Web: Search, Web Size Estimate, Ads and Duplicates Removal

Web Search History

- In 1993, early web robots (spiders) were built to collect URL's:
  - Wanderer
  - ALIWEB (Archie-Like Index of the WEB)
  - WWW Worm (indexed URL's and titles for regex search)
- In 1994, Stanford grad students David Filo and Jerry Yang started manually collecting popular web sites into a topical hierarchy called Yahoo.

Web Search History (cont)

- In early 1994, Brian Pinkerton developed WebCrawler as a class project at U Wash. (eventually became part of Excite and AOL).
- A few months later, Fuzzy Maudlin, a grad student at CMU developed Lycos. First to use a standard IR system as developed for the DARPA Tipster project. First to index a large set of pages.
- In late 1995, DEC developed Altavista. Used a large farm of Alpha machines to quickly process large numbers of queries. Supported boolean operators, phrases, and “reverse pointer” queries.

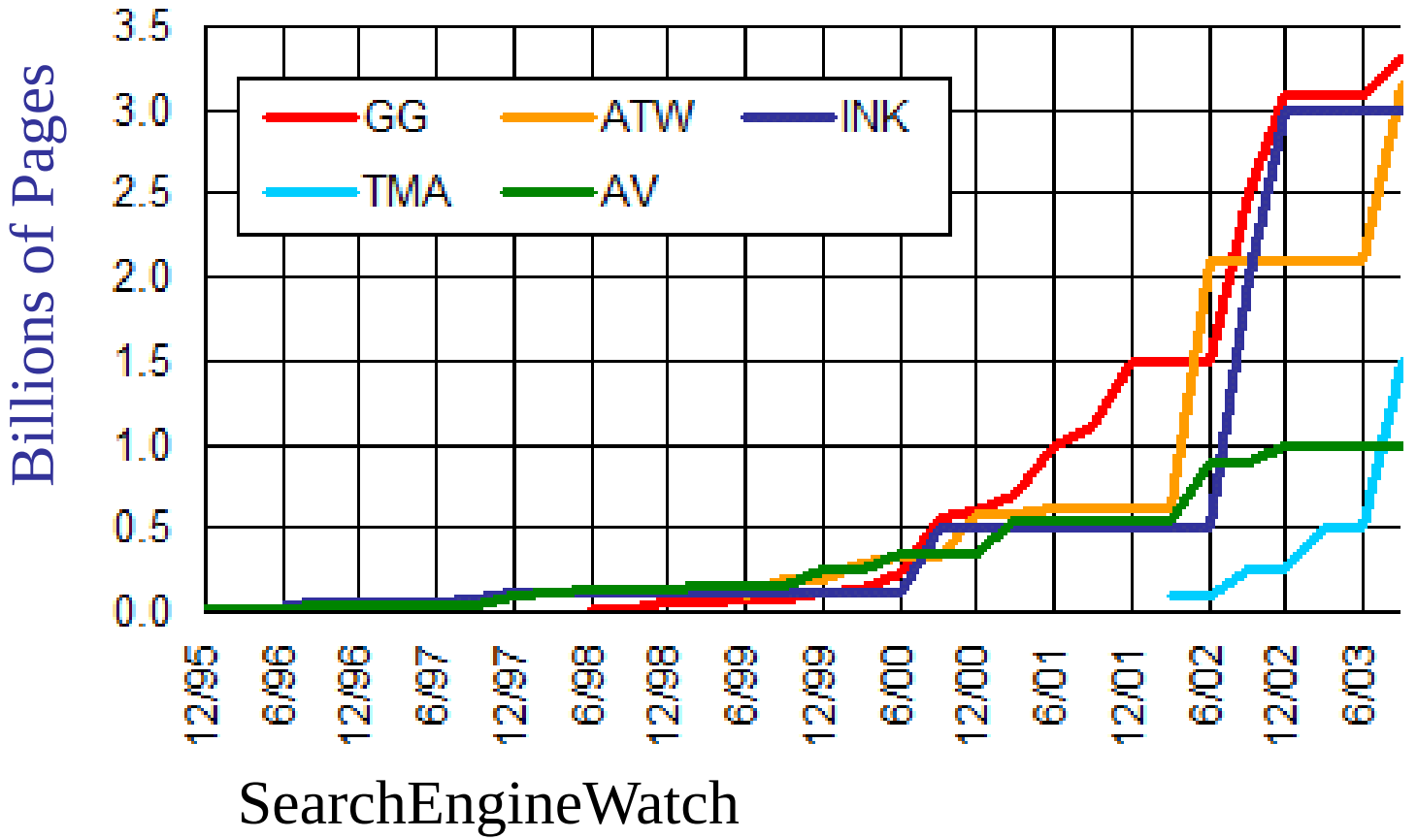
Web Search Recent History

- In 1998, Larry Page and Sergey Brin, Ph.D. students at Stanford, started Google. Main advance is use of *link analysis* to rank results partially based on authority.

Web Challenges for IR

- Distributed Data: Documents spread over millions of different web servers.
- Volatile Data: Many documents change or disappear rapidly (e.g. dead links).
- Large Volume: Billions of separate documents.
- Unstructured and Redundant Data: No uniform structure, HTML errors, up to 30% (near) duplicate documents.
- Quality of Data: No editorial control, false information, poor quality writing, typos, etc.
- Heterogeneous Data: Multiple media types (images, video, VRML), languages, character sets, etc.

Growth of Web Pages Indexed

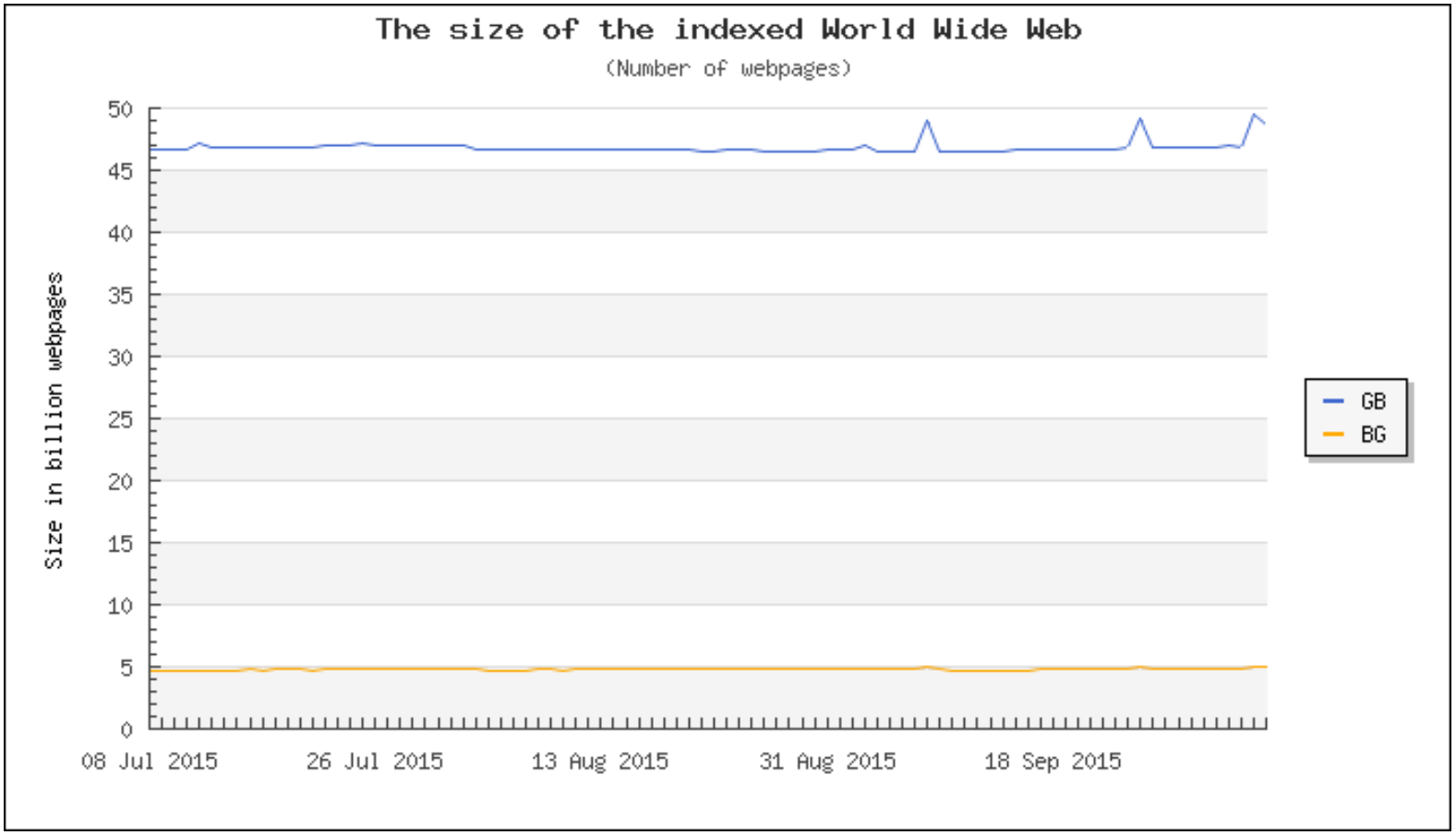


Google
Inktomi
AllTheWeb

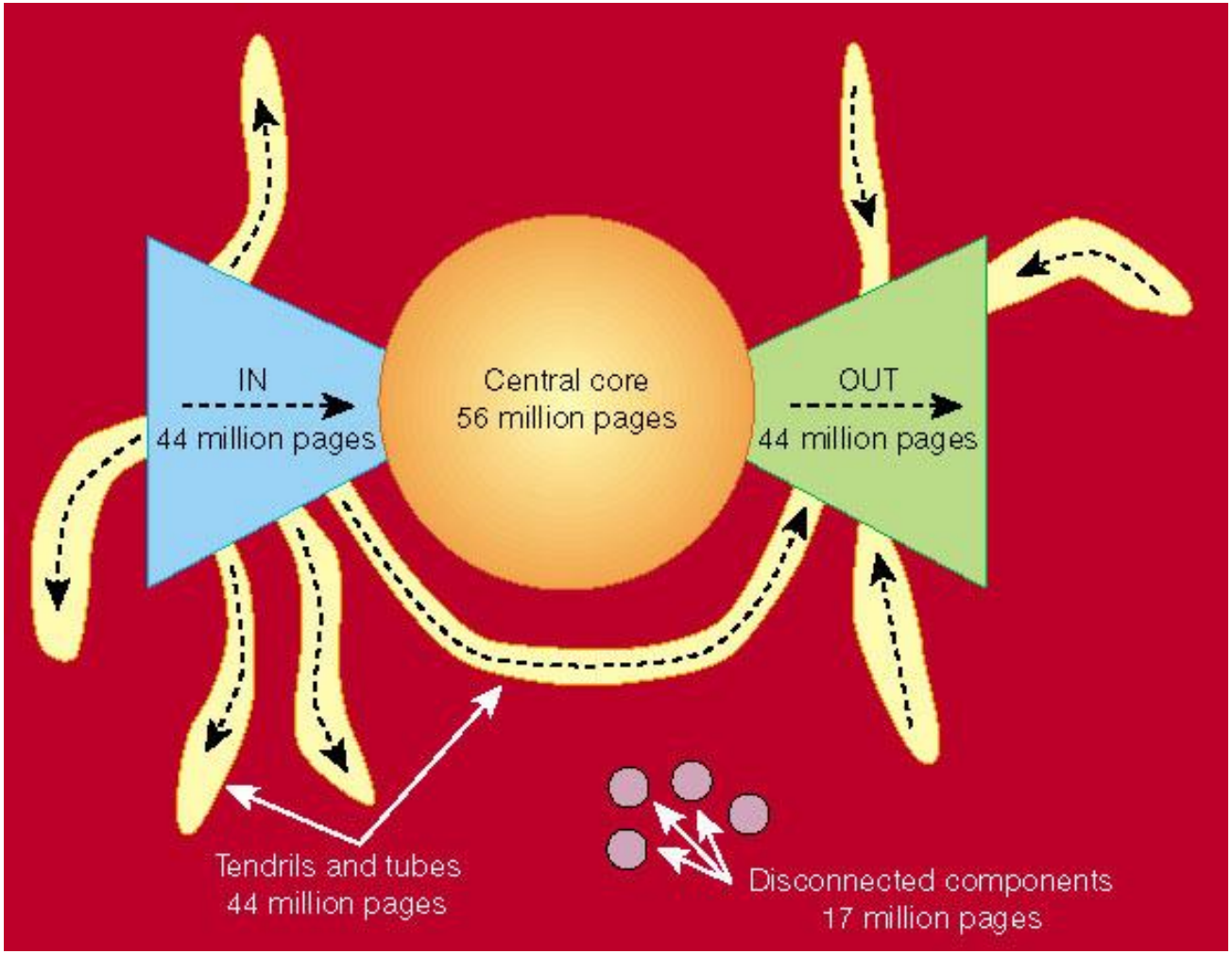
Teoma
Altavista

Assuming 20KB per page,
1 billion pages is about 20 terabytes of data.

Current Size of the Web



Graph Structure in the Web

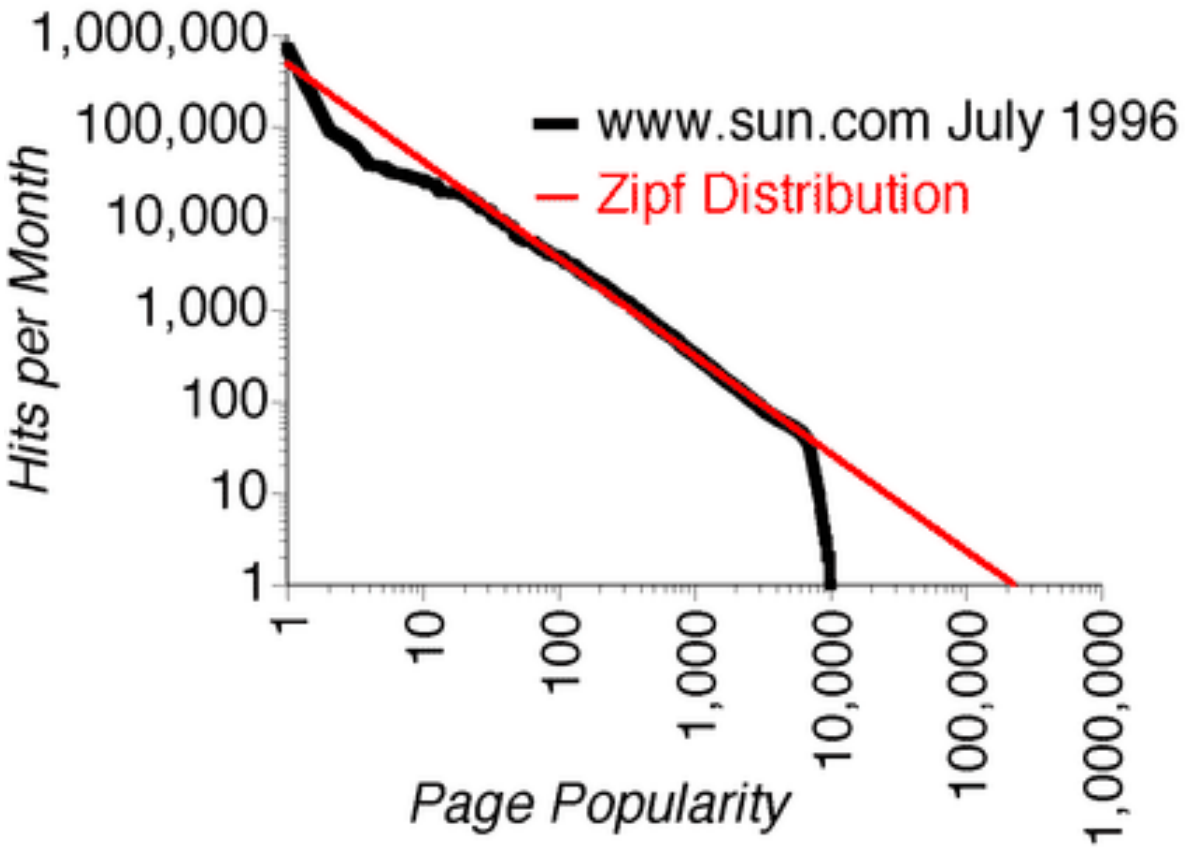


http://www9.org/w9cdrom/160/160.html

Zipf's Law on the Web

- Number of in-links/out-links to/from a page has a Zipfian distribution.
- Length of web pages has a Zipfian distribution.
- Number of hits to a web page has a Zipfian distribution.

Zipfs Law and Web Page Popularity



“Small World” (Scale-Free) Graphs

- Social networks and six degrees of separation.
  - Stanley Milgram Experiment
- Power law distribution of in and out degrees.
- Distinct from purely random graphs.
- “Rich get richer” generation of graphs (preferential attachment).
- Erdos number.
- Networks in biochemistry, roads, telecommunications, Internet, etc are “small word”

Manual Hierarchical Web Taxonomies

- Yahoo approach of using human editors to assemble a large hierarchically structured directory of web pages (closed in 2014).
- Open Directory Project is a similar approach based on the distributed labor of volunteer editors ("net-citizens provide the collective brain"). Used by most other search engines. Started by Netscape.
  - http://www.dmoz.org/
  - Wikipedia:

SIZE OF THE WEB

What is the size of the web ?

- Issues
  - The web is really infinite
    - Dynamic content, e.g., calendars
    - Soft 404: www.yahoo.com/<anything> is a valid page
  - Static web contains syntactic duplication, mostly due to mirroring (~30%)
  - Some servers are seldom connected
- Who cares?
  - Media, and consequently the user
  - Engine design
  - Engine crawl policy. Impact on recall.

What can we attempt to measure?

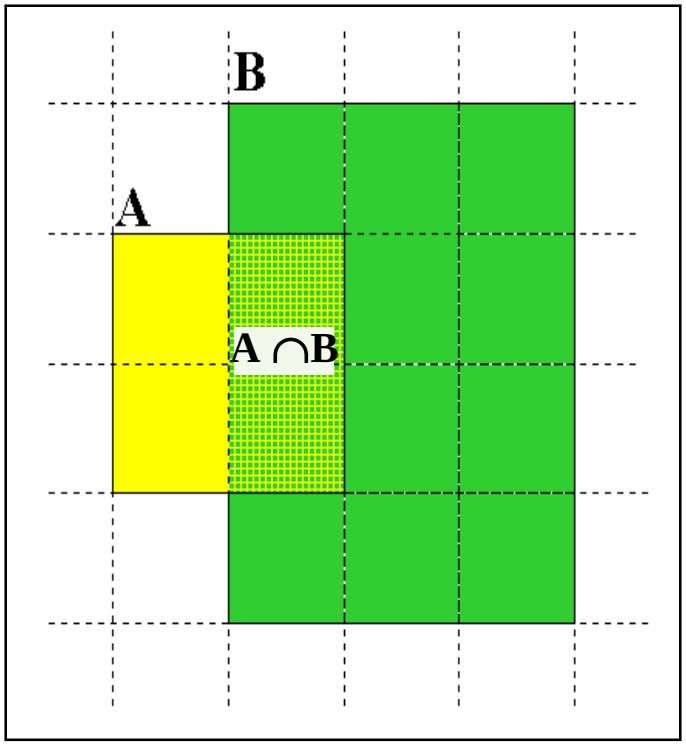
- The relative sizes of search engines
  - The notion of a page being indexed is still *reasonably* well defined.
  - Already there are problems
    - Document extension: e.g., engines index pages not yet crawled, by indexing anchortext.
    - Document restriction: All engines restrict what is indexed (first *n* words, only relevant words, etc.)

New definition?

- The statically indexable web is whatever search engines index.
- IQ is whatever the IQ tests measure.
- Different engines have different preferences
- max url depth, max count/host, anti-spam rules, priority rules, etc.
- Different engines index different things under the same URL:
- frames, meta-keywords, document restrictions, document extensions, ...

Relative Size from Overlap

Given two engines A and B



Sample URLs randomly from A

Check if contained in B and vice versa

$$A \cap B = (1/2) * \text{Size } A$$

$$A \cap B = (1/6) * \text{Size } B$$

$$(1/2)*\text{Size } A = (1/6)*\text{Size } B$$

$$\therefore \text{ Size } A / \text{ Size } B = (1/6)/(1/2) = 1/3$$

Each test involves: (i) Sampling (ii) Checking

Sampling URLs

- Ideal strategy: Generate a random URL and check for containment in each index.
- Problem: Random URLs are hard to find! Enough to generate a random URL contained in a given Engine.
- Approach 1: Generate a random URL contained in a given engine
  - Suffices for the estimation of relative size
- Approach 2: Random walks / IP addresses
  - In theory: might give us a true estimate of the size of the web (as opposed to just relative sizes of indexes)

Statistical methods

- Approach 1
  - Random queries
  - Random searches
- Approach 2
  - Random IP addresses
  - Random walks

Random URLs from random queries

- Generate random query: how?
  - **Lexicon:** 400,000+ words from a web crawl
  - **Conjunctive Queries:** w_1 and w_2
    e.g., vocalists AND rsi

Not an English dictionary

- Get 100 result URLs from engine A
- Choose a random URL as the candidate to check for presence in engine B
- This distribution induces a probability weight W(p) for each page.

Query Based Checking

- *Strong Query* to check whether an engine *B* has a document *D*:
  - Download *D*. Get list of words.
  - Use 8 low frequency words as AND query to *B*
  - Check if *D* is present in result set.
- Problems:
  - Near duplicates
  - Frames
  - Redirects
  - Engine time-outs
  - Is 8-word query good enough?

Advantages & disadvantages

- Statistically sound under the induced weight.
- Biases induced by random query
  - Query Bias: Favors content-rich pages in the language(s) of the lexicon
  - Ranking Bias: *Solution:* Use conjunctive queries & fetch all
  - Checking Bias: Duplicates, impoverished pages omitted
  - Document or query restriction bias: engine might not deal properly with 8 words conjunctive query
  - Malicious Bias: Sabotage by engine
  - Operational Problems: Time-outs, failures, engine inconsistencies, index modification.

Random searches

- Choose random searches extracted from a local log [Lawrence & Giles 97] or build "random searches" [Notess]
  - Use only queries with small result sets.
  - Count normalized URLs in result sets.
  - Use ratio statistics

Advantages & disadvantages

- Advantage
  - Might be a better reflection of the human perception of coverage
- Issues
  - Samples are correlated with source of log
  - Duplicates
  - Technical statistical problems (must have non-zero results, ratio average not statistically sound)

Random searches

- 575 & 1050 queries from the NEC RI employee logs
- 6 Engines in 1998, 11 in 1999
- Implementation:
  - Restricted to queries with < 600 results in total
  - Counted URLs from each engine after verifying query match
  - Computed size ratio & overlap for individual queries
  - Estimated index size ratio & overlap by averaging over all queries

Queries from Lawrence and Giles study

- *adaptive access control*
- *neighborhood preservation topographic*
- *hamiltonian structures*
- *right linear grammar*
- *pulse width modulation neural*
- *unbalanced prior probabilities*
- *ranked assignment method*
- *internet explorer favourites importing*
- *karvel thornber*
- *zili liu*
- *softmax activation function*
- *bose multidimensional system theory*
- *gamma mlp*
- *dvi2pdf*
- *john oliensis*
- *rieke spikes exploring neural*
- *video watermarking*
- *counterpropagation network*
- *fat shattering dimension*
- *abelson amorphous computing*

Random IP addresses

- Generate random IP addresses
- Find a web server at the given address
  - If there's one
- Collect all pages from server
  - From this, choose a page at random

Random IP addresses

- HTTP requests to random IP addresses
  - Ignored: empty or authorization required or excluded
  - [Lawr99] Estimated 2.8 million IP addresses running crawlable web servers (16 million total) from observing 2500 servers.
  - OCLC using IP sampling found 8.7 M hosts in 2001
    - Netcraft [Netc02] accessed 37.2 million hosts in July 2002
- [Lawr99] exhaustively crawled 2500 servers and extrapolated

Advantages & disadvantages

- Advantages
  - Clean statistics
  - Independent of crawling strategies
- Disadvantages
  - Doesn't deal with duplication
  - Many hosts might share one IP, or not accept requests
  - No guarantee all pages are linked to root page.
    - E.g.: employee pages
  - Power law for # pages/hosts generates bias towards sites with few pages.
    - But bias can be accurately quantified IF underlying distribution understood
  - Potentially influenced by spamming (multiple IP's for

Random walks

- View the Web as a directed graph
- Build a random walk on this graph
  - Includes various "jump" rules back to visited sites
    - Does not get stuck in spider traps!
    - Can follow all links!
  - Converges to a stationary distribution
    - Must assume graph is finite and independent of the walk.
    - Conditions are not satisfied (cookie crumbs, flooding)
    - Time to convergence not really known
  - Sample from stationary distribution of walk
  - Use the "strong query" method to check coverage by SE

Advantages & disadvantages

- Advantages
  - "Statistically clean" method, at least in theory!
  - Could work even for infinite web (assuming convergence) under certain metrics.
- Disadvantages
  - List of seeds is a problem.
  - Practical approximation might not be valid.
  - Non-uniform distribution
    - Subject to link spamming

Conclusions Regarding Web Size

- No sampling solution is perfect.
- Lots of new ideas ...
- ....but the problem is getting harder
- Quantitative studies are fascinating and a good research problem

Business Models for Web Search

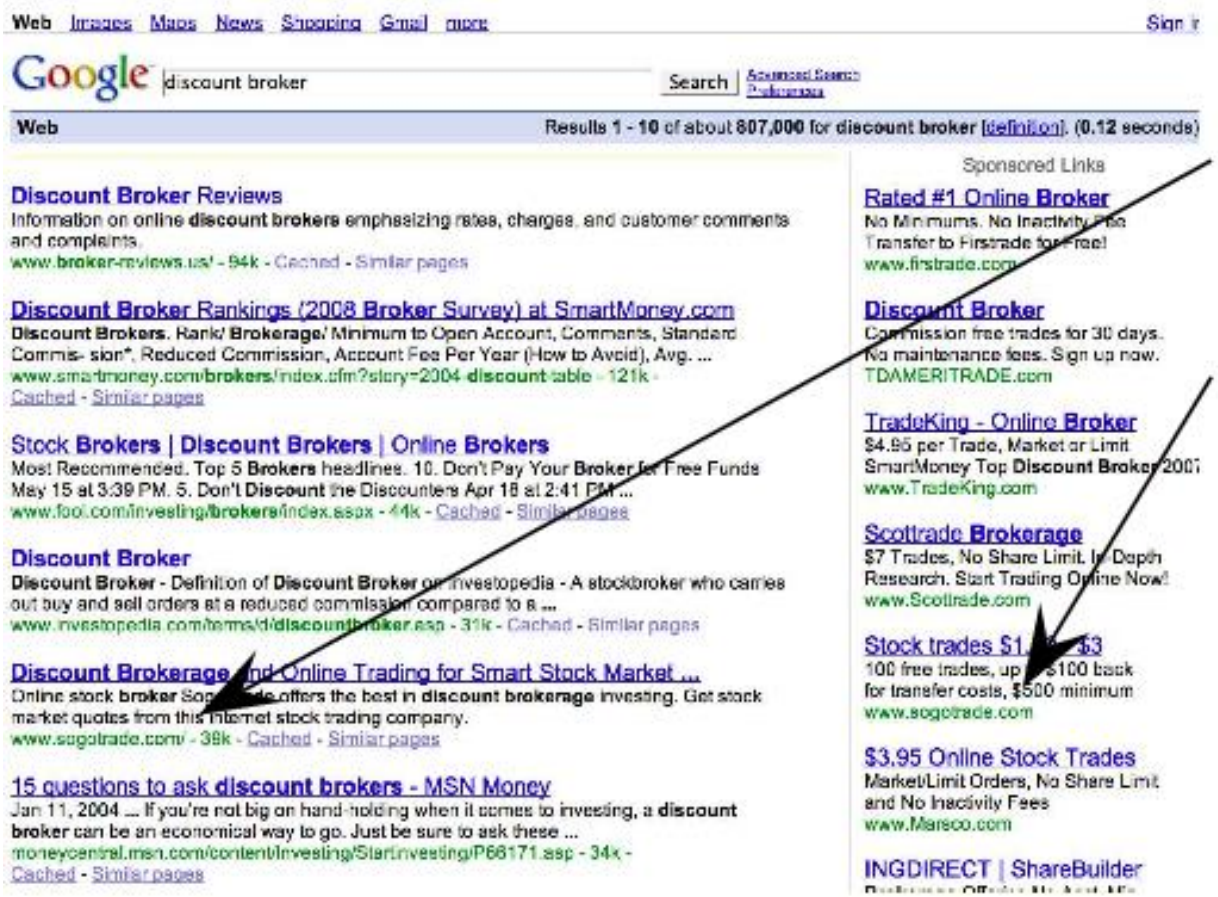
- Advertisers pay for banner ads on the site that do not depend on a user's query.
  - CPM: Cost Per Mille (thousand impressions). Pay for each ad display.
  - CPC: Cost Per Click. Pay only when user clicks on ad.
  - CTR: Click Through Rate. Fraction of ad impressions that result in clicks throughs. CPC = CPM / (CTR * 1000)
  - CPA: Cost Per Action (Acquisition). Pay only when user actually makes a purchase on target site.
- Advertisers bid for "keywords". Ads for highest bidders displayed when user query contains a purchased keyword.
  - PPC: Pay Per Click. CPC for bid word ads (e.g. Google AdWords).

History of Business Models

- Initially, banner ads paid thru CPM were the norm.
- GoTo Inc. formed in 1997 and originates and patents bidding and PPC business model.
- Google introduces AdWords in fall 2000.
- GoTo renamed Overture in Oct. 2001.
- Overture sues Google for use of PPC in Apr. 2002.
- Overture acquired by Yahoo in Oct. 2003.
- Google settles with Overture/Yahoo for 2.7 million shares of Class A common stock in Aug. 2004.

First generation of search ads: Goto (1996)

Two ranked lists: web pages (left) and ads (right)



SogoTrade appears in search results.

SogoTrade appears in ads.

Do search engines rank advertisers higher than non-advertisers?

All major search engines claim no.

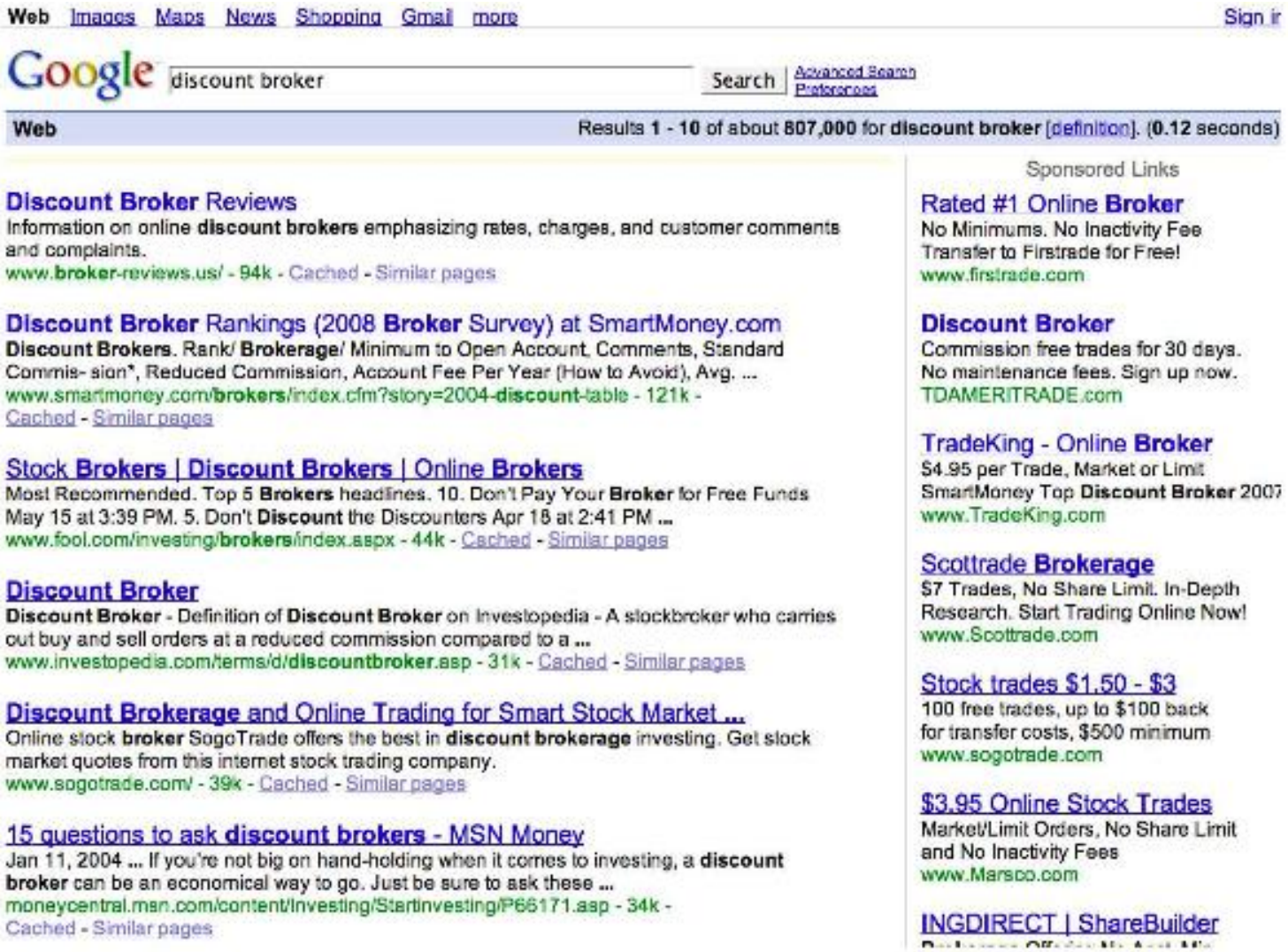
36

Do ads influence editorial content?

- Similar problem at newspapers / TV channels
- A newspaper is reluctant to publish harsh criticism of its major advertisers.
- The line often gets blurred at newspapers / on TV.
- No known case of this happening with search engines yet?

37

How are the ads on the right ranked?



How are ads ranked?

- Advertisers bid for keywords – sale by auction.
- Open system: Anybody can participate and bid on keywords.
- Advertisers are only charged when somebody clicks on your ad.
- How does the auction determine an ad's rank and the price paid for the ad?
- Basis is a second price auction, but with twists
- For the bottom line, this is perhaps the most important research area for search engines – computational advertising.
  - Squeezing an additional fraction of a cent from each ad means billions of additional revenue for the search engine.

39

How are ads ranked? Auction: المزادSecond Price Encourage higher bids with limited risk!

- First cut: according to bid price `a la Goto
  - Bad idea: open to abuse
  - Example: query [does my husband cheat?] → ad for divorce lawyer
  - We don't want to show nonrelevant ads.
- Instead: rank based on bid price and relevance
- Key measure of ad relevance: clickthrough rate
  - clickthrough rate = CTR = clicks per impressions
- Result: A nonrelevant ad will be ranked low.
  - Even if this decreases search engine revenue short-term
  - Hope: Overall acceptance of the system and overall revenue is maximized if users get useful information.
- Other ranking factors: location, time of day, quality and loading speed of landing page
- The main ranking factor: the query

Google's second price auction

advertiser	bid	CTR	ad rank	rank	paid
A	$4.00	0.01	0.04	4	(minimum)
B	$3.00	0.03	0.09	2	$2.68
C	$2.00	0.06	0.12	1	$1.51
D	$1.00	0.08	0.08	3	$0.51

- bid: maximum bid for a click by advertiser
- CTR: click-through rate: when an ad is displayed, what percentage of time do users click on it? CTR is a measure of relevance.
- ad rank: bid × CTR: this trades off (i) how much money the advertiser is willing to pay against (ii) how relevant the ad is
- rank: rank in auction
- paid: second price auction price paid by advertiser

Google's second price auction

advertiser	bid	CTR	ad rank	rank	paid
A	$4.00	0.01	0.04	4	(minimum)
B	$3.00	0.03	0.09	2	$2.68
C	$2.00	0.06	0.12	1	$1.51
D	$1.00	0.08	0.08	3	$0.51

Second price auction: The advertiser pays the minimum amount necessary to maintain their position in the auction (plus 1 cent).

$price_1 \times CTR_1 = bid_2 \times CTR_2$ (this will result in $rank_1 = rank_2$)

$price_1 = bid_2 \times CTR_2 / CTR_1$

$p_1 = bid_2 \times CTR_2/CTR_1 = 3.00 \times 0.03/0.06 = 1.50$
$p_2 = bid_3 \times CTR_3/CTR_2 = 1.00 \times 0.08/0.03 = 2.67$
$p_3 = bid_4 \times CTR_4/CTR_3 = 4.00 \times 0.01/0.08 = 0.50$

42

Keywords with high bids

According to http://www.cwire.org/highest-paying-search-terms/

$69.1	mesothelioma treatment options
$65.9	personal injury lawyer michigan
$62.6	student loans consolidation
$61.4	car accident attorney los angeles
$59.4	online car insurance quotes
$59.4	arizona dui lawyer
$46.4	asbestos cancer
$40.1	home equity line of credit
$39.8	life insurance quotes
$39.2	refinancing
$38.7	equity line of credit
$38.0	lasik eye surgery new york city
$37.0	2nd mortgage
$35.9	free car insurance quote

43

Search ads: A win-win-win?

- The search engine company gets revenue every time somebody clicks on an ad.
- The user only clicks on an ad if they are interested in the ad.
  - Search engines punish misleading and nonrelevant ads.
  - As a result, users are often satisfied with what they find after clicking on an ad.
- The advertiser finds new customers in a cost-effective way.

44

Not a win-win-win: Keyword arbitrage

- Buy a keyword on Google
- Then redirect traffic to a third party that is paying much more than you are paying Google.
  - E.g., redirect to a page full of ads
- This rarely makes sense for the user.
- Ad spammers keep inventing new tricks.
- The search engines need time to catch up with them.

45

Not a win-win-win: Violation of trademarks

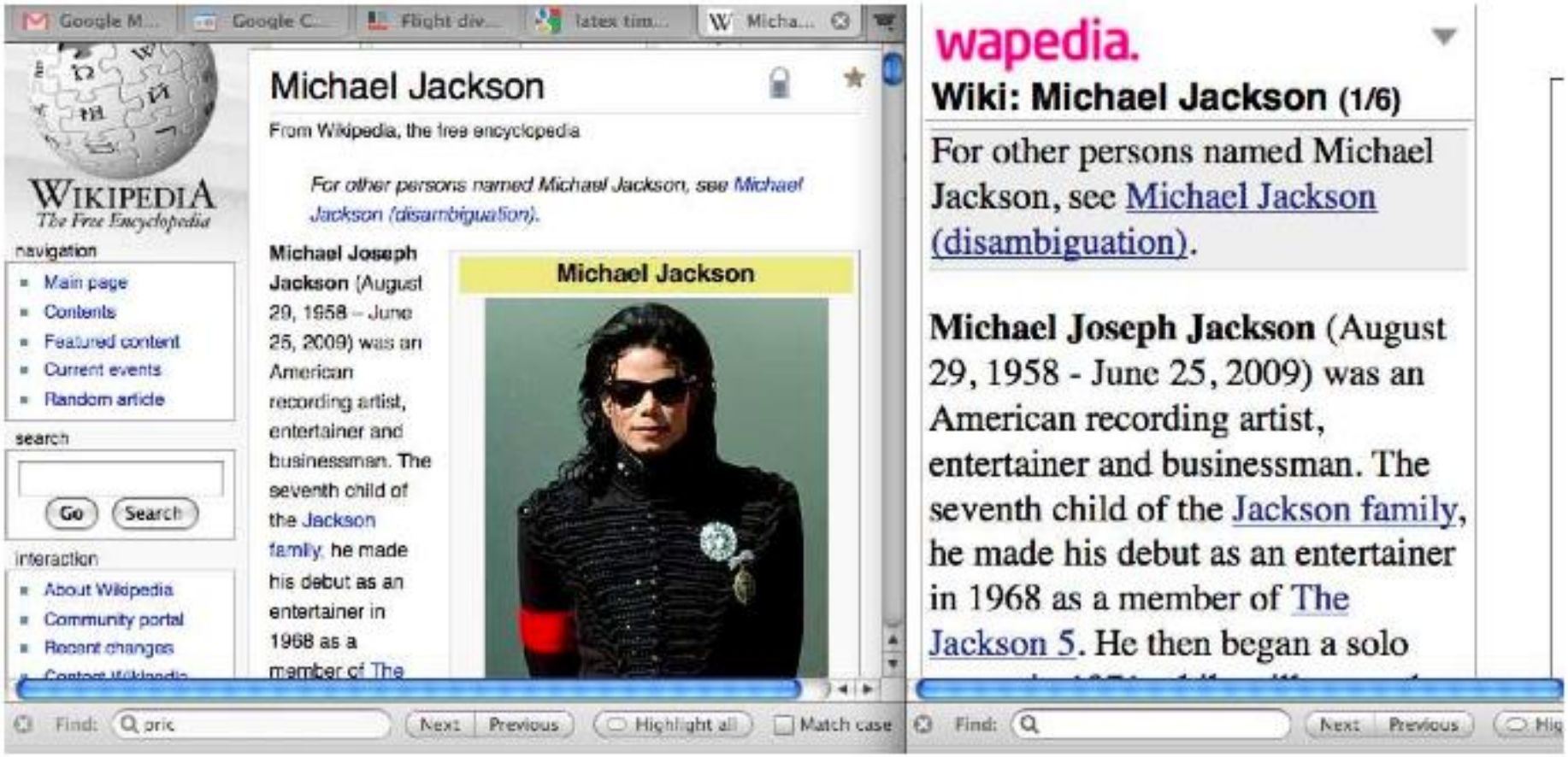
- Example: geico
- During part of 2005: The search term “geico” on Google was bought by competitors.
- Geico lost this case in the United States.
- Louis Vuitton lost similar case in Europe.
- See http://google.com/tm complaint.html
- It’s potentially misleading to users to trigger an ad off of a trademark if the user can’t buy the product on the site.

46

(Near) Duplicate Detection

- The web is full of duplicated content.
- More so than many other collections
- Exact duplicates
  - Easy to eliminate
  - E.g., use hash/fingerprint
- Near-duplicates
  - Abundant on the web
  - Difficult to eliminate
- For the user, it's annoying to get a search result with near-identical documents.
- Marginal relevance is zero: even a highly relevant document becomes nonrelevant if it appears below a (near-)duplicate.
- We need to eliminate near-duplicates.

Near-duplicates: Example



48

Detecting near-duplicates

- Compute similarity with an edit-distance measure
- We want "syntactic" (as opposed to semantic) similarity.
  - True semantic similarity (similarity in content) is too difficult to compute.
- We do not consider documents near-duplicates if they have the same content, but express it with different words.
- Use similarity threshold θ to make the call "is/isn't a near-duplicate".
- E.g., two documents are near-duplicates if similarity $> \theta = 80\%$.

49

Represent each document as set of **shingles**

- A shingle (قطعة قرميد)is simply a word n-gram.
- Shingles are used as features to measure syntactic similarity of documents.
- For example, for $n = 3$, "a rose is a rose is a rose" would be represented as this set of shingles:
  - { a-rose-is, rose-is-a, is-a-rose }
- We can map shingles to $1..2^m$ (e.g., $m = 64$, quite large) by fingerprinting.
- From now on: s_k refers to the shingle's fingerprint (map) in $1..2^m$.
- We define the similarity of two documents as the Jaccard coefficient of their shingle sets.

50

Recall: Jaccard coefficient

- A commonly used measure of overlap of two sets
- Let A and B be two sets
- Jaccard coefficient:

$$\text{JACCARD}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

$$(A \neq \emptyset \text{ or } B \neq \emptyset)$$

- JACCARD(A,A) = 1
- JACCARD(A,B) = 0 if $A \cap B = 0$
- A and B don't have to be the same size.
- Always assigns a number between 0 and 1.

51

Jaccard coefficient: Example

- Three documents:

  d_1: "Jack London traveled to Oakland"

  d_2: "Jack London traveled to the city of Oakland"

  d_3: "Jack traveled from Oakland to London"

- Based on shingles of size 2 (2-grams or bigrams), what are the Jaccard coefficients $J(d_1, d_2)$ and $J(d_1, d_3)$?
- $J(d_1, d_2) = 3/8 = 0.375$
- $J(d_1, d_3) = 0$
- Note: very sensitive to dissimilarity

52

Represent each document as a sketch

- The number of shingles per document is large (why?).
- To increase efficiency, we will use a sketch, a cleverly chosen subset of the shingles of a document.
- The size of each sketch is, say, $n = 200$ . . .
- . . . and is defined by a set of permutations $\pi_1 \ldots \pi_{200}$.
- Each π_i is a random permutation on $1..2^m$
- The sketch of d is defined as:

$$< \min_{s \in d} \pi_1(s), \min_{s \in d} \pi_2(s), \ldots, \min_{s \in d} \pi_{200}(s) >$$

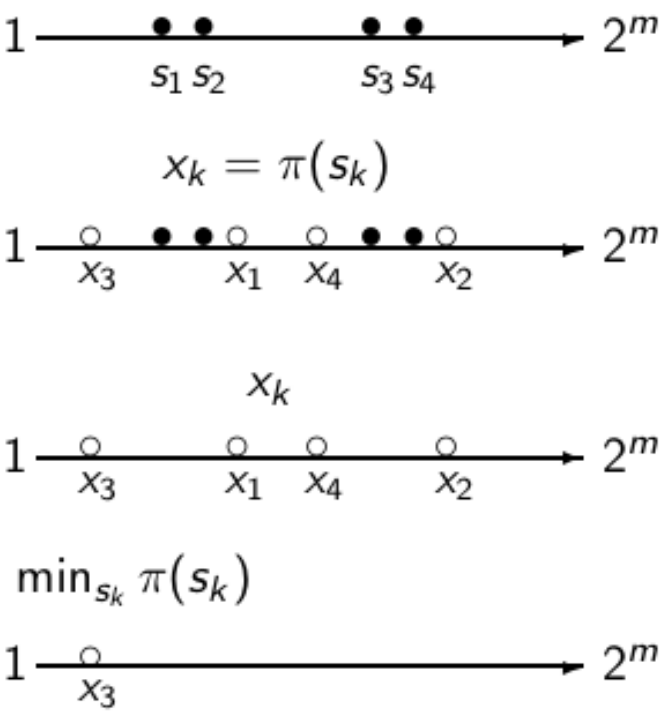
(a vector of 200 numbers).

So we have 200 permutations that are computed on all docs

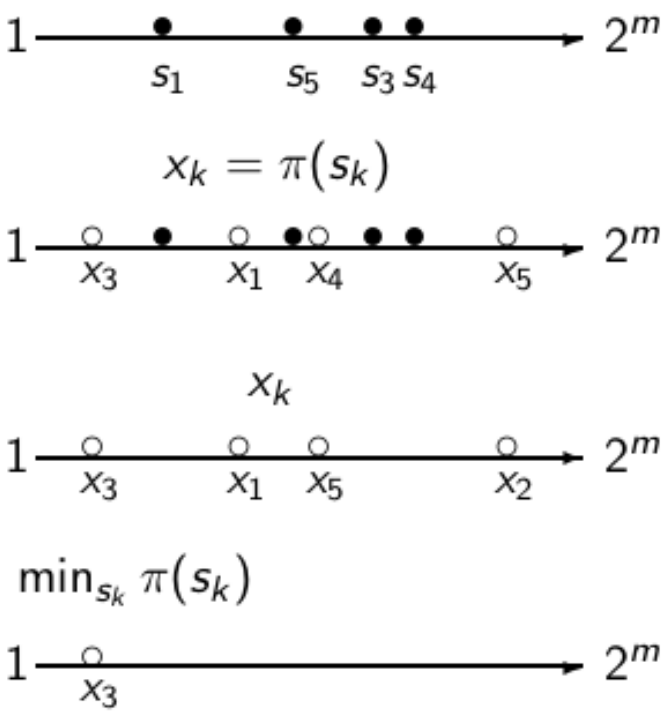
53

The Permutation and minimum: Example

document 1: $\{s_k\}$



document 2: $\{s_k\}$



We use $\min_{s \in d1} \pi(s) = \min s \in d_2\ \pi(s)$ as a test for: are d_1 and d_2 near-duplicates? In this case: permutation π says: $d_1 \approx d_2$

54

Computing Jaccard for sketches

- Sketches: Each document is now a vector of $n = 200$ numbers.
- Much easier to deal with than the very high-dimensional space of shingles
- But how do we compute Jaccard?

55

Computing Jaccard for sketches (2)

- How do we compute Jaccard?
- Let U be the union of the set of shingles of d_1 and d_2 and I the intersection.
- There are $|U|!$ permutations on U.
- For $s' \in I$, for how many permutations π do we have
  $\text{argmin}_{s \in d1}\ \pi(s) = s' = \text{argmin}_{s \in d2}\ \pi(s)$?
- Answer: $(|U| - 1)!$
- There is a set of $(|U| - 1)!$ different permutations for each s in I. $\Rightarrow |I|(|U| - 1)!$ permutations make
  $\text{argmin}_{s \in d1}\ \pi(s) = \text{argmin}_{s \in d2}\ \pi(s)$ true
- Thus, the proportion of permutations that make
  $\min_{s \in d1}\ \pi(s) = \min_{s \in d2}\ \pi(s)$ true is:

$$\frac{|I|(|U| - 1)!}{|U|!} = \frac{|I|}{|U|} = J(d_1, d_2)$$

56

Estimating Jaccard

- Thus, the proportion of successful permutations is the Jaccard coefficient.
  - Permutation π is successful iff $\min_{s \in d1} \pi(s) = \min_{s \in d2} \pi(s)$
- Picking a permutation at random and outputting 1 (successful) or 0 (unsuccessful) is a Bernoulli trial.
- Estimator of probability of success: proportion of successes in n Bernoulli trials. ($n = 200$)
- Our sketch is based on a random selection of permutations.
- Thus, to compute Jaccard, count the number k of successful permutations for $< d_1, d_2 >$ and divide by $n = 200$.
- $k/n = k/200$ estimates $J(d_1, d_2)$.

57

Implementation

- We use hash functions as an efficient type of permutation:

  $h_i : \{1..2^m\} \rightarrow \{1..2^m\}$

- Scan all shingles s_k in union of two sets in arbitrary order
- For each hash function h_i and documents d_1, d_2, . . .: keep slot for minimum value found so far
- If h_i (s_k) is lower than minimum found so far: update slot

58

Example (2 permutations: mod functions)

	d_1	d_2
s_1	1	0
s_2	0	1
s_3	1	1
s_4	1	0
s_5	0	1

$h(x) = x \bmod 5$

$g(x) = (2x + 1) \bmod 5$

$\min(h(d_1)) = 1 \neq 0 = \min(h(d_2))$ $\min(g(d_1)) = 2 \neq 0 = \min(g(d_2))$

$\hat{J}(d_1, d_2) = \frac{0+0}{2} = 0$

	d_1 slot		d_2 slot	
h		∞		∞
g		∞		∞
$h(1) = 1$	1	1	0	∞
$g(1) = 3$	3	3	–	∞
$h(2) = 2$	0	1	2	2
$g(2) = 0$	–	3	0	0
$h(3) = 3$	3	1	3	2
$g(3) = 2$	2	2	2	0
$h(4) = 4$	4	1	0	2
$g(4) = 4$	4	2	–	0
$h(5) = 0$	–	1	0	0
$g(5) = 1$	0	2	1	0

final sketches

59

Exercise

	d_1	d_2	d_3
s_1	0	1	1
s_2	1	0	1
s_3	0	1	0
s_4	1	0	0

$h(x) = 5x + 5 \bmod 4$ Estimate $\hat{J}(d_1, d_2)$,

$g(x) = (3x + 1) \bmod 4$

$\hat{J}(d_1, d_3)$, $\hat{J}(d_2, d_3)$

60

Solution (1)

	d_1	d_2	d_3
s_1	0	1	1
s_2	1	0	1
s_3	0	1	0
s_4	1	0	0

$h(x) = 5x + 5 \bmod 4$

$g(x) = (3x + 1) \bmod 4$

	d_1 slot	d_2 slot	d_3 slot
	∞	∞	∞
	∞	∞	∞
$h(1) = 2$	– ∞	2 2	2 2
$g(1) = 0$	– ∞	0 0	0 0
$h(2) = 3$	3 3	– 2	3 2
$g(2) = 3$	3 3	– 0	3 0
$h(3) = 0$	– 3	0 0	– 2
$g(3) = 2$	– 3	2 0	– 0
$h(4) = 1$	1 1	– 0	– 2
$g(4) = 1$	1 1	– 0	– 0

final sketches

61

Solution (2)

$$\hat{J}(d_1, d_2) = \frac{0+0}{2} = 0$$
$$\hat{J}(d_1, d_3) = \frac{0+0}{2} = 0$$
$$\hat{J}(d_2, d_3) = \frac{0+1}{2} = 1/2$$

62

Shingling: Summary

- Input: N documents
- Choose n-gram size for shingling, e.g., $n = 5$
- Pick 200 random permutations, represented as hash functions
- Compute N sketches: 200 × N matrix shown on previous slide, one row per permutation, one column per document
- Compute $\frac{N \cdot (N-1)}{2}$ pairwise similarities
- Transitive closure of documents with similarity > θ
- Index only one document from each equivalence class

63

Efficient near-duplicate detection

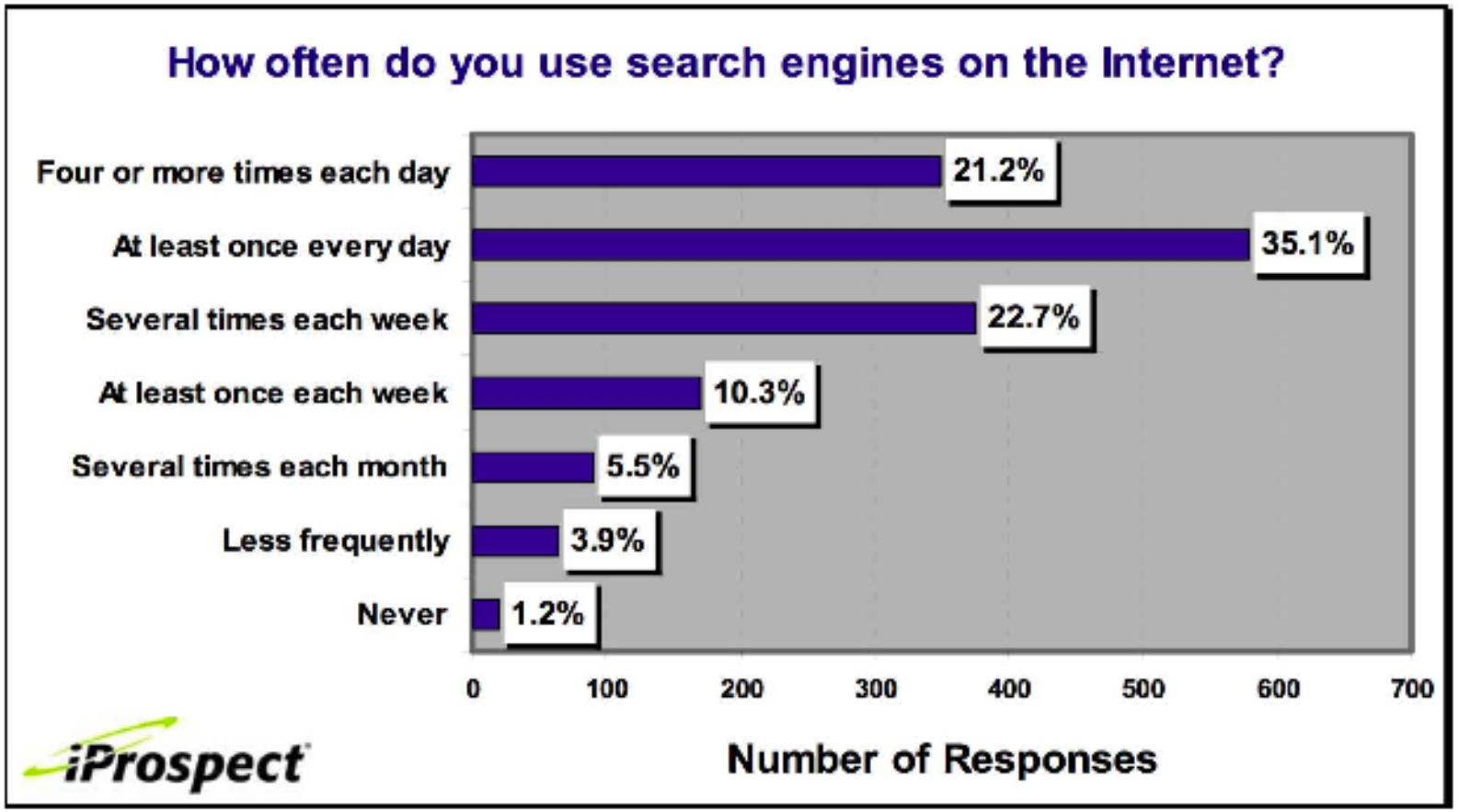
- Now we have an extremely efficient method for estimating a Jaccard coefficient for a single pair of two documents.
- But we still have to estimate $O(N^2)$ coefficients where N is the number of web pages.
- Still intractable
- One solution: locality sensitive hashing (LSH)
- Another solution: sorting (Henzinger 2006)

64

User Interfaces

- HTML supports various types of program input in forms, including:
  - Text boxes
  - Menus
  - Check boxes
  - Radio buttons
- When user submits a form, string values for various *parameters* are sent to the server program for processing.
- Server program uses these values to compute an appropriate HTML response page.

Search is the top activity on the web



66

User Query Length

- Users tend to enter short queries.
  - Study in 1998 gave average length of 2.35 words.
- More recent evidence that queries are getting longer.

Percentage of U.S. clicks by number of keywords				
Subject	**Jan-08**	**Dec-08**	**Jan-09**	**Year-over-year percent change**
1 word	20.96%	20.70%	20.29%	**-3%**
2 words	24.91%	24.13%	23.65%	**-5%**
3 words	22.03%	21.94%	21.92%	**0%**
4 words	14.54%	14.67%	14.89%	**2%**
5 words	8.20%	8.37%	8.68%	**6%**
6 words	4.32%	4.47%	4.65%	**8%**
7 words	2.23%	2.40%	2.49%	**12%**
8+ words	2.81%	3.31%	3.43%	**22%**
Note: Data is based on four-week rolling periods (ending Jan. 31, 2009; Dec. 27, 2008; and Jan. 26, 2008) from the Hitwise sample of 10 million U.S. Internet users.				
Source: Hitwise, an Experian company				