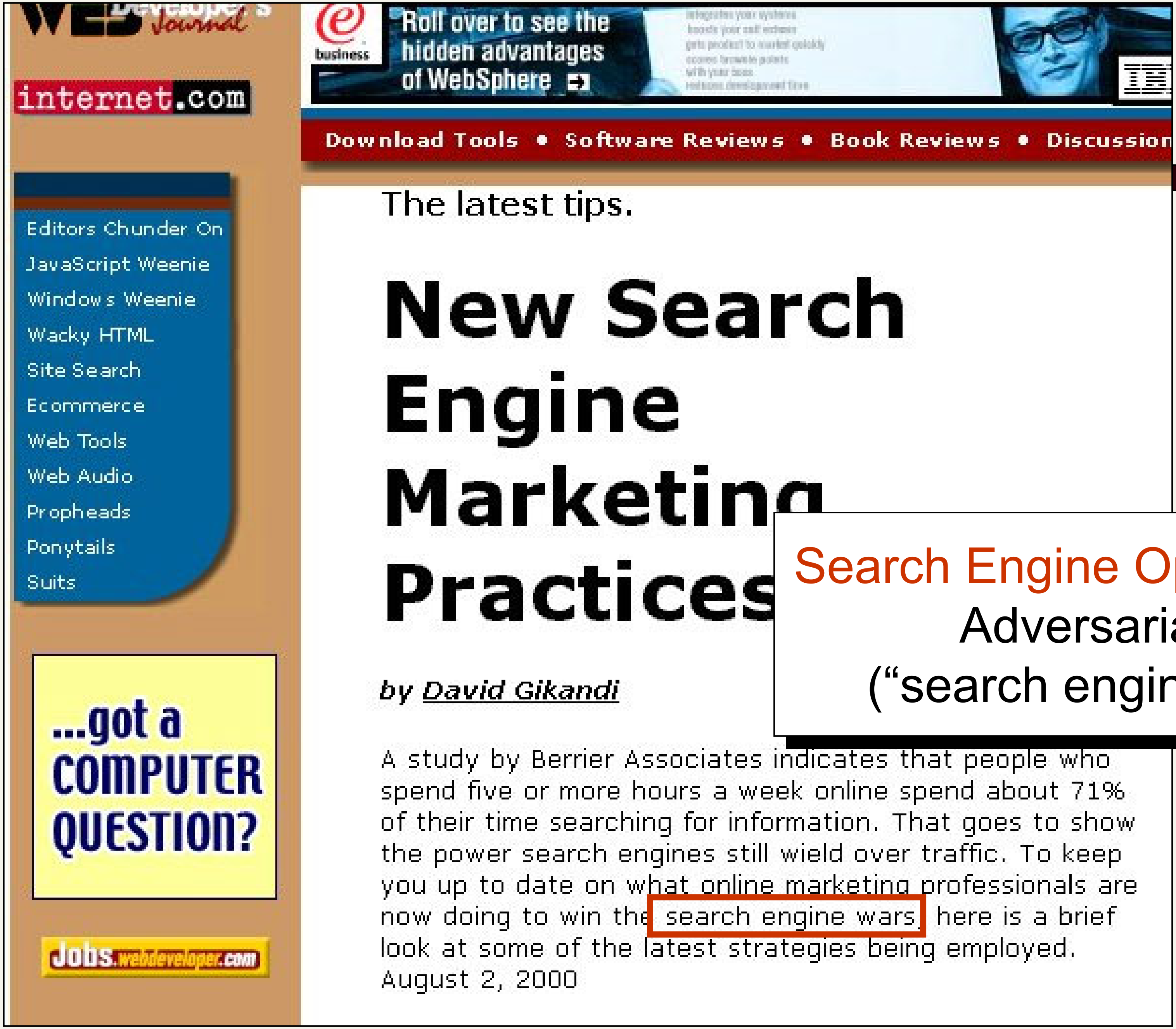
consumer electronic

Stores

Check out Compute Stores from IBM & Others!

Examples from July 2002

Search Engine Optimization I
Adversarial IR
(“search engine wars”)



The latest tips.

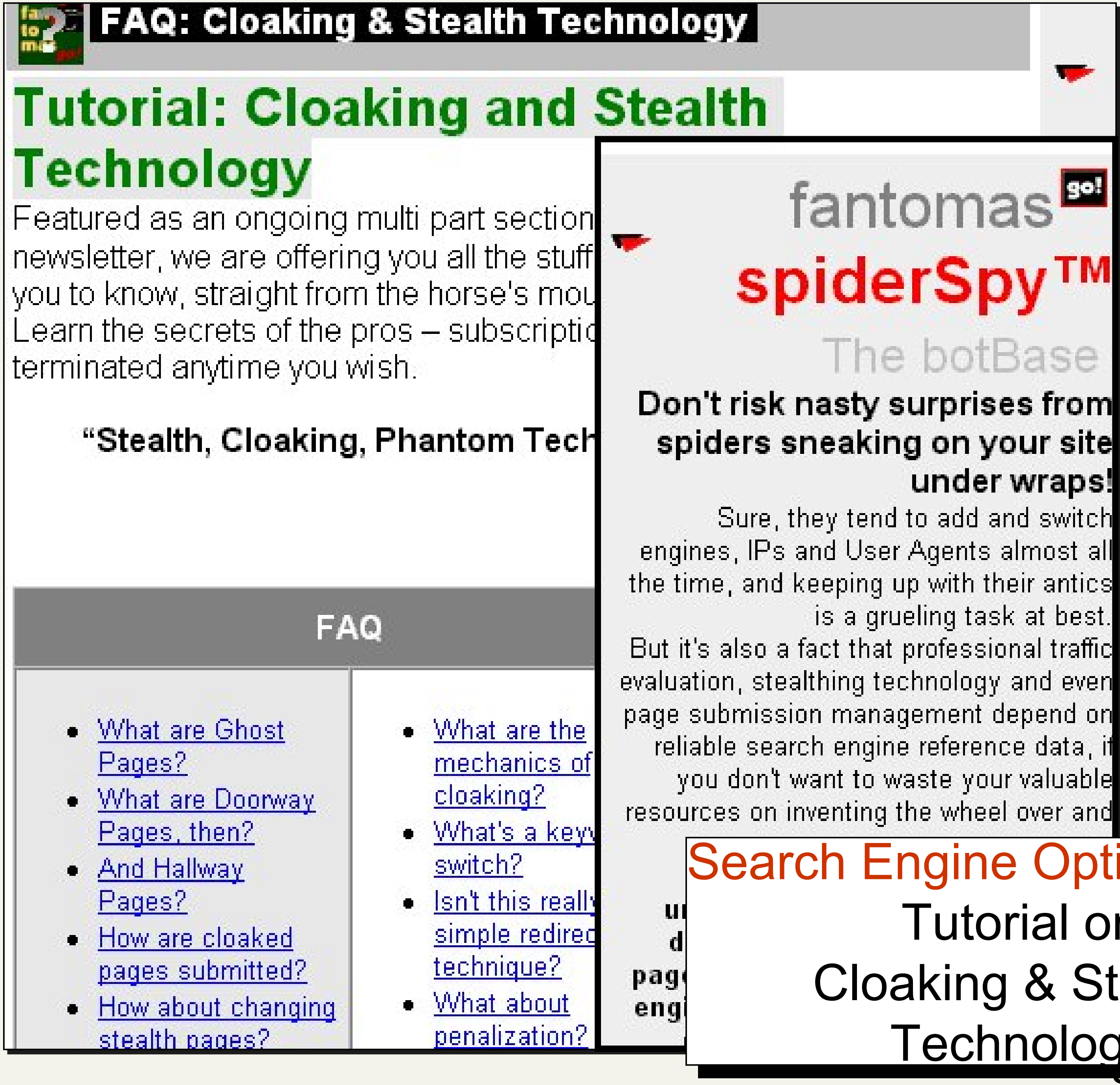
New Search Engine Marketing Practices

by David Gikandi

A study by Berrier Associates indicates that people who spend five or more hours a week online spend about 71% of their time searching for information. That goes to show the power search engines still wield over traffic. To keep you up to date on what online marketing professionals are now doing to win the search engine wars, here is a brief look at some of the latest strategies being employed.
August 2, 2000

Search Engine Optimization II

Tutorial on Cloaking & Stealth Technology



FAQ: Cloaking & Stealth Technology

Tutorial: Cloaking and Stealth Technology

Featured as an ongoing multi part section newsletter, we are offering you all the stuff you to know, straight from the horse's mou Learn the secrets of the pros – subscriptic terminated anytime you wish.

"Stealth, Cloaking, Phantom Tech

FAQ

- What are Ghost Pages?
- What are Doorway Pages, then?
- And Hallway Pages?
- How are cloaked pages submitted?
- How about changing stealth pages?

- What are the mechanics of cloaking?
- What's a keyw switch?
- Isn't this really simple redirec technique?
- What about penalization?

fantomas go!

spiderSpy™

The botBase

Don't risk nasty surprises from spiders sneaking on your site under wraps!

Sure, they tend to add and switch engines, IPs and User Agents almost all the time, and keeping up with their antics is a grueling task at best.

But it's also a fact that professional traffic evaluation, stealthing technology and even page submission management depend on reliable search engine reference data, if you don't want to waste your valuable resources on inventing the wheel over and

u d pag engi

The war against spam

- Quality signals - Prefer authoritative pages based on:
  - Votes from authors (linkage signals)
  - Votes from users (usage signals)
- Policing of URL submissions
  - Anti robot test
- Limits on meta-keywords
- Robust link analysis
  - Ignore statistically implausible linkage (or text)
  - Use link analysis to detect spammers (guilt by association)

The war against spam

- Spam recognition by machine learning
  - Training set based on known spam
- Family friendly filters
  - Linguistic analysis, general classification techniques, etc.
  - For images: flesh tone detectors, source text analysis, etc.
- Editorial intervention
  - Blacklists
  - Top queries audited
  - Complaints addressed

Duplicate/Near-Duplicate Detection

- *Duplication*: Exact match with fingerprints
- *Near-Duplication*: Approximate match
  - Overview
    - Compute syntactic similarity with an edit-distance measure
    - Use similarity threshold to detect near-duplicates
      - E.g., Similarity > 80% => Documents are “near duplicates”
      - Not transitive though sometimes used transitively

Computing Near Similarity

- Features:
  - Segments of a document (natural or artificial breakpoints) [Brin95]
  - *Shingles* (Word N-Grams) [Brin95, Brod98]

    “a rose is a rose is a rose” =>

    a_rose_is_a

    rose_is_a_rose

    is_a_rose_is
- Similarity Measure
  - TFIDF [Shiv95]
  - Set intersection [Brod98]

    (Specifically, Size_of_Intersection / Size_of_Union)

Shingles + Set Intersection

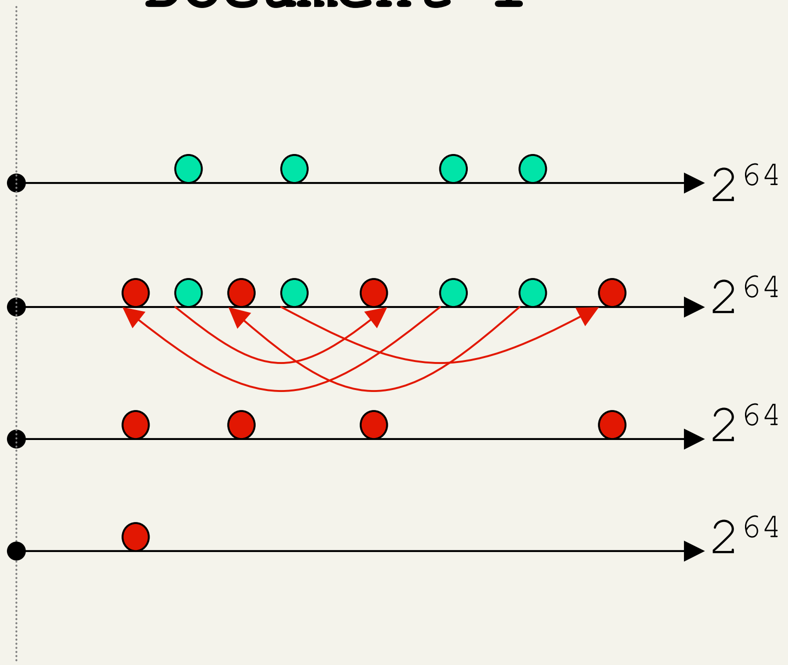
- Computing exact set intersection of shingles between all pairs of documents is expensive and infeasible
  - Approximate using a cleverly chosen subset of shingles from each (a sketch)

Shingles + Set Intersection

- Estimate size_of_intersection / size_of_union based on a short sketch ([Brod97, Brod98])
  - Create a “sketch vector” (e.g., of size 200) for each document
  - Documents which share more than t (say 80%) corresponding vector elements are similar
  - For doc D, sketch[i] is computed as follows:
    - Let f map all shingles in the universe to $0..2^m$ (e.g., f = fingerprinting)
    - Let π_i be a specific random permutation on $0..2^m$
    - Pick sketch[i] := MIN π_i (f(s)) over all shingles s in D

Computing Sketch[i] for Doc1

Document 1



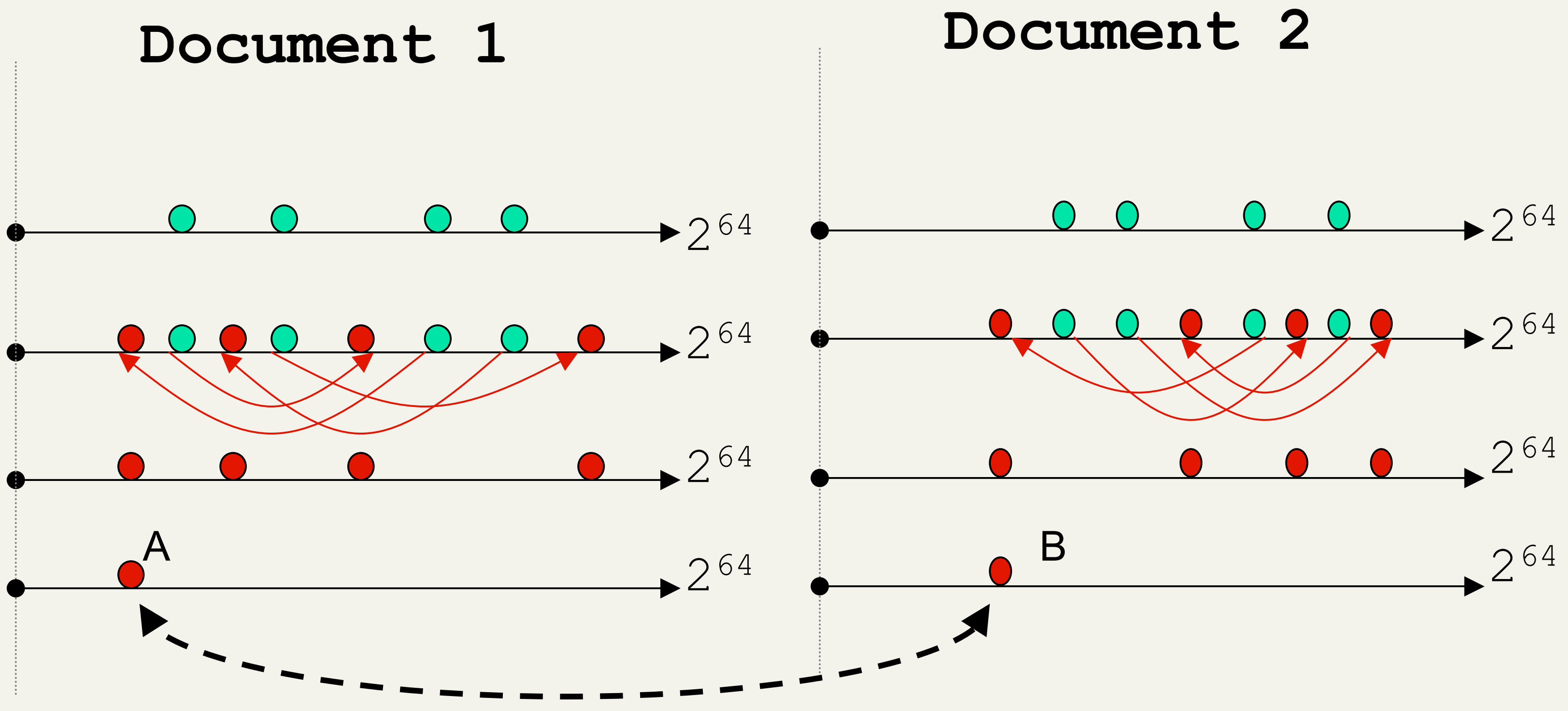
Start with 64 bit shingles

Permute on the number line

with π_i

Pick the min value

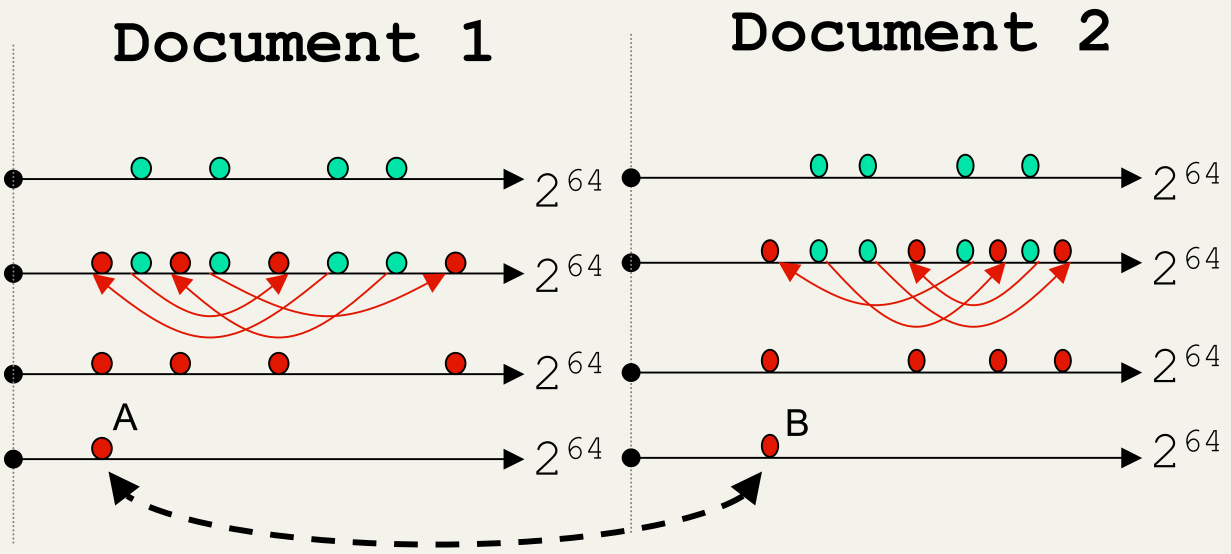
Test if Doc1.Sketch[i] = Doc2.Sketch[i]



Are these equal?

Test for 200 random permutations: $\pi_1, \pi_2, \ldots \pi_{200}$

However…



A = B iff the shingle with the MIN value in the union of Doc1 and Doc2 is common to both (I.e., lies in the intersection)

This happens with probability:

`Size_of_intersection / Size_of_union`

Question

- Document D1=D2 iff size_of_intersection=size_of_union ?

Mirror Detection

- Mirroring is systematic replication of web pages across hosts.
  - Single largest cause of duplication on the web
- Host1/α and Host2/β are <u>mirrors</u> iff

  For all (or most) paths p such that when

  http://Host1/ α / p exists

  http://Host2/ β / p exists as well

  with identical (or near identical) content, and vice versa.

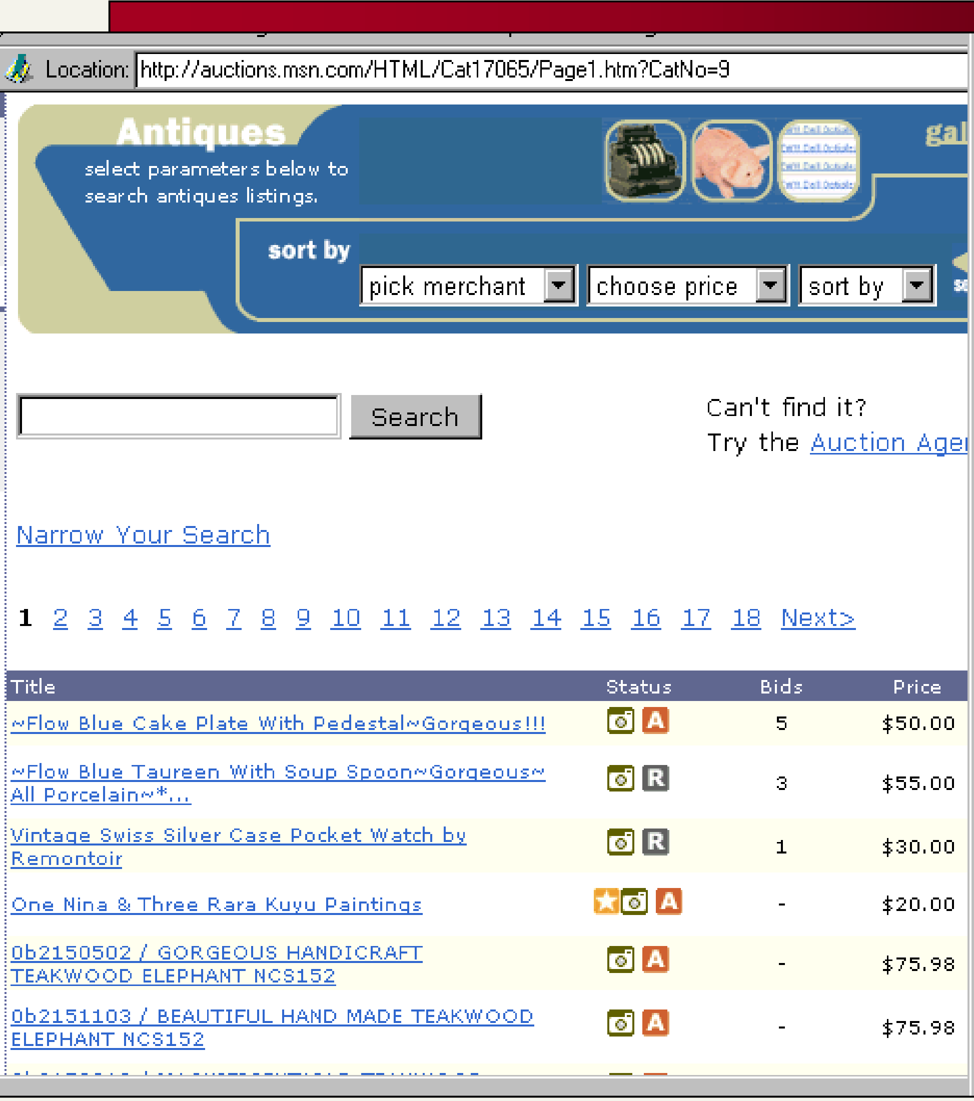
Mirror Detection example

- http://www.elsevier.com/ and http://www.elsevier.nl/
- Structural Classification of Proteins
  - http://scop.mrc-lmb.cam.ac.uk/scop
  - http://scop.berkeley.edu/
  - http://scop.wehi.edu.au/scop
  - http://pdb.weizmann.ac.il/scop
  - http://scop.protres.ru/

Repackaged Mirrors

Auctions.msn.com

Auctions.lycos.com


Location: http://auctions.msn.com/HTML/Cat17065/Page1.htm?CatNo=9

Antiques

select parameters below to search antiques listings.

sort by

pick merchant ▾ | choose price ▾ | sort by ▾

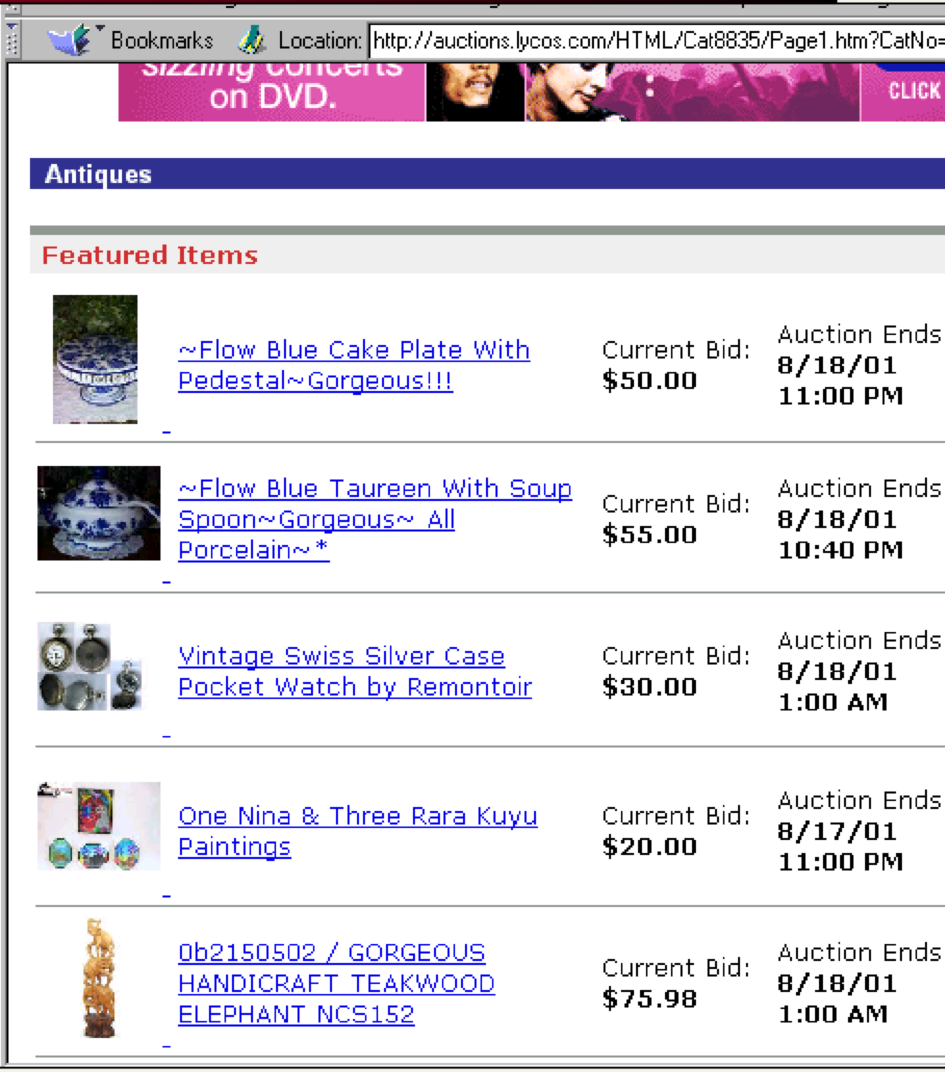
Search

Can't find it?
Try the Auction Agen

Narrow Your Search

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 Next>

Title	Status	Bids	Price
~Flow Blue Cake Plate With Pedestal~Gorgeous!!!	A	5	$50.00
~Flow Blue Taureen With Soup Spoon~Gorgeous~ All Porcelain~*...	R	3	$55.00
Vintage Swiss Silver Case Pocket Watch by Remontoir	R	1	$30.00
One Nina & Three Rara Kuyu Paintings	★ A	-	$20.00
0b2150502 / GORGEOUS HANDICRAFT TEAKWOOD ELEPHANT NCS152	A	-	$75.98
0b2151103 / BEAUTIFUL HAND MADE TEAKWOOD ELEPHANT NCS152	A	-	$75.98


Bookmarks Location: http://auctions.lycos.com/HTML/Cat8835/Page1.htm?CatNo=

Antiques

Featured Items

~Flow Blue Cake Plate With Pedestal~Gorgeous!!!	Current Bid: $50.00	Auction Ends 8/18/01 11:00 PM
~Flow Blue Taureen With Soup Spoon~Gorgeous~ All Porcelain~*	Current Bid: $55.00	Auction Ends 8/18/01 10:40 PM
Vintage Swiss Silver Case Pocket Watch by Remontoir	Current Bid: $30.00	Auction Ends 8/18/01 1:00 AM
One Nina & Three Rara Kuyu Paintings	Current Bid: $20.00	Auction Ends 8/17/01 11:00 PM
0b2150502 / GORGEOUS HANDICRAFT TEAKWOOD ELEPHANT NCS152	Current Bid: $75.98	Auction Ends 8/18/01 1:00 AM

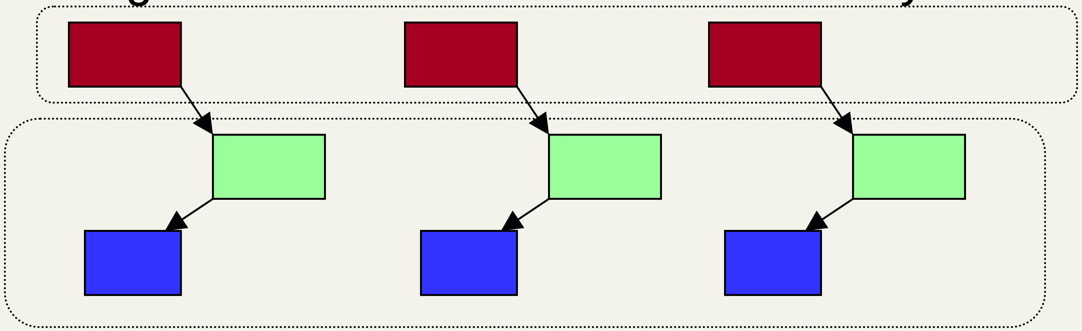
Aug

Motivation

- Why detect mirrors?
  - Smart crawling
    - Fetch from the fastest or freshest server
    - Avoid duplication
  - Better connectivity analysis
    - Combine inlinks
    - Avoid double counting outlinks
  - Redundancy in result listings
    - “If that fails you can try: <mirror>/samepath”
  - Proxy caching

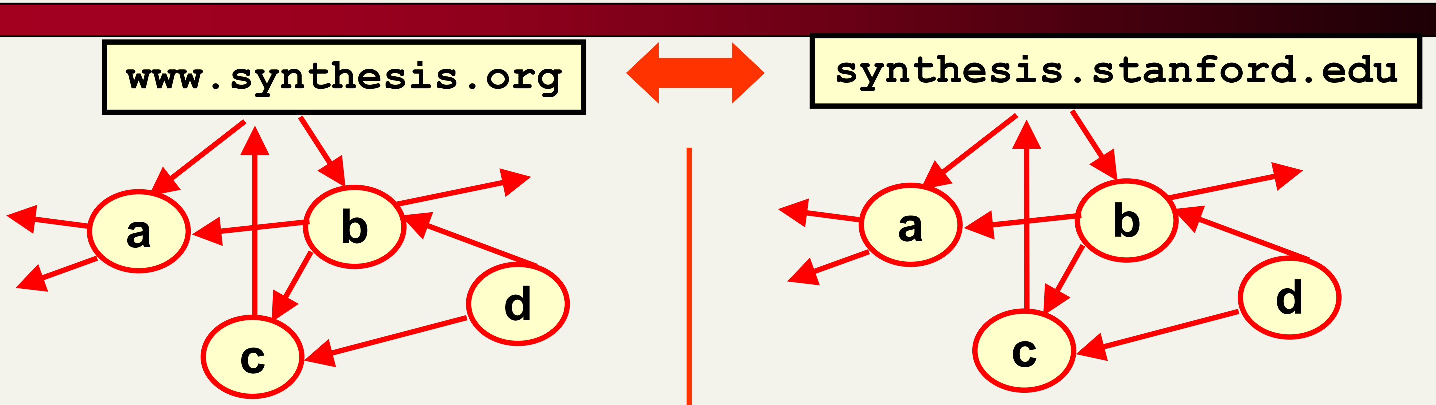
Bottom Up Mirror Detection [Cho00]

- Maintain clusters of subgraphs
- Initialize clusters of trivial subgraphs
  - Group near-duplicate single documents into a cluster
- Subsequent passes
  - Merge clusters of the same cardinality and corresponding linkage



  - Avoid decreasing cluster cardinality
- To detect mirrors we need:
  - Adequate path overlap
  - Contents of corresponding pages within a small time range

Can we use URLs to find mirrors?


www.synthesis.org

synthesis.stanford.edu

www.synthesis.org/Docs/ProjAbs/synsys/synalysis.html
www.synthesis.org/Docs/ProjAbs/synsys/visual-semi-quan
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-mechatro
www.synthesis.org/Docs/myr5/mech/mech-take-home.htm
www.synthesis.org/Docs/myr5/synsys/experiential-learning
www.synthesis.org/Docs/myr5/synsys/mm-mech-dissec.ht
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.h

synthesis.stanford.edu/Docs/ProjAbs/deliv/high-tech-…
.stanford.edu/Docs/ProjAbs/mech/mech-enhanced…
synthesis.stanford.edu/Docs/ProjAbs/mech/mech-intro-…
synthesis.stanford.edu/Docs/ProjAbs/mech/mech-mm-case-…
synthesis.stanford.edu/Docs/ProjAbs/synsys/quant-dev-new-…
synthesis.stanford.edu/Docs/annual.report96.final.html
synthesis.stanford.edu/Docs/annual.report96.final_fn.html
rd.edu/Docs/myr5/assessment
myr5/assessment/assessment-…
is.stanford.edu/Docs/myr5/assessment/mm-forum-kiosk-…
is.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
is.stanford.edu/Docs/myr5/cicee/comp-integ-analysis.html

Top Down Mirror Detection
[Bhar99, Bhar00c]

- E.g.,
  www.synthesis.org/Docs/ProjAbs/synsys/synalysis.html
  synthesis.stanford.edu/Docs/ProjAbs/synsys/quant-dev-new-teach.html
- What features could indicate mirroring?
  - Hostname similarity:
    - word unigrams and bigrams: { www, www.synthesis, synthesis, …}
  - Directory similarity:
    - Positional path bigrams { 0:Docs/ProjAbs, 1:ProjAbs/synsys, … }
  - IP address similarity:
    - 3 or 4 octet overlap
    - Many hosts sharing an IP address => virtual hosting by an ISP
  - Host outlink overlap
  - Path overlap
    - Potentially, path + sketch overlap

Implementation

- Phase I - Candidate Pair Detection
  - Find features that pairs of hosts have in common
  - Compute a list of host pairs which might be mirrors
- Phase II - Host Pair Validation
  - Test each host pair and determine extent of mirroring
    - Check if 20 paths sampled from Host1 have near-duplicates on Host2 and vice versa
    - Use transitive inferences:

      IF Mirror(A,x) AND Mirror(x,B) THEN Mirror(A,B)

      IF Mirror(A,x) AND !Mirror(x,B) THEN !Mirror(A,B)
- Evaluation
  - 140 million URLs on 230,000 hosts (1999)
  - Best approach combined 5 sets of features
    - Top 100,000 host pairs had precision = 0.57 and recall = 0.86

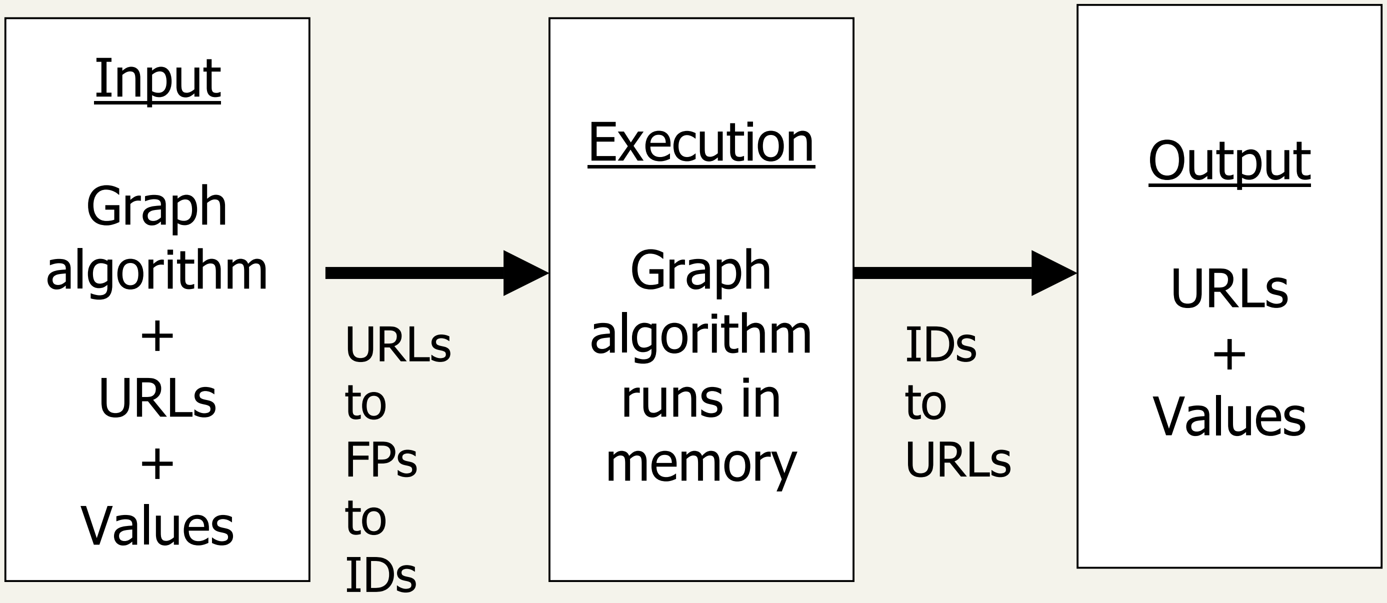
WebIR Infrastructure

- Connectivity Server
  - Fast access to links to support for link analysis
- Term Vector Database
  - Fast access to document vectors to augment link analysis

Connectivity Server
[*CS1:* Bhar98b, *CS2 & 3:* Rand01]

- Fast web graph access to support connectivity analysis
- Stores mappings in memory from
  - URL to outlinks, URL to inlinks
- Applications
  - HITS, Pagerank computations
  - Crawl simulation
  - Graph algorithms: web connectivity, diameter etc.
    - more on this later
  - Visualizations

Usage



Input

Graph algorithm + URLs + Values

URLs to FPs to IDs

Execution

Graph algorithm runs in memory

IDs to URLs

Output

URLs + Values

Translation Tables on Disk

URL text: 9 bytes/URL (compressed from ~80 bytes)

FP(64b) -> ID(32b): 5 bytes

ID(32b) -> FP(64b): 8 bytes

ID(32b) -> URLs: 0.5 bytes

ID assignment

- Partition URLs into 3 sets, sorted lexicographically
  - High: Max degree > 254
  - Medium: 254 > Max degree > 24
  - Low: remaining (75%)
- IDs assigned in sequence (densely)

Adjacency lists

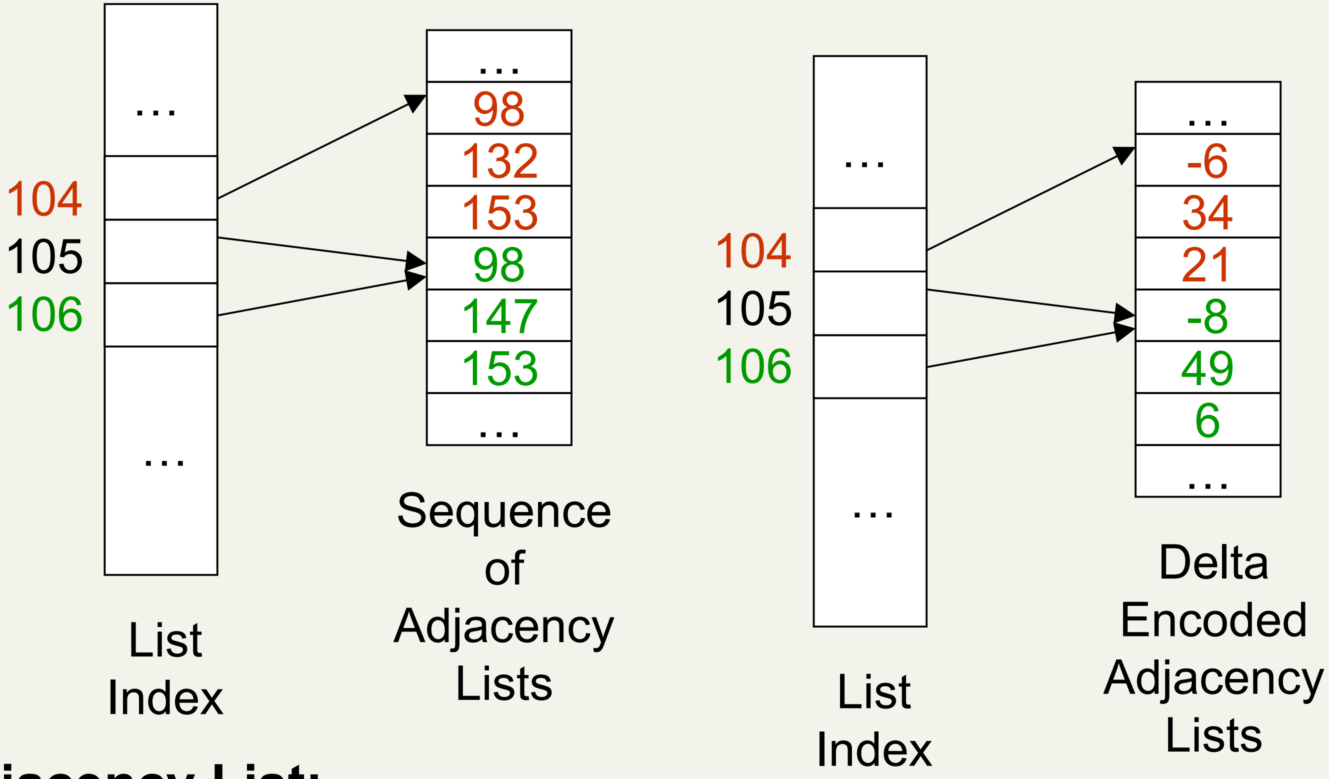
- In memory tables for Outlinks, Inlinks
- List index maps from a Source ID to start of adjacency list

E.g., HIGH IDs:

Max(indegree , outdegree) > 254

ID	URL
…	
9891	www.amazon.com/
9912	www.amazon.com/jobs/
…	
9821878	www.geocities.com/
…	
40930030	www.google.com/
…	
85903590	www.yahoo.com/

Adjacency List Compression - I



104
105
106

…
…

List
Index

…
98
132
153
98
147
153
…

Sequence
of
Adjacency
Lists

104
105
106

…
…

List
Index

…
-6
34
21
-8
49
6
…

Delta
Encoded
Adjacency
Lists

- **Adjacency List:**
  - Smaller delta values are exponentially more frequent (80% to same host)
  - Compress deltas with variable length encoding (e.g., Huffman)
- **List Index pointers:** 32b for high, Base+16b for med, Base+8b for low
  - Avg = 12b per pointer

Adjacency List Compression - II

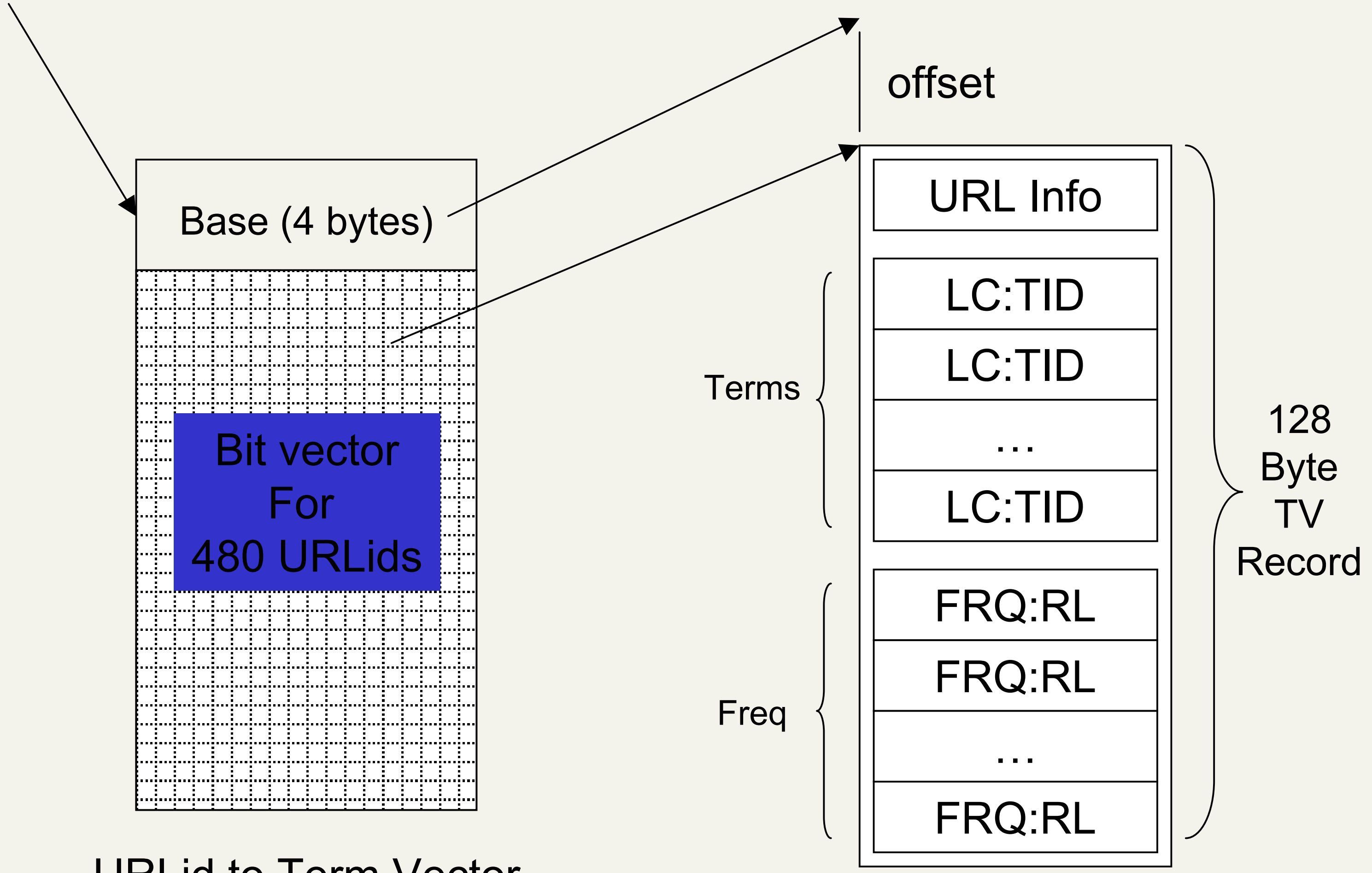
- Inter List Compression
  - Basis: Similar URLs may share links
    - Close in ID space => adjacency lists may overlap
  - Approach
    - Define a <u>representative adjacency list</u> for a block of IDs
      - Adjacency list of a reference ID
      - Union of adjacency lists in the block
    - Represent adjacency list in terms of deletions and additions *when it is cheaper to do so*
  - Measurements
    - Intra List + Starts: 8-11 bits per link (580M pages/16GB RAM)
    - Inter List: 5.4-5.7 bits per link (870M pages/16GB RAM.)

Term Vector Database [Stat00]

- Fast access to 50 word term vectors for web pages
  - Term Selection:
    - Restricted to middle 1/3rd of lexicon by document frequency
    - Top 50 words in document by TF.IDF.
  - Term Weighting:
    - Deferred till run-time (can be based on term freq, doc freq, doc length)
- Applications
  - Content + Connectivity analysis (e.g., Topic Distillation)
  - Topic specific crawls
  - Document classification
- Performance
  - Storage: 33GB for 272M term vectors
  - Speed: 17 ms/vector on AlphaServer 4100 (latency to read a disk block)

Architecture

URLid * 64 /480



URLid to Term Vector Lookup