X-Informatics
Web Search and Text Mining Part II

2013
Geoffrey Fox
gcf@indiana.edu
http://www.infomall.org/X-InformaticsSpring2013/index.html

Associate Dean for Research,
School of Informatics and Computing
Indiana University Bloomington
2013

Big Data Ecosystem in One Sentence

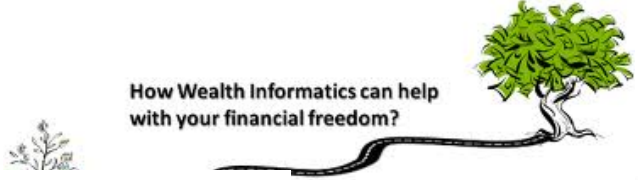
Use Clouds running Data Analytics Collaboratively processing Big Data to solve problems in X-Informatics (or e-X)

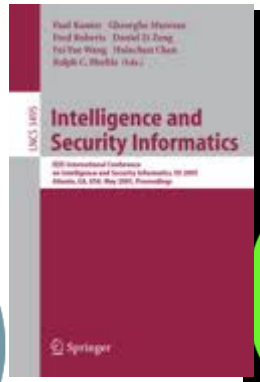
X = Astronomy, Biology, Biomedicine, Business, Chemistry, Climate, Crisis, Earth Science, Energy, Environment, Finance, Health, Intelligence, Lifestyle, Marketing, Medicine, Pathology, Policy, Radar, Security, Sensor, Social, Sustainability, Wealth and Wellness with more fields (physics) defined implicitly

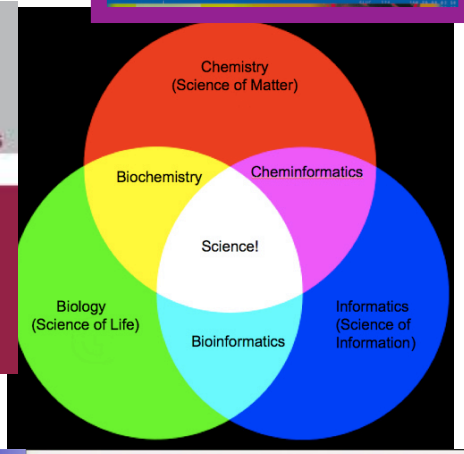
Spans Industry and Science (research)

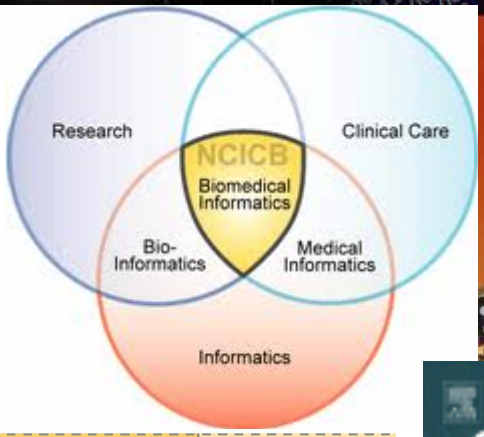
Education: Data Science see recent New York Times articles

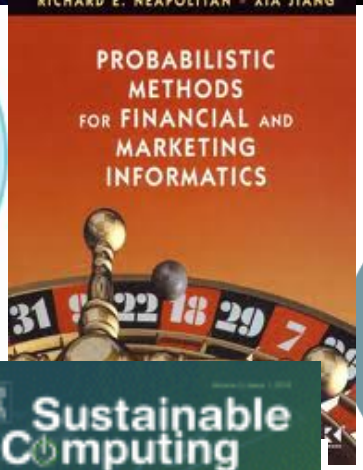
http://datascience101.wordpress.com/2013/04/13/new-york-times-data-science-articles/

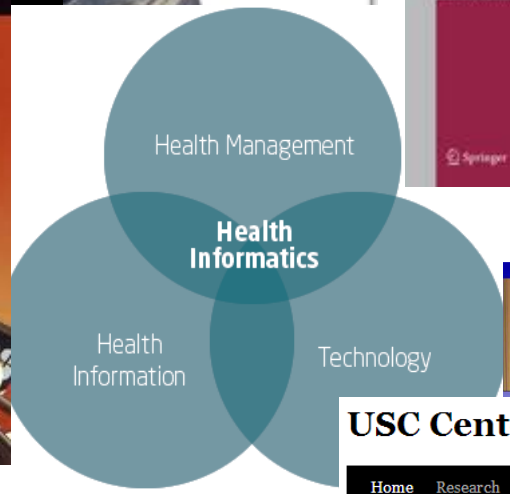


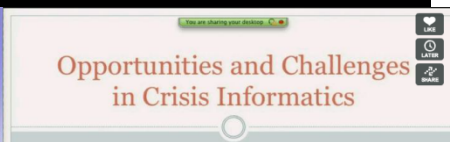


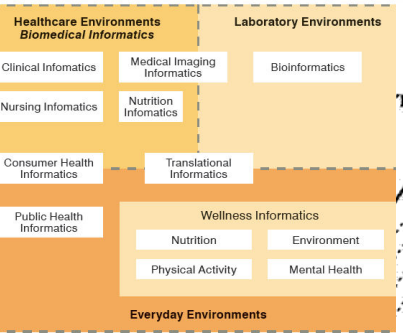


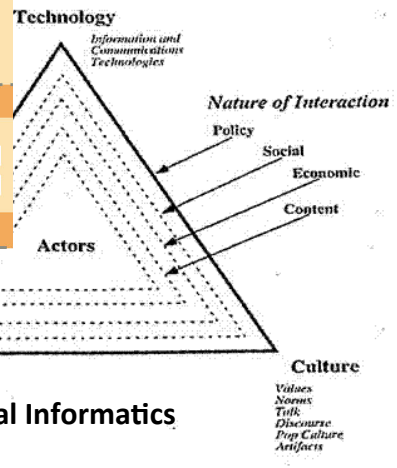








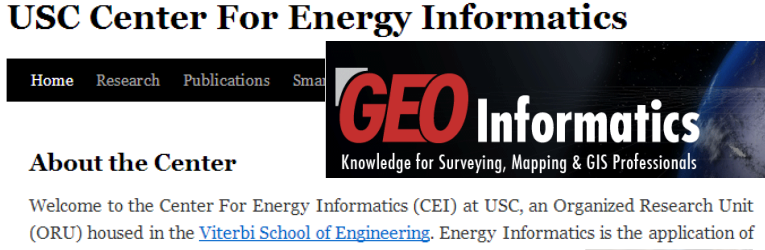


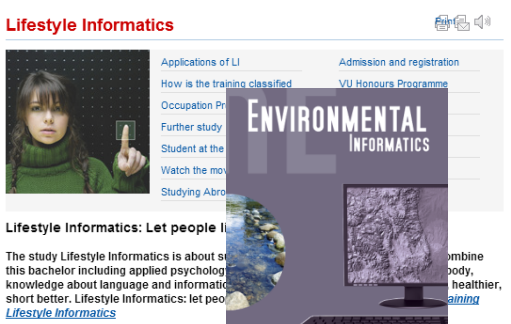


Noelia Penelope Greer (Ed.)

Business Informatics

Information technology, Management,





Data Analytics for Web Search

"Web Data Analytics"

1) Get the digital data (from web or from scanning)
    1) How to crawl web (? Solved "engineering" problem)
2) Pre process data to get searchable things (words, positions)
3) Form Inverted Index mapping words to documents
4) Rank relevance of documents: **PageRank**
5) Lots of technology for advertising, "reverse engineering" "preventing reverse engineering"
6) **1) 2) 3) 5) above are easily parallel over documents**
    - **4) is a nontrivial parallel algorithm over PageRank as different PageRanks are closely coupled by iteration**
7) Structure of the Internet and its people and pages (research but also commercially important)
8) Clustering of documents into topics (as in Google News)
    1) A lot of research – not clear what works
9) We noted value of Bayes converting Mathematics of frequency into Mathematics of belief

Document Preparation

Document Preparation

INPUT DOCUMENT

DOCUMENT TEXT

Y2K Around the World

As computers all over the world switched to 2000, few *Y2K bugs* were reported in several labs. […]

TOKENIZATION

y2k around the world as computers all over the world switched to 2000 few y2k bugs were reported in several labs […]

FILTRATION

y2k world computers world switched 2000 y2k bugs reported labs […]

STEMMING

y2k world computer world switch 2000 y2k bug report lab […]

DOCUMENT REPRESENTATION

y2k (2), world (2), computer (1), switch (1), 2000 (1), bug (1), report (1), lab (1)

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

Character Sequence Decoding

DOCUMENT TEXT

Y2K Around the World

As computers all over the world switched to 2000, few *Y2K bugs* were reported in several labs. […]

- Let's assume some document has to be indexed
- **First step: Getting a textual representation**
- Sounds easy, but might be pretty complicated
  - Many different document formats: DOC, plain text, PDF, HTML, XLS, PPT, RTF, XML, …

Some OCR pretty flaky!

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

Tokenization

DOCUMENT TEXT

Y2K Around the World

As computers all over the world switched to 2000, few *Y2K bugs* were reported in several labs. […]

TOKENIZATION

y2k around the world as computers all over the world switched to 2000 few y2k bugs were reported in several labs […]

- **Tokenization:**
  - Remove formatting information (e.g. HTML tags)
  - Remove punctuation
  - Carry out basic normalization (e.g. remove capitalization)
  - **Goal:** Convert the text into a **sequence of "tokens"**

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

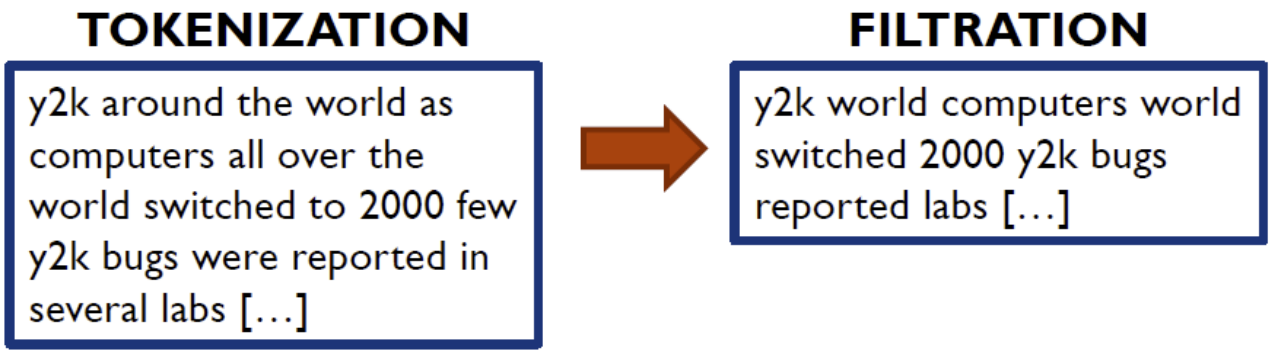
Tokenization (5)

- **Normalization** handles most of the problematic cases
- Define **equivalence classes** of character sequences that get mapped to the same token
  - U.S.A. and USA
  - naïve and naive
- Define these classes implicitly by **transformation rules**
  - **Omit all accents**
  - **Remove periods** between two characters, where there is no whitespace around (e.g. in U.S.A.)
  - Do **case folding,** i.e. reduce all letters to lower case
  - Maybe you need exceptions for names: windows ≠ Windows (Not important, since users ask **queries in lowercase** anyway)

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

Filtration

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

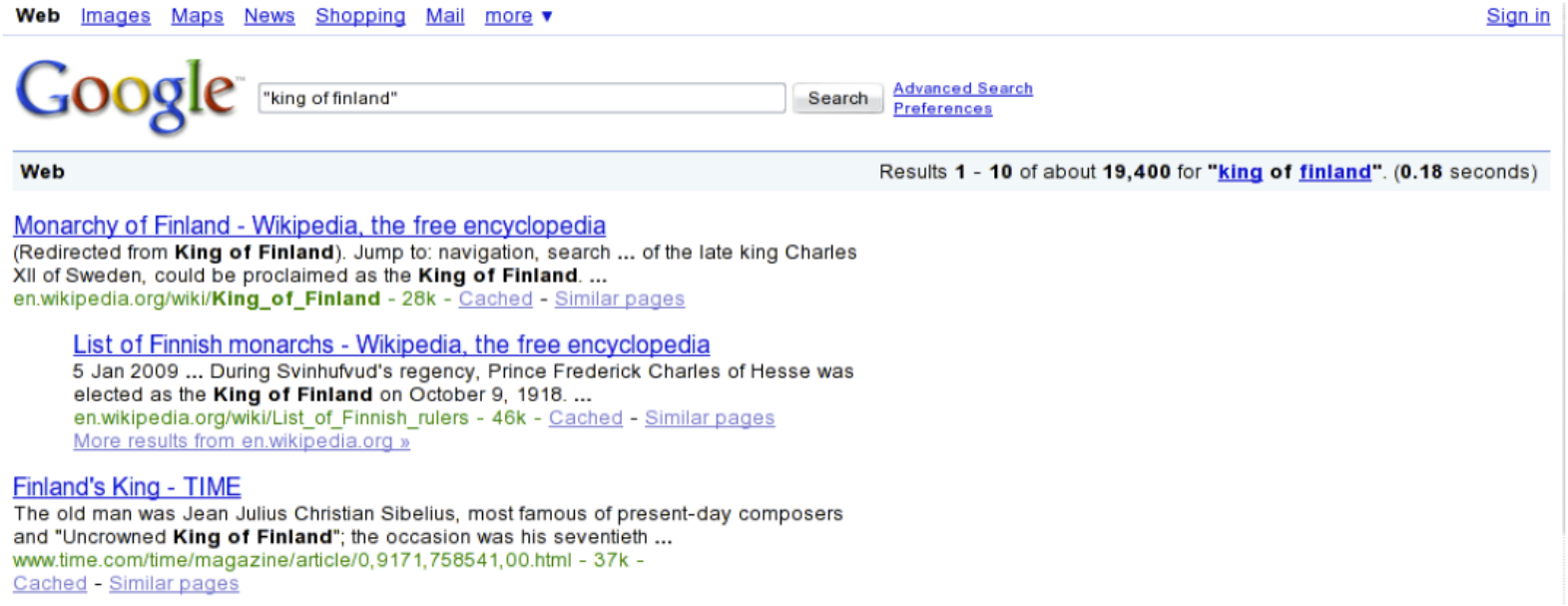


- Removal of **stop words!**
- **Stop words:**
  Extremely common words, which are of little value in selecting which documents match a user's query
- **Examples:** a, an, and, are, as, at, be, by, for, from, has, he, in, is, it, its, of, on, that, the, to, was, were, which, will, with
- "to be or not to be"?

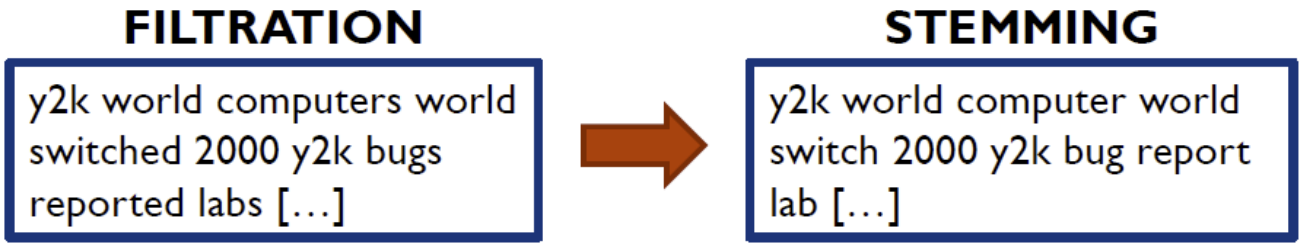
Filtration (2)

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

- In classical IR systems, stop words have been rigorously deleted
- But stop words are needed for **phrase queries,** e.g. “King of Finland” or “As We May Think”
- For example, **Google** does not remove stop words:



Stemming

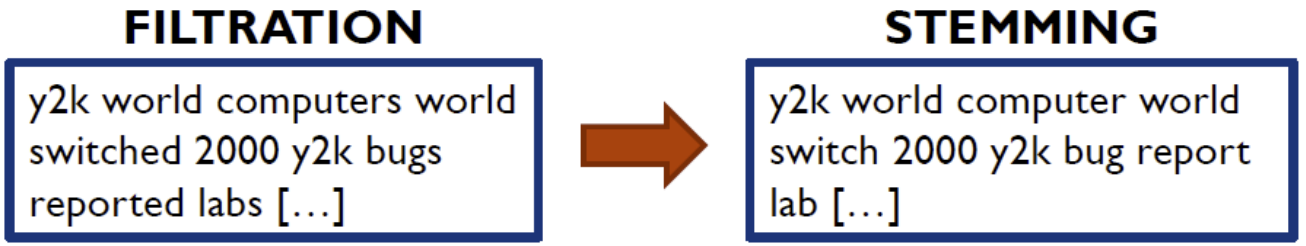


- **Lemmatization:**
  The process of grouping together the different **inflected forms** of a word, e.g. walking → walk, better → good

- **Stemming:**
  Trying to do lemmatization using crude heuristics, usually without knowledge of the word's context or any rules of grammar, e.g. walking → walk, but better → better

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

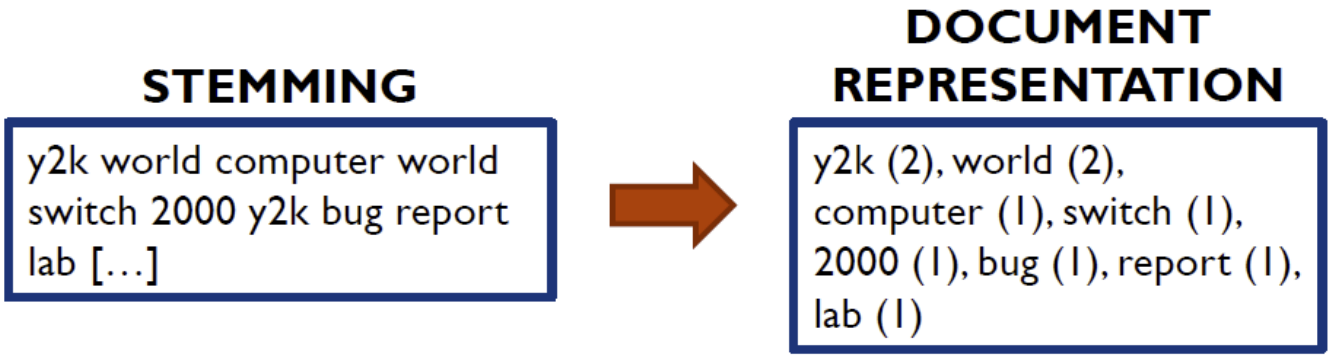
Further Normalization



- There are **some additional** things apart from stemming that **could** be done during this step
- Take care of **umlauts** and **accents**
  - Usually, just remove them (e.g. ä → a) or transliterate them (e.g. ä → ae)
  - But be careful: unbeschränkt ≠ unbeschrankt
- Take care of **synonyms,** e.g. auto → car
  - Again, be **very careful** when doing this!

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

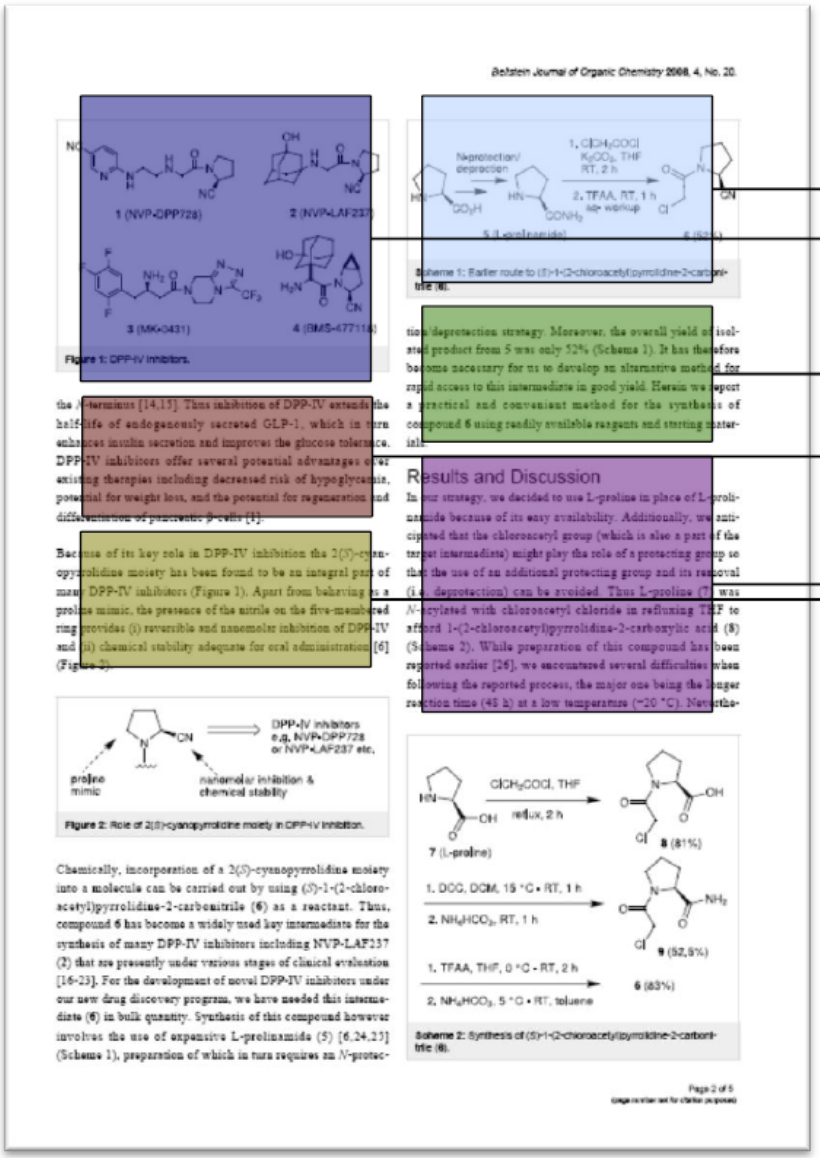
Document Representation



- Finally, we arrive at the **bag-of-words representation**
- The preparation of original documents is finished now, but we need to talk about **efficient data structures for managing the inverted indexes…**

Segmentation

Detour



Beilstein Journal of Organic Chemistry 2008, 4, No. 20.

Figure 1: DPP-IV inhibitors.

Figure 2: Role of 2(S)-cyanopyrrolidine moiety in DPP-IV inhibition.

Scheme 1: Earlier route to (S)-1-(2-chloroacetyl)pyrrolidine-2-carbonitrile (6).

Results and Discussion

Scheme 2: Synthesis of (S)-1-(2-chloroacetyl)pyrrolidine-2-carbonitrile (6).

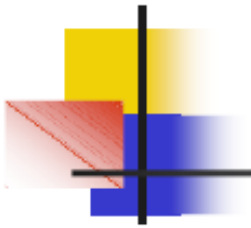
Needed in Google Scholar or CiteSeer

Textual output

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

Inverted Index

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html



The Inverted Index Data Structure

- An Inverted Index consists of 2 elements:
  - The lexicon (AKA vocabulary, dictionary)
  - The inverted file (AKA postings file)
- The Lexicon is the set of all indexing units (terms) in a given collection
- The Inverted file is a set of *postings lists* – a list per term. The list consists of *posting elements.*
  - The list of term t holds the locations (documents + offsets therein) where t appeared.
  - Additional data (*payload*) is often encoded per location
  - Many variations and degrees of freedom
- Supports efficient lookup from word to occurrences
- Essentially a way of representing and accessing (by row) a huge and very sparse matrix

Offset = location in document for phrases

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

Inverted Index Example

The good
1 2
the bad
3 4
and the ugly
5 6 7

Doc #1

As good as it gets, and more
1 2 3 4 5 6 7

Doc #2

Is it ugly and bad? It is, and more!
1 2 3 4 5 6 7 8 9

Doc #3

Lexicon

Term	DF	
The	1	(1; 1,3,6)
Good	2	(1; 2) – (2; 2)
Bad	2	(1; 4) – (3; 5)
And	3	(1; 5) – (2; 6) – (3; 4,8)
Ugly	2	(1; 7) – (3; 3)
As	1	(2; 1,3)
It	2	(2; 4) – (3; 2,6)
Gets	1	(2; 5)
More	2	(2, 7) – (3, 9)
Is	1	(3; 1,7)

Basic questions:

- How to efficiently build an inverted index?
- How does the inverted index efficiently support the various retrieval models (Boolean, Vector Space, etc.)?
- How does the inverted index handle complex operators, e.g. phrase queries?

Recap: Inverted Indexes

- In **Boolean retrieval,** queries have been evaluated using **inverted indexes** (aka inverted files)
- Document collection:
  - Document1 = {step, mankind}
  - Document2 = {step, China}
- Inverted index:
  - step: {Document1, Document2}
  - mankind: {Document1}
  - China: {Document2}
- Query:
  - "mankind AND step"

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

Indexing

Indexing:
"The process of assigning keywords to each document"

- These **keywords** are used as a (possibly intermediate) **representation** of each document
- **Some problems** we are faced with:
  - "3/12/91" vs. "Mar 12, 1991" vs. "12/3/1991"
  - "<h1>Document heading</h1>" vs. "Document heading"
  - computer vs. computers vs. Computer vs. computer's
  - aren't vs. are not

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

Index Construction

Index Construction

- **Building an inverted index looks easy:**
  1. Assign an ID to each document: **docID**
  2. Run the **document preparation** process on each document
  3. Compile a list of all **index terms**
  4. Assign an ID to each index term: **termID**
  5. Create a list of all **(termID, docID, tf) triplets**
  6. **Sort** this list: Primarily by termID, secondarily by docID
- Essentially, this corresponds to a **matrix transposition**
- Let's have a look at an example…

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

Example

- Our **example collection** of six documents:
  1. The old night keeper keeps the keep in the town
  2. In the big old house in the big old gown
  3. The house in the town had the big old keep
  4. Where the old night keeper never did sleep
  5. The night keeper keeps the keep in the night
  6. And keeps in the dark and sleeps in the light
- Case-folding, stopping, and stemming reduces the vocabulary to **ten index terms:**

1. big	4. house	7. night	10. town
2. dark	5. keep	8. old	
3. gown	6. light	9. sleep	

Example (2)

tf	1	2	3	4	5	6	7	8	9	10
1					3		1	1		1
2	2		1	1				2		
3	1			1	1			1		1
4					1		1	1	1	
5					3		2			
6		1			1	1			1	

- Then, we get the following **(termID, docID, tf) triplets:**

(5, 1, 3), (7, 1, 1), (8, 1, 1), (10, 1, 1), (1, 2, 2), (3, 2, 1), (4, 2, 1), (8, 2, 2), (1, 3, 1), (4, 3, 1), (5, 3, 1), (8, 3, 1), (10, 3, 1), (5, 4, 1), (7, 4, 1), (8, 4, 1), (9, 4, 1), (5, 5, 3), (7, 5, 2), (2, 6, 1), (5, 6, 1), (6, 6, 1), (9, 6, 1)

Then sort by termID and then docID

Large Collections

- **Building the inverted index isn't difficult**
  if the whole document collection fits in **main memory**
- Now, let's get serious:
  Typical collections are very large…
- What can we do?
  - **Sort-based inversion:**
    Use a external (i.e. disk-based) sorting algorithm that works on compressed disk blocks (for performance reasons)
  - **Merge-based inversion:**
    Read and index documents in memory until a fixed capacity is exceeded; when memory is full, the index is flushed to disk and merged with the index already stored on disk

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

Index Representations

- The problem of building the index essentially is solved
  - OK, Google uses massive replication and data distribution but these are very special requirements…
- Now, how to store an inverted index on disk?
- Since disk accesses are very expensive (e.g. compared to computations), there are **two major requirements:**

> Keep the index as small as possible!
>
> Read as little data as possible from disk!

- Since computational power comes at (almost) no cost, **effective data compression** is our first way to go!

Dynamic Indexing

- How to handle changes to document collection?
  - New documents
  - Updated documents
  - Deleted documents
- **Simple (but inefficient) solution:**
  Rebuild the index from scratch
- **Better solution:**
  Keep an auxiliary in-memory index that keeps track of all changes
  - If the auxiliary index gets too large, it is merged with the main index

Query Structure and Processing

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

Query Result Caching

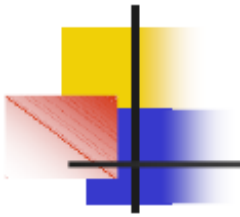
Exploits locality of reference in the query streams that are submitted to search engines:

- Query popularities vary widely, from the extremely popular to the very rare
- Although a minority, significant number of users still view multiple *result pages* per query (motivates prefetching)

Successful caching and prefetching of results can:

- Lower the number/cost of query executions
- Shorten the engine's response time and lower its hardware requirements

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

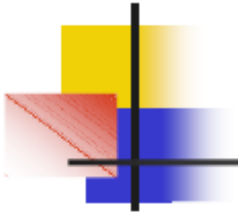


Query Log Analysis

Analysis of 7160190 queries submitted to AltaVista during September 2001:

- The 7160190 queries requested 4496503 distinct result pages, belonging to 2657410 distinct query strings
- 67% of the 2657410 strings were only requested once; the most popular string was queried 31546 times
- 48% of the result pages were only requested once; the 50 most requested pages account for almost 2% of all requests
- Qualitative behavior consistent with analyses of query logs of other engines

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html



Result Caching and Prefetching

- Simple LRU-based caches of query results can achieve cache hit-rates of about 30% (Markatos, 2000)
- Prefetching deeper result pages, along with more complex caching policies, can achieve hit rates of over 50%
- What is the cost associated with prefetching additional results?
- What happens when the index changes?
- BTW, what about caching of other data in search engines, e.g. postings lists?

Least Recently Used (LRU): discards the least recently used items first

Query Processing

- **Boolean retrieval:**
  Process queries as we already have discussed it

- **Vector space retrieval:**
  Answer the query "dark, keep, night" by scanning through the postings lists for "dark," "night," and "keep," while accumulating scores for each document
  - Let's have a look at an example…

Vector Space Retrieval

- **Query = "dark keep night"**
- Vector representation: Simple term frequencies
  - Query vector = (0, 1, 0, 0, 1, 0, 1, 0, 0, 0)
- Similarity measure:
  Simple scalar product

Term	Posting List
2: dark	(6, 1)
5: keep	(1, 3), (3, 1), (4, 1), (5, 3), (6, 1)
7: night	(1, 1), (4, 1), (5, 2)

- Process query by scanning though the three lists and add up term frequencies for each occurring document
- This gives the following final scores:

Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6
3 + 1 = 4	0	1	1 + 1 = 2	3 + 2 = 5	1 + 1 = 2

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

Phrase Queries

- A special type of queries are **phrase queries**
- **Example:** "King of Finland"

- **Three strategies** to process phrase queries:
  - **Postprocessing:**
    Initially, ignore the word order and do Boolean retrieval;
    In a second step, search through the documents found and return only the ones containing the phrase
  - **Store word positions:**
    Add word positions to each posting so that the locations of terms in documents can be checked during query evaluation
  - **Partial phrase indexes:**
    Create a (partial) index containing phrases

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

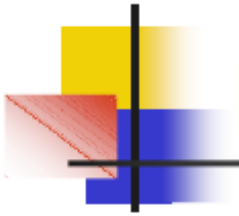
Link Structure Analysis including PageRank

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

The Information Need Behind the Query [Broder, SIGIR Forum 36(2), 2002]

- Informational — **Find Topics**
  - I want to learn more about the "roman empire"
  - Many possible fine results
- Navigational
  - Find me the home page of "el al"
  - In principle, a single correct result
  - In practice, correct result may depend on language/locale
- Transactional
  - I want to buy a "digital camera"
  - Good results: e-tailors that sell digital cameras

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html



Navigational Queries & Anchor Text

- Anchor text – the highlighted clickable text of a link
- Physically in page A, but actually describes the content of page B
- Many times, concisely defines page B
  - Often, better than the text present on page B itself
- Anchor text is the most important factor in treating navigational queries
- Case study: the query "engine" on Microsoft's Live Search
- Famous pop-culture examples: "French Military Victories", "miserable failure"

A screen capture showing what you find when you go to Google, type in "french military victories," and click the "I'm feeling lucky" button

See Also: Funniest Political Pictures of All Time

< — Previous Picture | Next Picture — >

BACK TO INDEX: Funny Political Pictures

More French Jokes

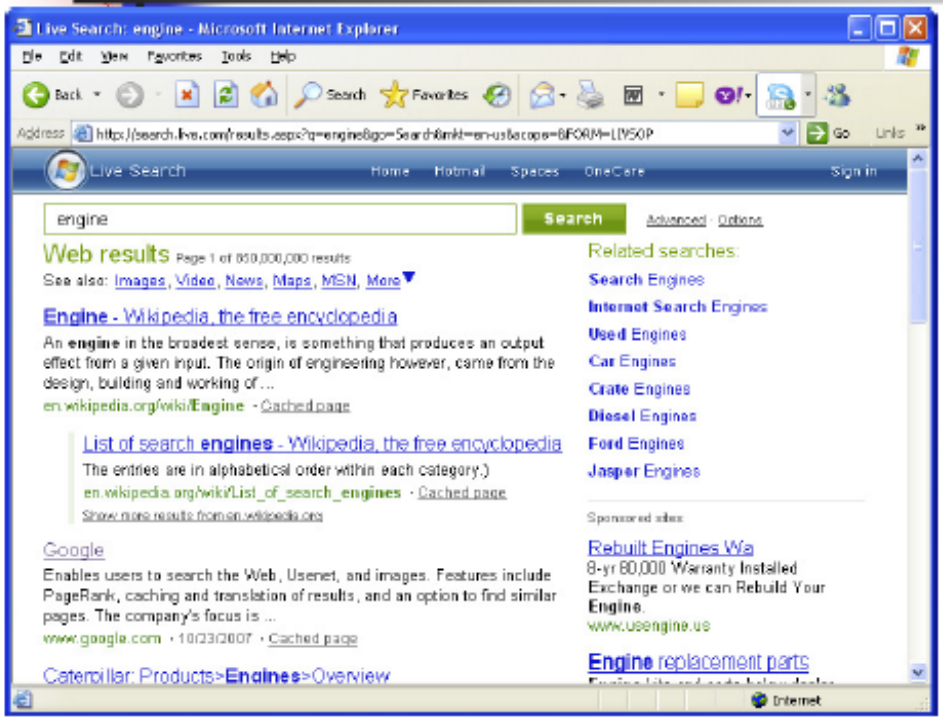
Email This Pic to a Friend | Subscribe to Newsletter

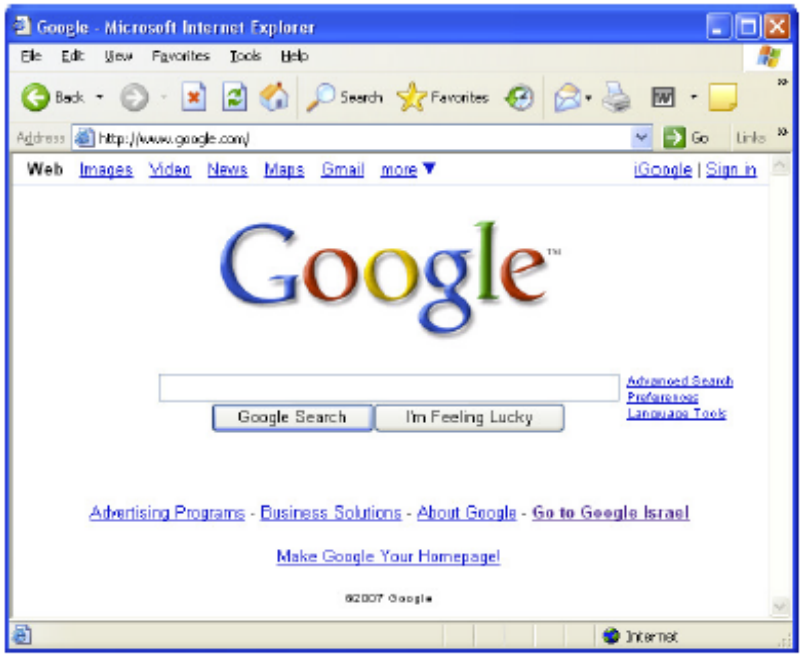
Related Articles

- CustomizeGoogle Firefox Extension Review - Review of CustomizeGoogle Firefo...
- Getting a Big Company to Do the Right Thing - Reader Stories: Entrepreneurs...
- Why Hyperlink Names Matter to Google Search Results

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

Case study: the query "engine" on Microsoft's Live Search



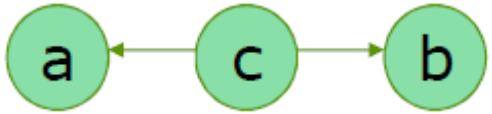


The word "engine" does not appear
Anywhere on the page, nor in its source…

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

Link-Structure Analysis

- Beyond anchor-text: the connectivity patterns between Web pages contain a gold mine of information
- A link from page a to page b can often be interpreted as:
  1. A recommendation, by a's author, of the contents of b.
  2. Evidence that pages a,b share some topic of interest.
- A co-citation of a and b (by a third page c) may also constitute evidence that a,b share some topic of interest

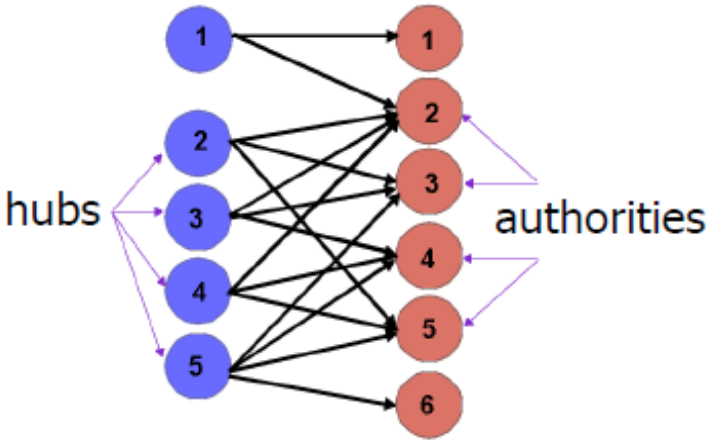


http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

Hubs and Authorities

- Notions proposed by Jon Kleinberg (1998)
- Two types of quality Web pages that pertain to a given topic

hubs: resource lists, containing links to many authorities



Authorities: pages containing authoritative content on the topic

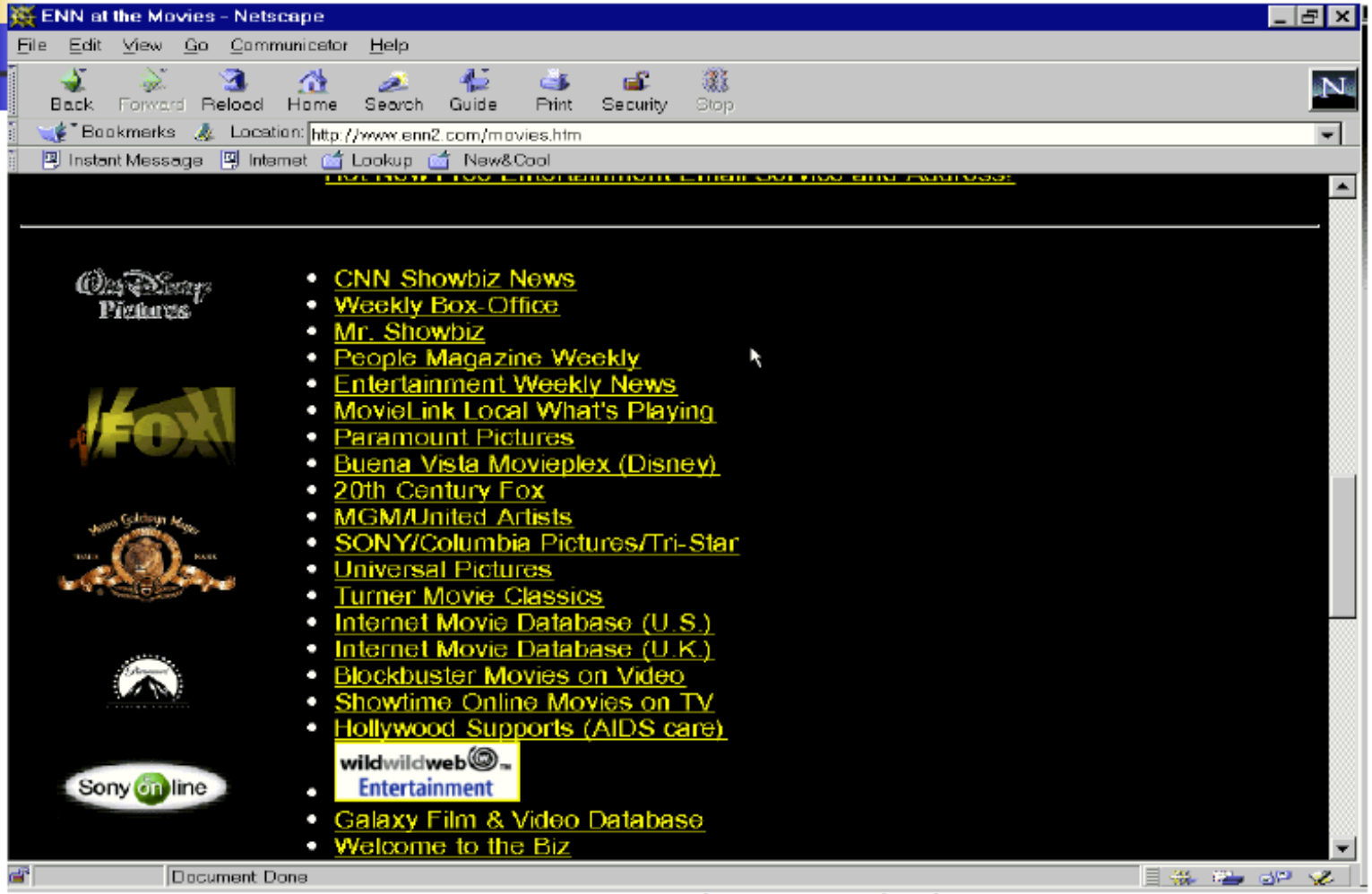
- HITS (Hypertext Induced Topic Search): an algorithm that identifies hubs and authorities, given a topic of interest t
- Based on the Perron-Frobenius theory of non-negative matrices

Example – an Authority for the Query "movies"

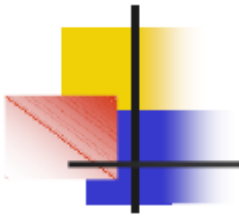
http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

Example – a Hub for the Query "movies"



http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

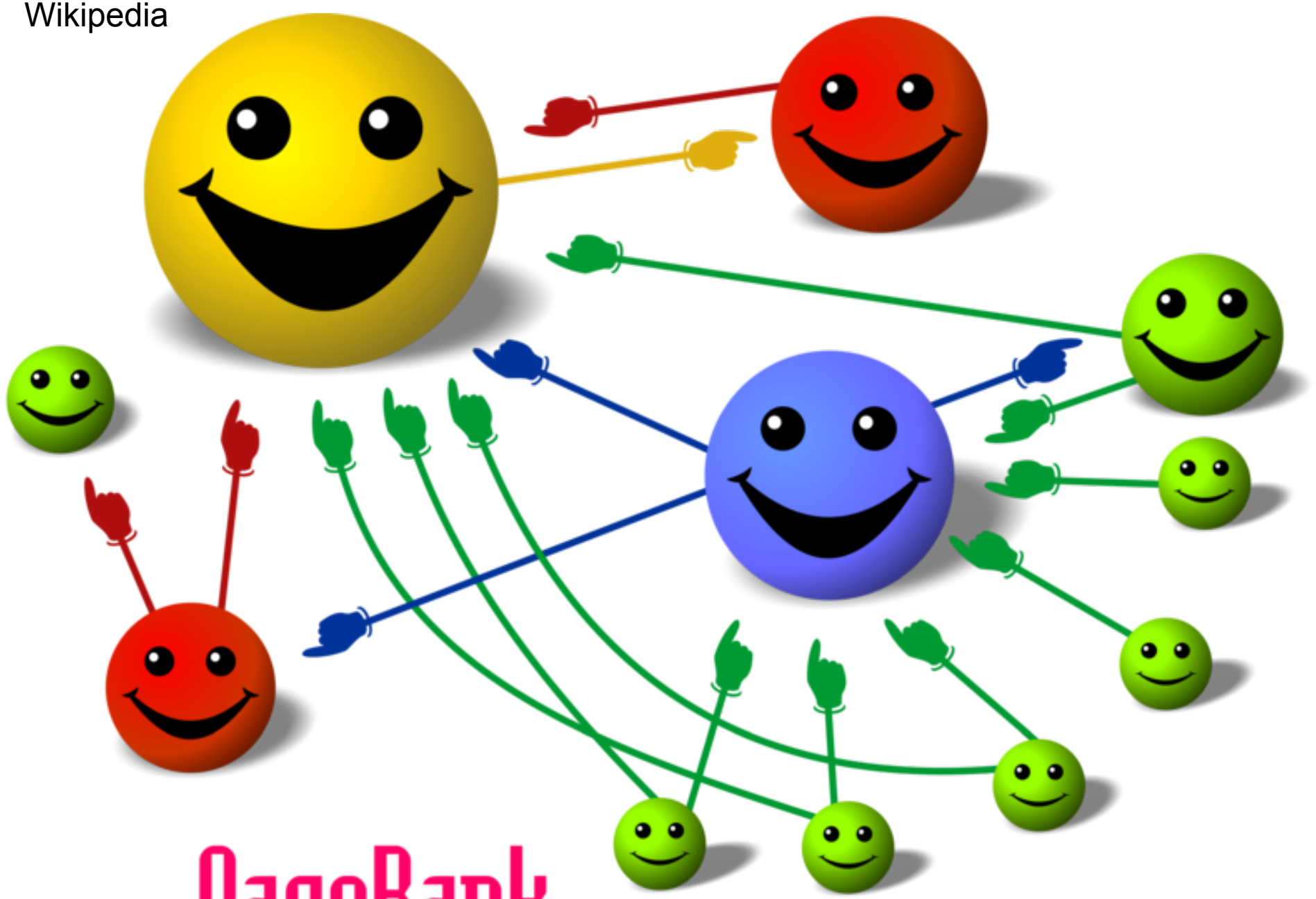


PageRank (Brin & Page, 1998)

- Named after Google's co-founder, Larry Page
- A global, query independent importance measure of Web pages
- A page is considered important if it receives many links from *important* pages
- Based on Markov chains and random walks

Markov Chains are roughly random walks
i.e. Sequence of random browsing selected either from current page (0.85) or from world (0.15)

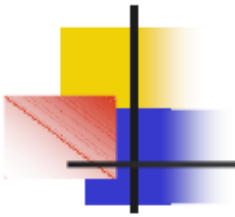
Wikipedia



PageRank

Size of face proportional to PageRank

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html



PageRank: "Random Surfer" Model

A random surfer moves from page to page.

Upon leaving page *p*, the surfer chooses one of two actions:

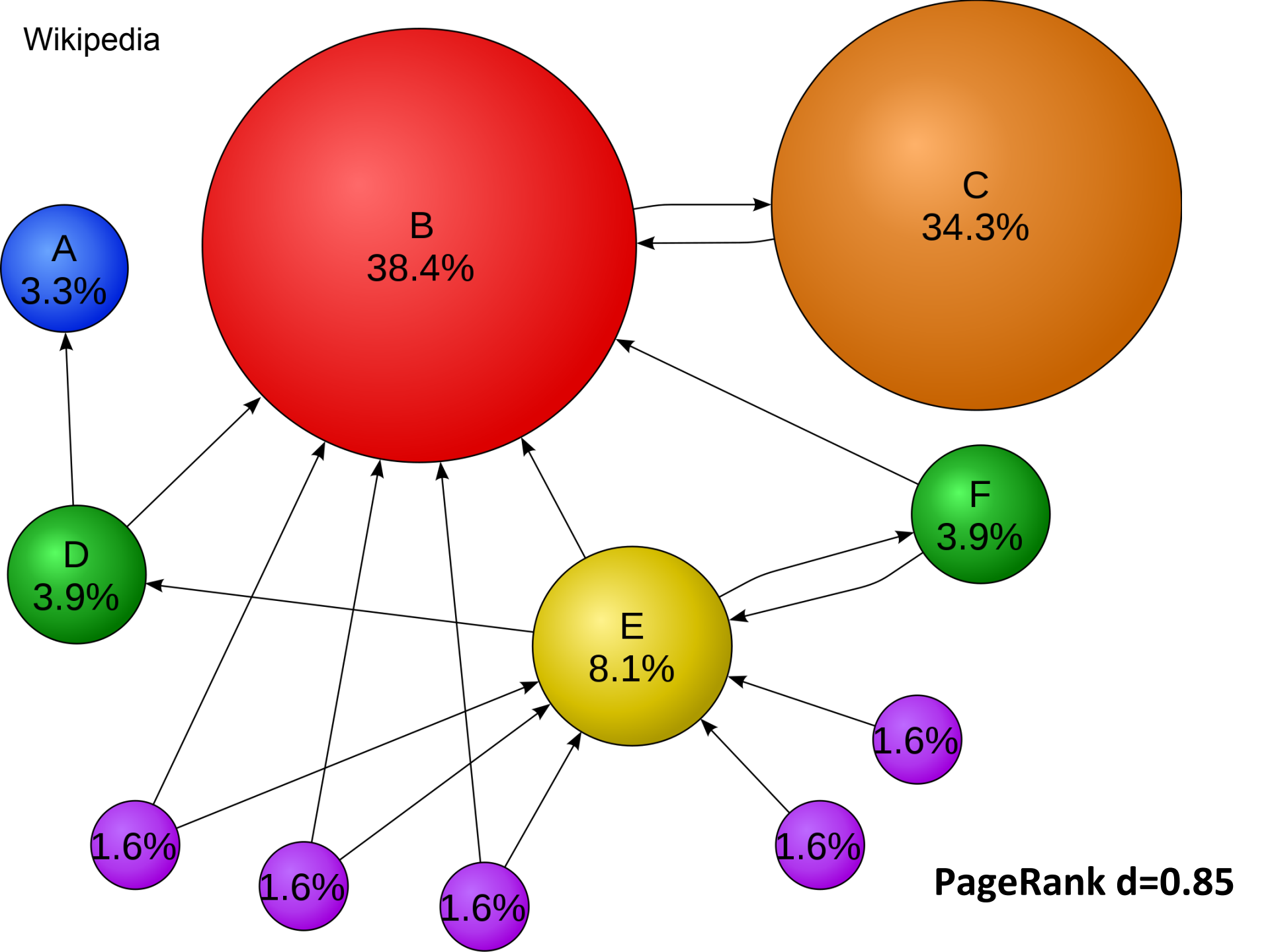
1. Follows an outgoing link of *p* (chosen uniformly at random), w.p. d
2. Jumps to an arbitrary Web page (chosen uniformly at random), w.p. 1-d

The vector of PageRanks is the stationary distribution of this (ergodic) random walk

d = 0.85

PageRank

- PageRank is probability that Page will be visited by a surfer is clicks each link on page with equal probability
  - minor corrections for pages with no outgoing links
- Found Iteratively with each page getting at each iteration a contribution equal to its page rank divided by #Links on page
- PR(Page *i*) = $\Sigma_{\text{Page } j \text{ pointing at } i}$ PR(Page *j*)/(Number of Pages linked on Page *j*)
- One adds to this the chance 1-*d* that surfer types a random URL into web browser.
- That takes PageRank to *d* times above plus (1 - *d*) divided by total number of pages on web
- On general principles, this will converge whatever the starting point
  - It can be written as iterative matrix multiplication


Wikipedia
A
3.3%
B
38.4%
C
34.3%
D
3.9%
E
8.1%
F
3.9%
1.6%
1.6%
1.6%
1.6%
1.6%
PageRank d=0.85

Related Applications

- Thinking of Page Rank as reputation
- A version of PageRank has recently been proposed as a replacement for the traditional Institute for Scientific Information (ISI) impact factor, and implemented at eigenfactor.org. Instead of merely counting total citation to a journal, the "importance" of each citation is determined in a PageRank fashion.
  - Impact Factor is number of citations of each article
  - The Eigenfactor score of a journal is an estimate of the percentage of time that library users spend with that journal. The Eigenfactor algorithm corresponds to a simple model of research in which readers follow chains of citations as they move from journal to journal.
- A similar new use of PageRank is to rank academic doctoral programs based on their records of placing their graduates in faculty positions. In PageRank terms, academic departments link to each other by hiring their faculty from each other (and from themselves).

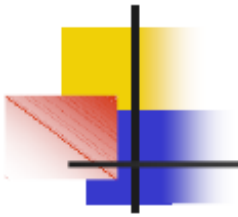
Order	Journal	Percentile	EF ↓	AI ↓
1	NATURE ISSN: 0028-0836	EF: 100 AI: 100	1.65524	20.373
2	P NATL ACAD SCI USA ISSN: 0027-8424	EF: 100 AI: 99	1.60168	4.8961
3	SCIENCE ISSN: 0036-8075	EF: 100 AI: 100	1.41162	17.5248
4	PHYS REV LETT ISSN: 0031-9007	EF: 100 AI: 98	1.14457	3.5175
5	J AM CHEM SOC ISSN: 0002-7863	EF: 100 AI: 97	0.817303	2.7989
6	PHYS REV B ISSN: 1098-0121	EF: 100 AI: 89	0.756039	1.4281
7	J BIOL CHEM ISSN: 0021-9258	EF: 100 AI: 94	0.742127	2.0307
8	APPL PHYS LETT ISSN: 0003-6951	EF: 100 AI: 89	0.675752	1.3875
9	NEW ENGL J MED ISSN: 0028-4793	EF: 100 AI: 100	0.66383	21.304
10	CELL ISSN: 0092-8674	EF: 100 AI: 100	0.660816	20.554

EF= Eigenfactor
AI = Article Influence over the first five years after publication

Eigenfactor scores are scaled so that the sum of the *Eigenfactor* scores of all journals listed in Thomson's Journal Citation Reports (JCR) is **100**

Article Influence scores are normalized so that the mean article in the entire Thomson Journal Citation Reports database has an article influence of **1.00**

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

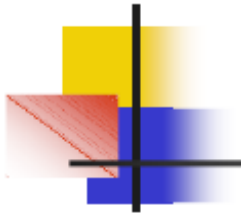


Link Analysis Algorithms

- Many variants and refinements of both HITS and PageRank have been proposed
  - Course will cover SALSA & Topic-Sensitive PageRank
- Other approaches include:
  - Max-flow techniques [Flake et al., SIGKDD 2000]
  - Machine learning and Bayesian techniques
- Examples of applications:
  - Ranking pages (topic specific/global importance)
  - Categorization, clustering, finding related pages
  - Identifying virtual communities
- A wealth of literature

None done here!

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html



Stability of Link Analysis

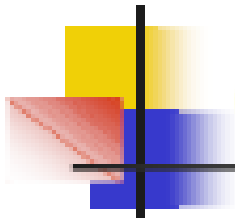
- IR algorithms should be robust – small changes in the input should not dramatically alter the results
- The Web changes constantly: pages and links are created, deleted and modified

? Do the known link-based algorithms produce stable scores/rankings?

? Does stability of scores imply stability of rankings?

Different research approaches have been applied to these questions, some problems are still unsolved

Summary Issues



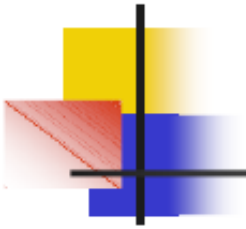
Comparing the Models

- Differ primarily in their theoretical basis, and in how "relevance" is defined and measured
  - Boolean model assumes relevance is related to term occurrence
  - Vector space model assumes relevance is related to similarity
  - Probabilistic model relies on probability estimates
- The Boolean model does not provide for varied relevance scores and is considered to be the weakest classic model

model

- In theory, the probabilistic should supply the best predictions of relevance given enough information
- Experimentally, the vector space model often outperforms the probabilistic model on general collections

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

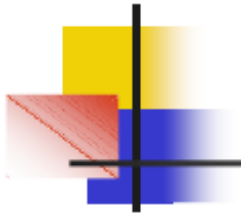


Classic and Modern IR Issues

- Polysemy (ambiguity): does "Jordan" refer to Michael Jordan, the Jordan River, or the Hashemite Kingdom of Jordan?
  - Can we ensure that our SERP* will represent all aspects?
- Synonymy: elevator/lift, movie/film
- Do the models hold when queries are very short?
- Short queries require the addition of proximity considerations (query term "hits" should be in proximity to each other) – how does this fit with the models?
- Information Retrieval merits a course of its own!

* SERP – Search Results Page

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html



Size & Dynamics of the Web

- Billions of pages, tens of millions of sites
- Hundreds of terabytes (10^{14}) of data
- Projected growth rate: doubling in size every two years (some estimate rate is much quicker)
- New pages are added; existing pages are modified and deleted
- Unstructured information in chaotic disorder
- Many media types
- Widely distributed, both geographically and organizationally

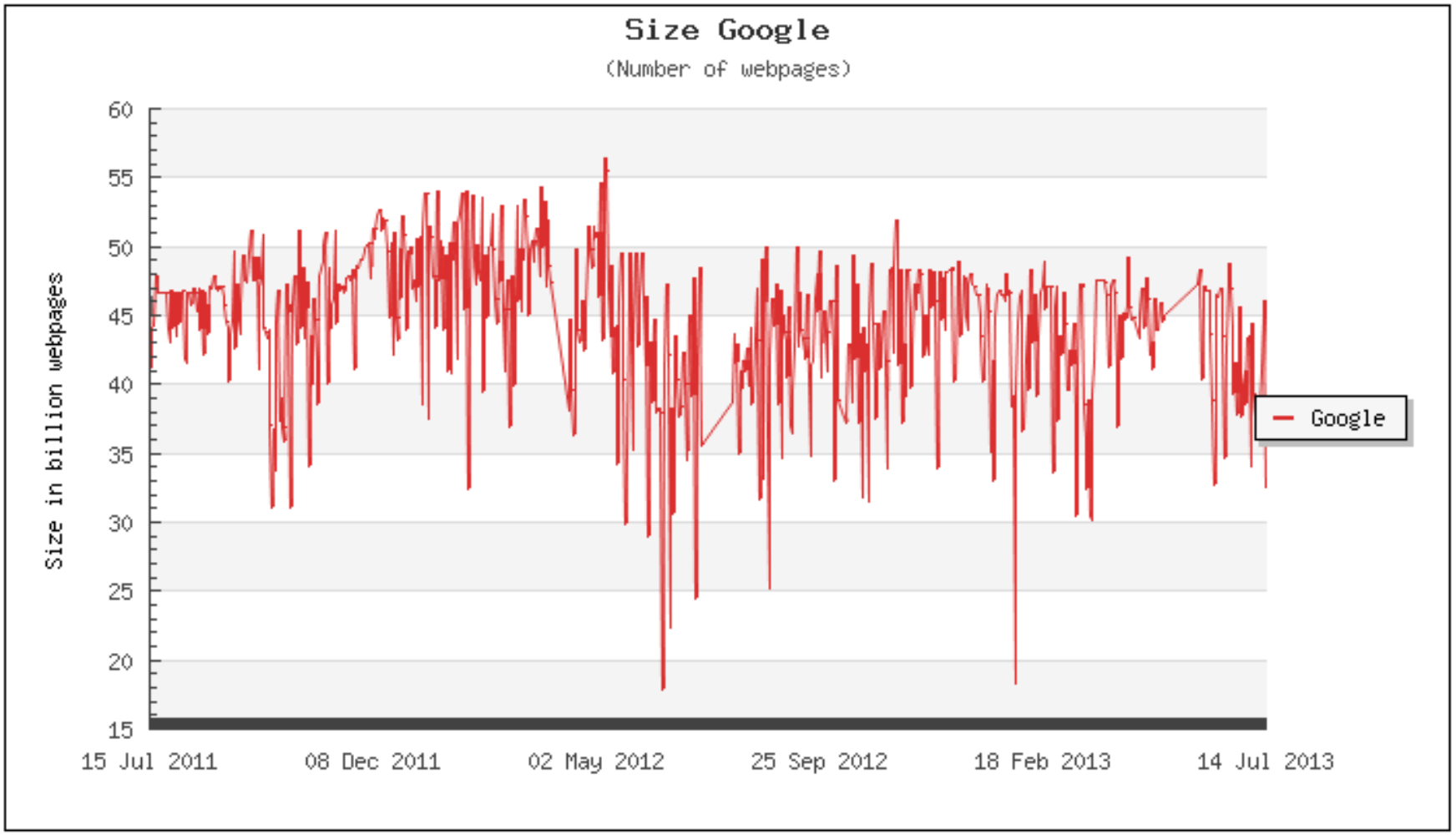
http://www.worldwidewebsize.com/

Estimated from number of pages returned to simple queries and fraction of documents satisfying queries

Google

The size of the World Wide Web: Estimated size of Google's index

Last Month | Last Three Months | Last Year | Last Two Years

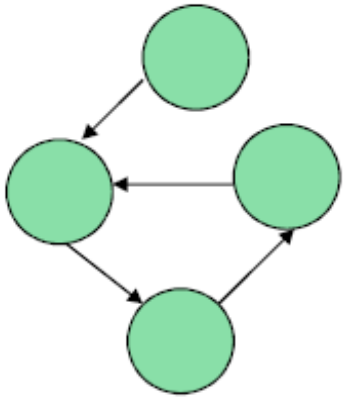


The Web as a Graph

See start of previous lesson

Pages as graph nodes, hyperlinks as edges

- Sometimes sites are taken as the nodes

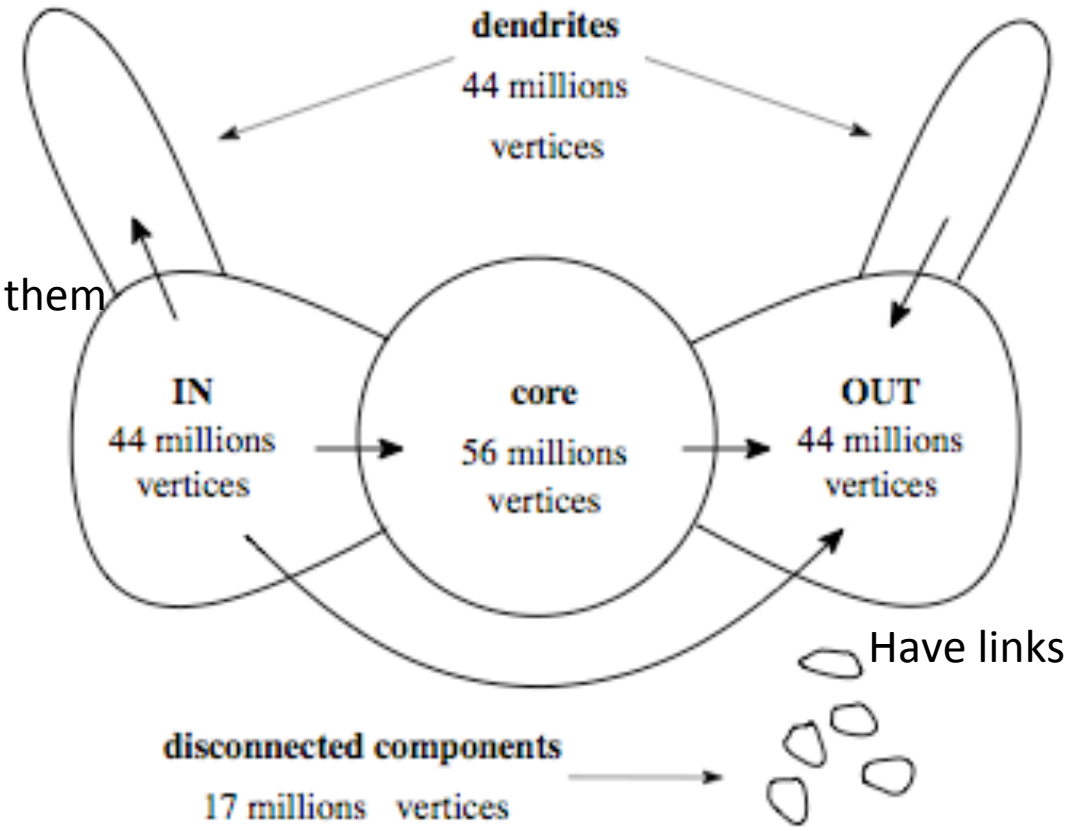


Some natural questions:

1. Distribution of the number of in-links to a page
2. Distribution of the number of out-links from a page
3. Distribution of the number of pages in a site
4. Connectivity: is it possible to reach most pages from most pages?
5. Are there models that explain how the Web graph evolved to its current shape?

Jean-Loup Guillaume and Matthieu Latapy

Have links going from them



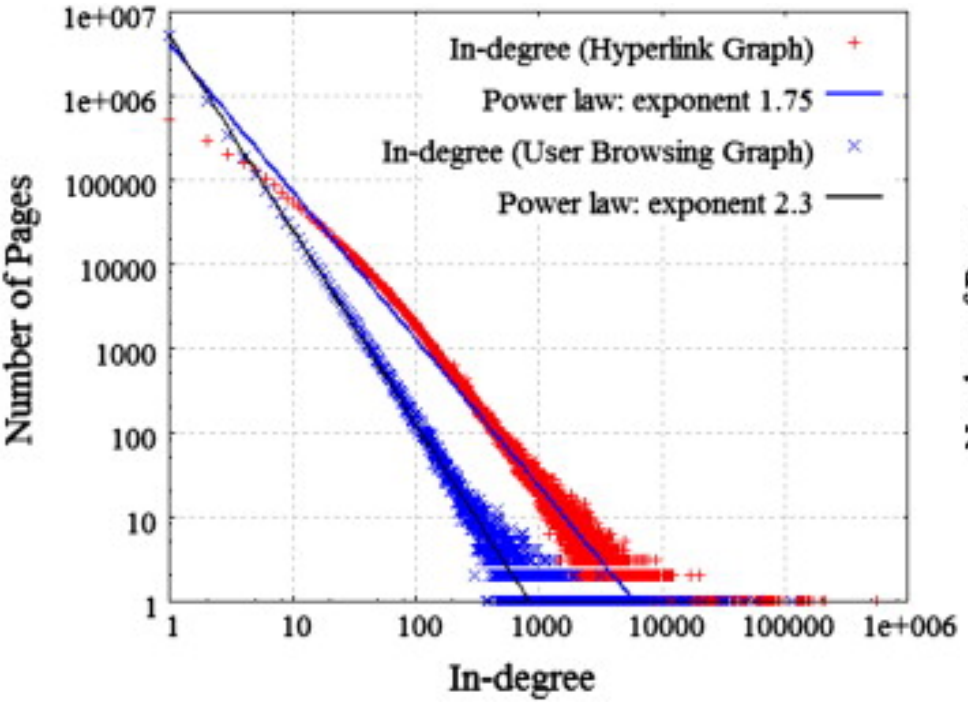
Have links going into them

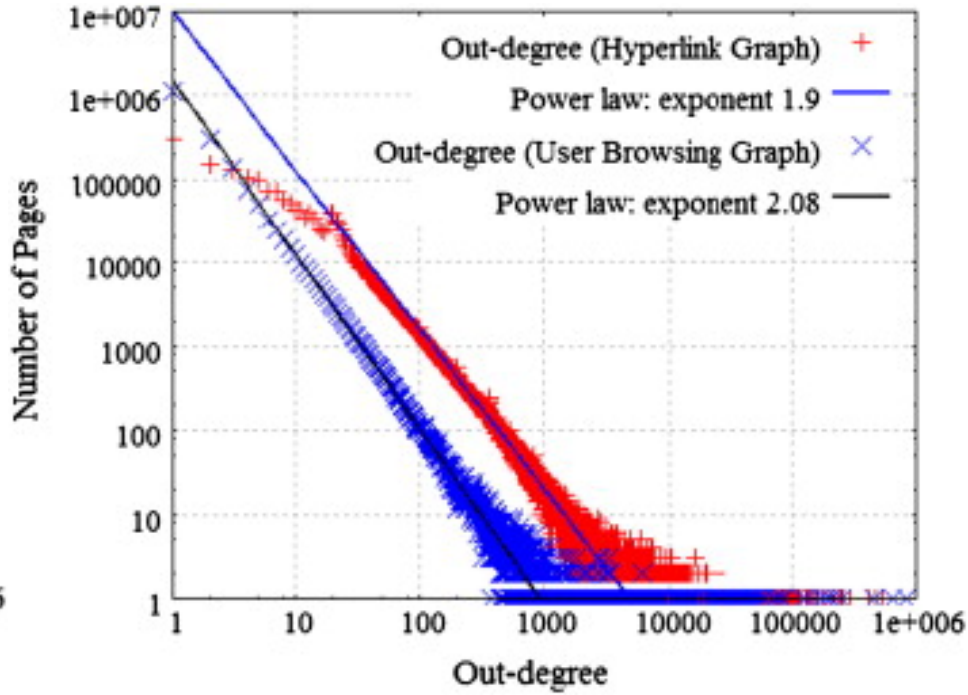
Fig. 3: *The bow-tie macroscopic structure of the Web graph [BKM+00]: the core, the IN component, the OUT component and the dendrites. Each of these parts contains around one quarter of the pages, the disconnected part being reduced to less than 10% of the whole.*

http://hal.archives-ouvertes.fr/docs/00/05/44/58/PDF/webgraph.pdf

Universal Power Laws for Links

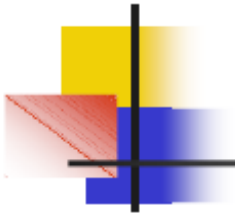
http://www.sciencedirect.com/science/article/pii/S0167923612001844





Crawling the Web

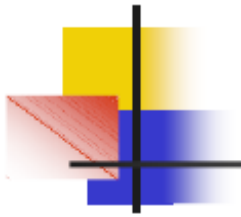
http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html



Crawlers - Framework

- The World Wide Web can be viewed as a directed graph
- Nodes are web objects, such as html pages, image files, pdf files, forms, etc...
- Nodes are uniquely identified by URLs
- Edges are html hyperlinks, pointing from an html page to its embedded objects or to other web objects
- Edges are uniquely identified by their source and target nodes

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html



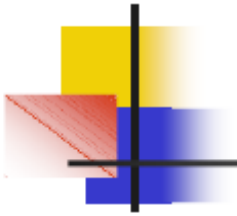
Generic Web Crawling Algorithm

Given a root set of distinct URLs:

- Put the root URLs in a queue
- While queue is not empty:
  - Take the first URL x out of the queue
  - Retrieve the contents of x from the web
  - Do whatever you want with it...
  - Mark x as visited
  - If x is an html page, parse it to find hyperlink URLs
  - For each hyperlink URL y within x:
    - If y is not marked visited, put y into queue

This is often the computational bottleneck

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html



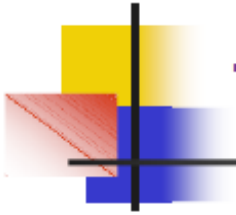
Crawler Design Issues

- Writing a simple crawler is a fairly easy task – one can be written in Java in a few lines of code
- However, there are many issues that need to be considered when writing a robust Web-scale crawler:

- Redirections
- Encrypted and password protected data
- Politeness - avoiding server load and respecting Robot Exclusion Protocol
- Script handling
- Language and encoding issues
- The challenge of dynamic pages
- Duplicate pages
- Crawl prioritization and variable revisit rates to ensure freshness
- Focused crawling
- Concurrency and distribution issues
- And many many more...

Web Advertising and Search

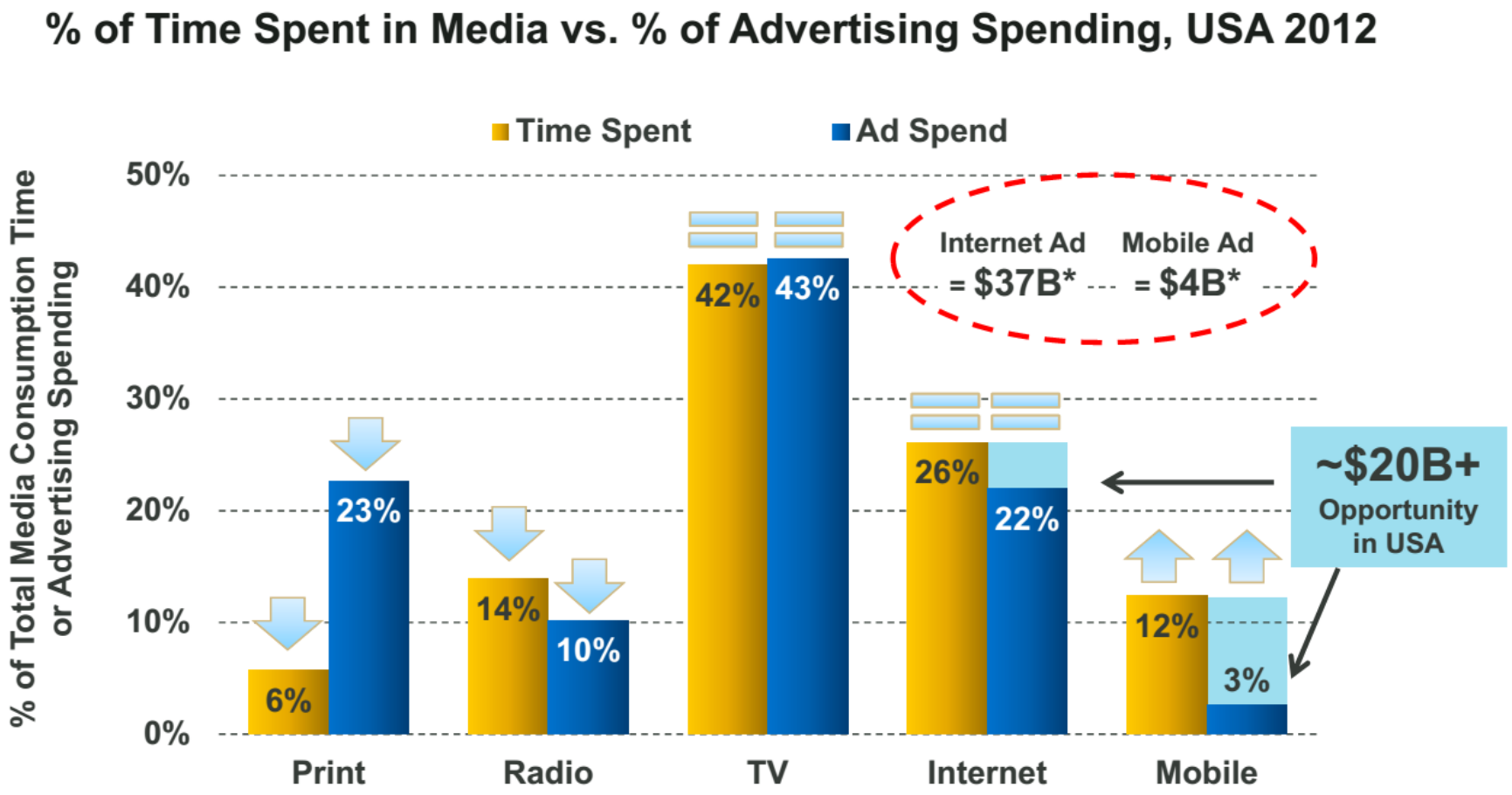
http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html



The Advertising Market

- Internet advertising is a 10^{10} business that is growing faster than old media advertising (radio, TV, newspapers & magazines, mail, outdoors)
- Online advertising budgets still lagging in proportion to time spent online, which is growing fast at the expense of old media
  - Seen as a driver to continued growth of online advertising market
- Traditional advertising:
  - Few, expensive opportunities
  - Targeting en-masse, by immediate context only
  - Difficult to measure effectiveness
- Internet advertising:
  - Billions of opportunities daily
  - Open to personalization via rich context of impression
  - Effectiveness is measurable: can measure click-through rates as % of impressions, and conversions as % of clicks

Material Upside for Mobile Ad Spend vs. Mobile Usage



Meeker/Wu May 29 2013 Internet Trends D11 Conference

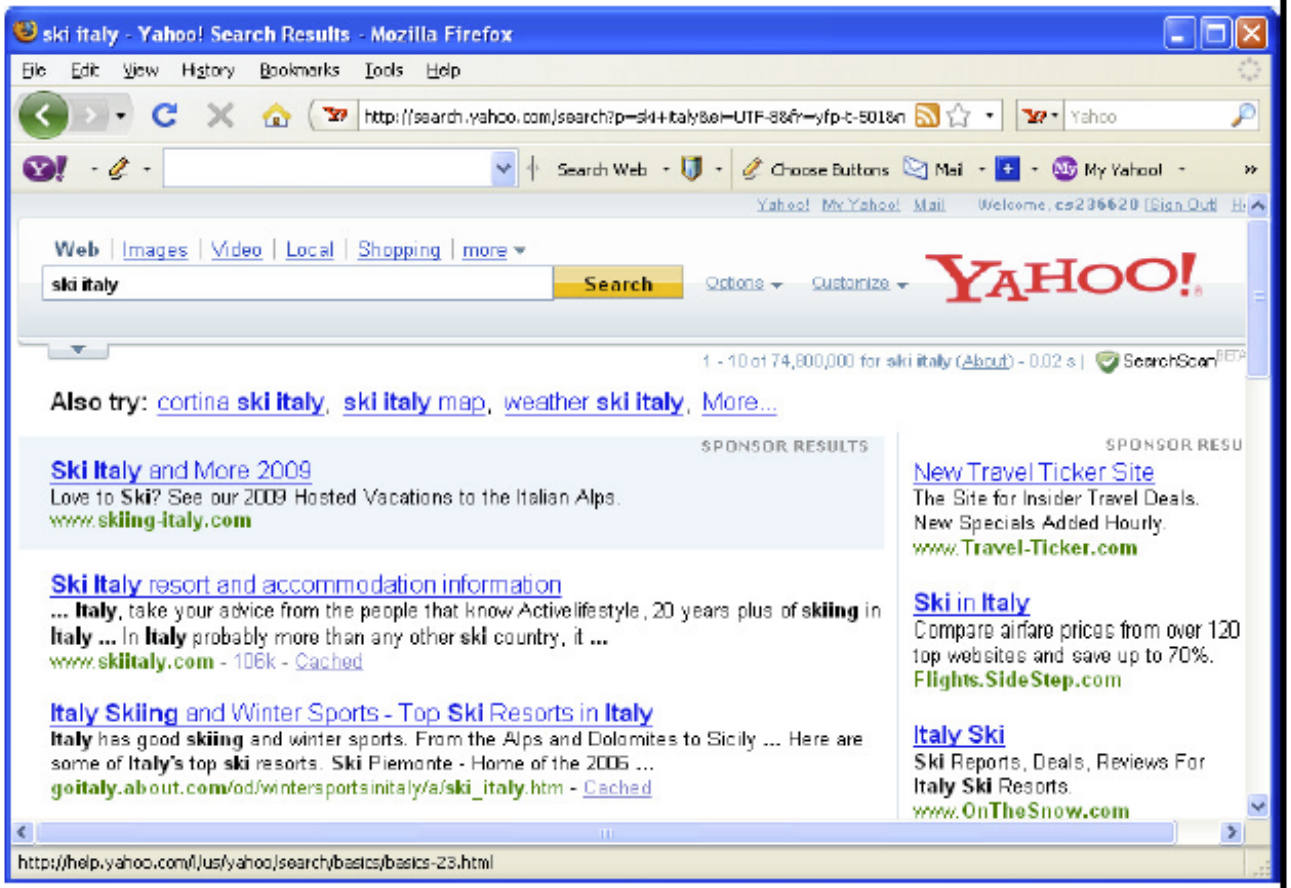
KPCB

Note: *Internet advertising reached $37B in USA in 2012 per IAB, Mobile advertising reached $4B per eMarketer. Print includes newspaper and magazine. $20B opportunity calculated assuming Internet and Mobile ad spend share equal their respective time spent share. Source: Time spent and ad spend share data based on eMarketer (adjusted to exclude outdoors / classified media spend), 12/12.

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

Internet Advertising Types: Sponsored Search

textual ads served on search results pages, triggered by query terms on which advertisers bid



http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

Internet Advertising Types: Contextual Ads (a.k.a. Content Match)

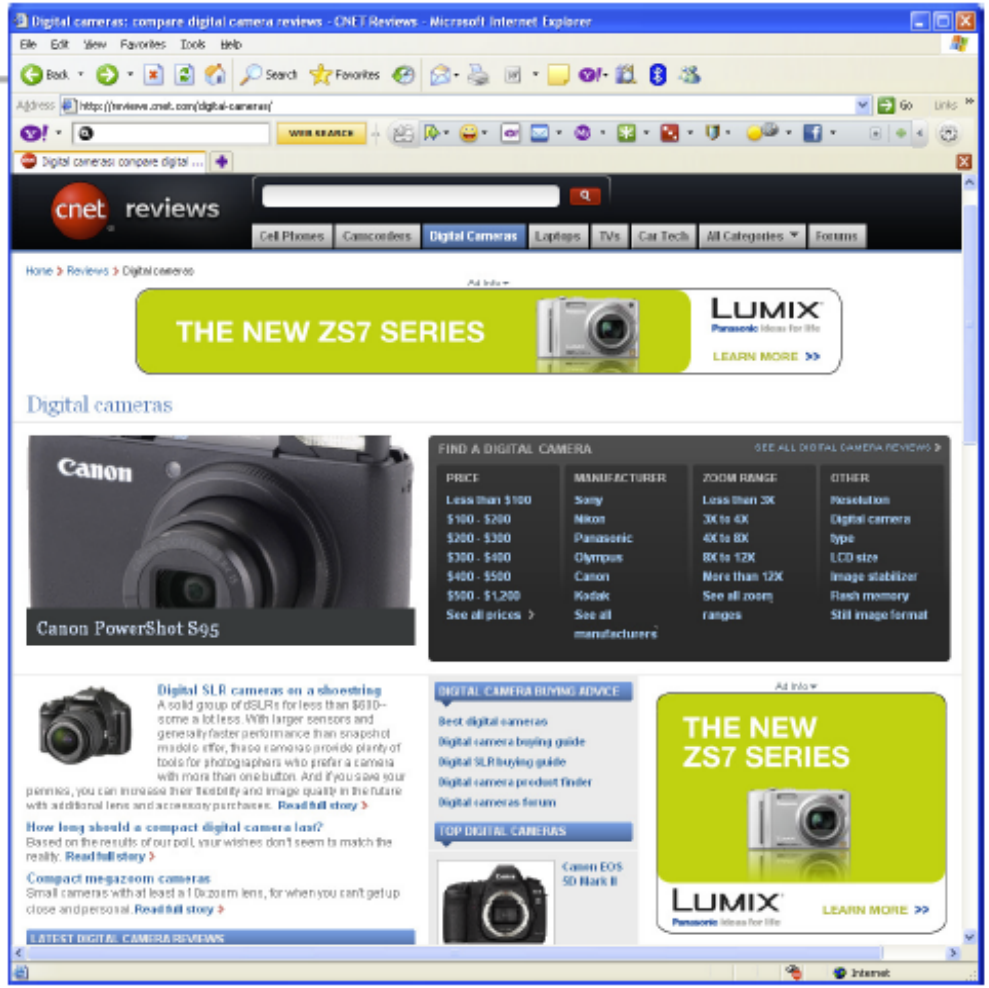
textual ads served on third party ("publisher") pages, triggered by content of page on which advertisers bid

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

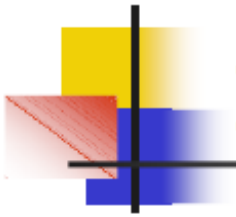
Internet Advertising Types: Display Ads

Graphical elements (e.g. banners) of standard sizes, served on publisher pages, mainly targeting audience demographics

- Guaranteed delivery (GD): advertiser buys a display campaign months in advance, requesting ## impressions to certain demographics in a given time period; ad agency pays fine for under-delivery
- Non-guaranteed delivery (NGD): advertiser bids in real time for impressions, with no guarantees that ads will be shown



http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

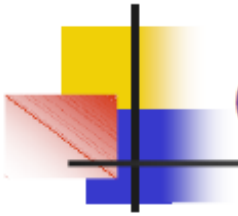


Internet Advertising Models

1. Cost per Impression (sometimes called Cost per *Mille*, i.e. 1000 impressions) – advertiser pays for ad to be shown
   - Main model for display advertising
2. Cost per Click – advertiser only pays when ad is clicked by user
   - Main model for sponsored search and content match
   - Sponsored search, 2005: 50% of clicks < $0.25, 80% < $0.50
3. Cost per Action – advertiser only pays when the user's interaction with the ad is followed by some transaction (not necessarily purchase)

http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

GD = Guaranteed Delivery
NGD = Non Guaranteed Delivery



Computational Advertising Challenges

- Conflicting goals:
  - Advertiser: campaign effectiveness – maximize ROI
  - Publisher: revenue without jeopardizing image of site
  - User: relevant ads, or at least ads that do not harm the experience
  - Ad agency (search engine): own revenue, while balancing goals of other parties; sometimes ad agency is also the publisher
- Ad agency challenges:
  - Which ads to trigger per query/page/impression? How to order them? What to charge for them?
  - In general, how to set the rules of the marketplace so as to ensure goals of parties are met (Game Theory/Mechanism Design)?
  - Which guaranteed delivery contracts to accept? How to balance GD with NGD?
  - Helping advertisers choose bid phrases effectively
  - Combating spam and fraud
  - Scale!!

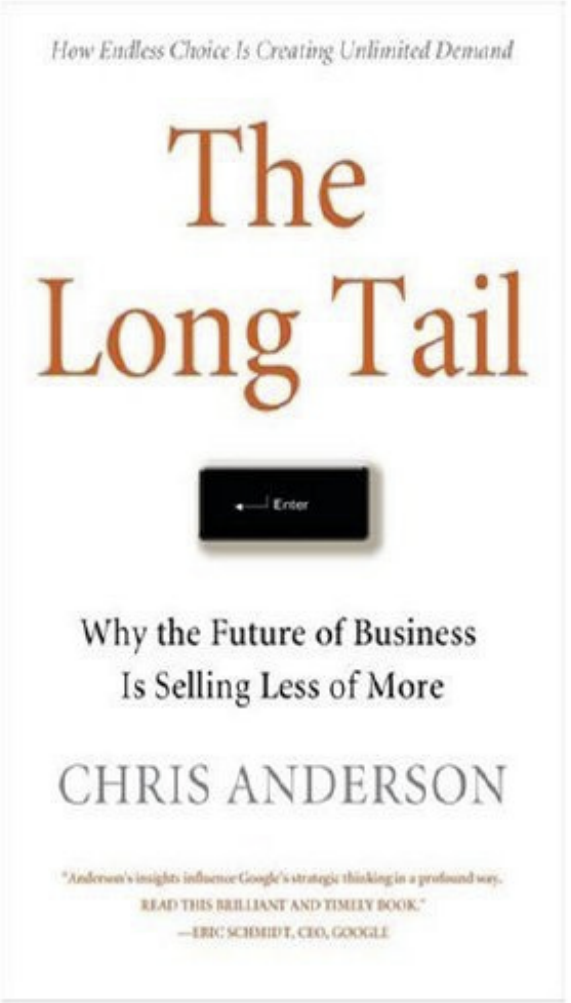
http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

The Long Tail

How the infinite choice available in e-commerce and the ease of publishing online have redefined the business of media & entertainment

Mottos:

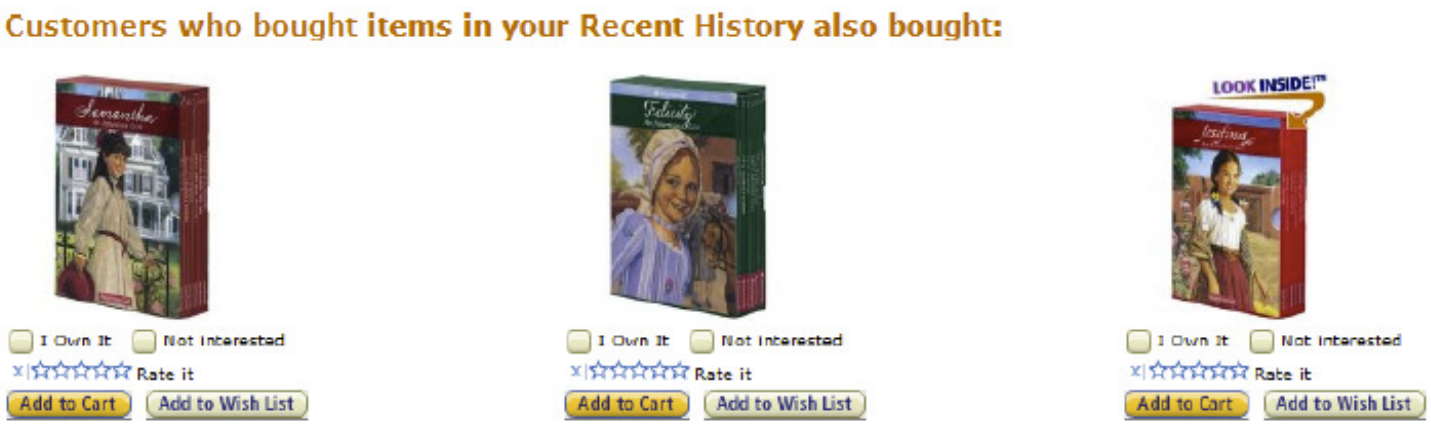
- "The biggest money is in the smallest sales"
- "Embrace niches"



http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html

Long Tail Economics and Recommender Systems

- People cannot intelligently navigate product or media catalogs consisting of millions of items
  - e-Inventories of books, songs, movies, albums are practically infinite
  - In particular, niche audiences cannot find niche content
- Recommender systems, collaborative filtering and social tagging services are key to Long Tail economics
  - Find an item you like, and we'll point you to similar items
- Pioneered by Amazon; today, used "everywhere"



Clustering and Topic Models

Grouping Documents Together

- The **responses to a search query** give you a group documents
- If we represent documents as points in a space, we can try to **identify regions**
  - **Clustering:** Nearby regions of points
  - **Support Vector Machine:** Chop space up into parts
  - (Gaussian) **Mixture Models**: A type of fuzzy clustering
  - **K-Nearest Neighbors** (if have examples)
- Alternatively we can determine "hidden meaning" with a **topic model** (defined by dynamic small bags of words)
  - Latent Semantic Indexing
  - Latent Dirichlet Allocation
  - With lots of variants of these methods to find "**latent factors**"

Topic Models

- Illustrated by Google News
- These try to group documents by Topics such as "Presidential Election" and not by inclusion of particular phrases
- You imagine each document is a set of topics (the latent factors) and each topic is a bag of words.
- Find the best set of topics and best set of words in topics

Yet Another Example

- Reuters-21578 collection
  - 21578 short newswire messages from 1987
- Top-3 results when querying for **taxes reagan** using LSI:

FITZWATER SAYS REAGAN STRONGLY AGAINST TAX HIKE
WASHINGTON, March 9 - White House spokesman Marlin Fitzwater said President Reagan's record in opposing tax hikes is "long and strong" and not about to change.

ROSTENKOWSKI SAYS WILL BACK U.S. TAX HIKE, BUT DOUBTS PASSAGE WITHOUT REAGAN SUPPORT

WHITE HOUSE SAYS IT OPPOSED TO TAX INCREASE AS UNNECESSARY

A Latent Factor Finding Method

- **The last document doesn't mention the term "reagan"!**

http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws

An example of DA-PLSI/PLSA

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
percent	stock	soviet	bush	percent
million	market	gorbachev	dukakis	computer
year	index	party	percent	aids
sales	million	i	i	year
billion	percent	president	jackson	new
new	stocks	union	campaign	drug
company	trading	gorbachevs	poll	virus
last	shares	government	president	futures
corp	new	new	new	people
share	exchange	news	israel	two

Top 10 popular words for each of 5 topics found from the AP news dataset divided into 30 topics. Processed by DA-PLSI and showing only 5 of 30 topics