Introduction to Search Engines

Andras Benczur

Insitute for Computer Science and Control
Hungarian Academy of Sciences

Overview of the three talks

- Search Engines
  - Architecture, Size of the Web
  - Web Bots, Indexing
  - Elements of Search Ranking, Learning to Rank
  - Web Spam
  - PageRank
- Distributed data processing systems
  - Hadoop – Word Count, Indexing
  - PageRank over Hadoop
  - Beyond Hadoop

About the presenter

András Benczúr
benczur@sztaki.hu

- Head of a large young team
- Research
  - Web (spam) classification
  - Hyperlink and social network analysis
  - Distributed software, Flink Streaming
- Collaboration- EU
  - NADINE – Dima et al.
  - European Data Science research – EIT Digital
    Berlin, Stockholm, Aalto, …
  - Future Internet Research
    with Internet Memory
- Collaboration- Hungary
  - Gravity, the recommender company
  - AEGON Hungary
  - Search engine for Telekom etc.
  - Ericsson mobile logs

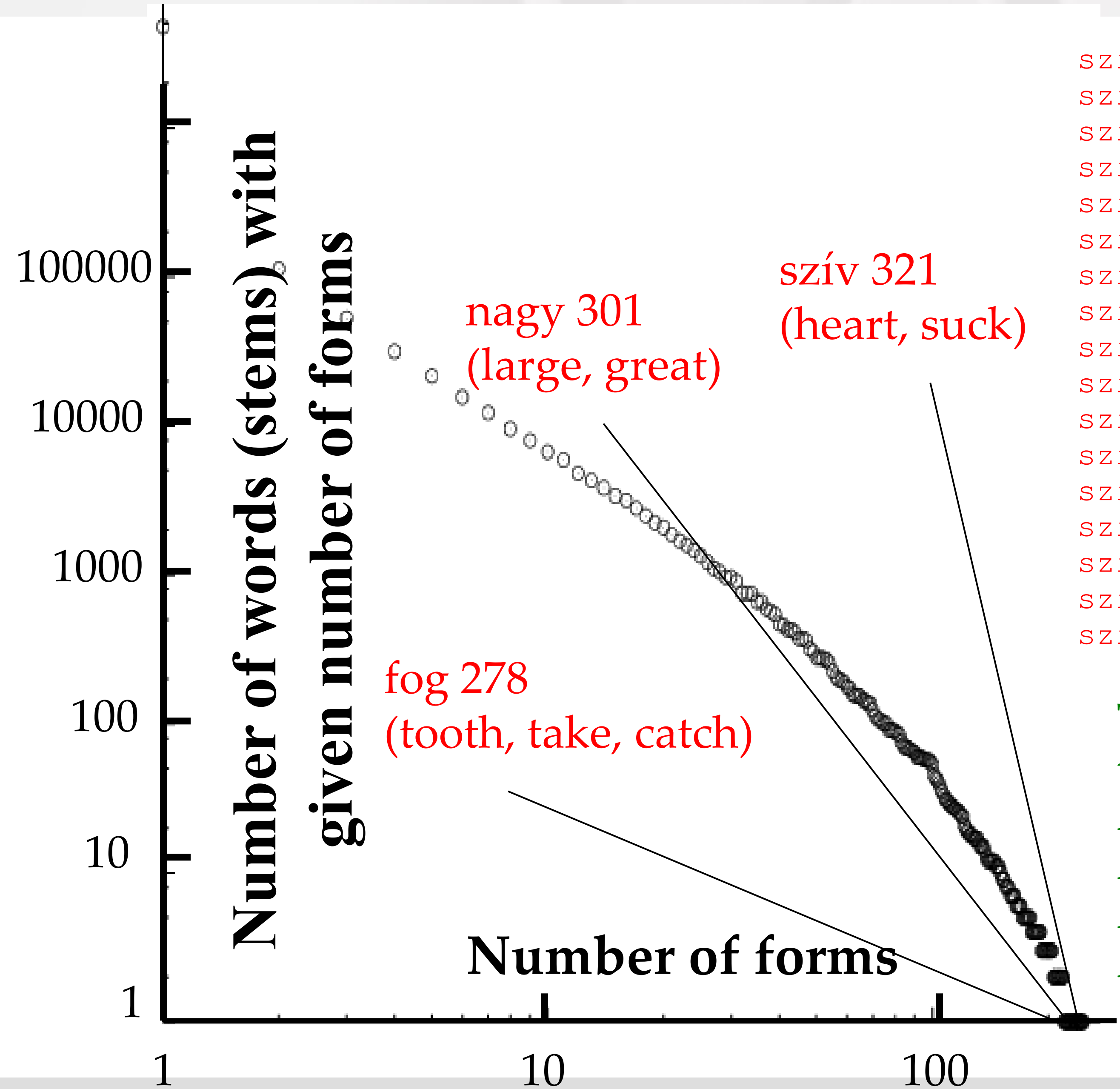
Search Engines

Architecture

Size of the Web

Crawling, Indexing, Ranking

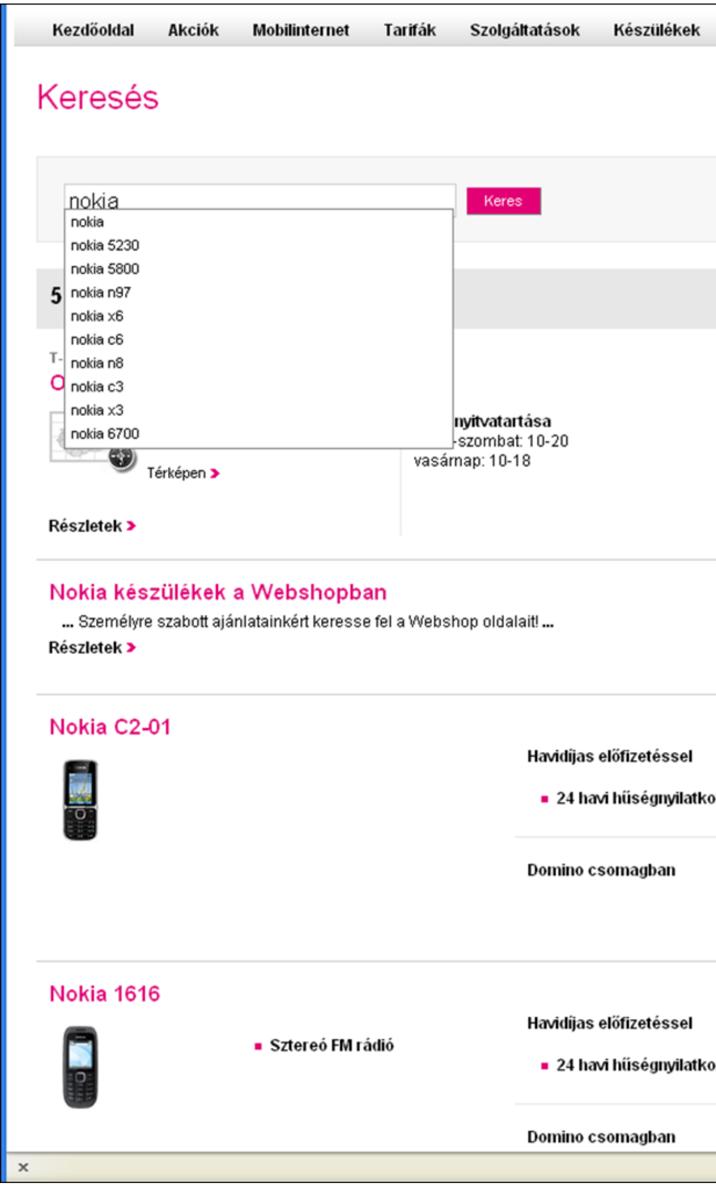
My start with search engines in 2002

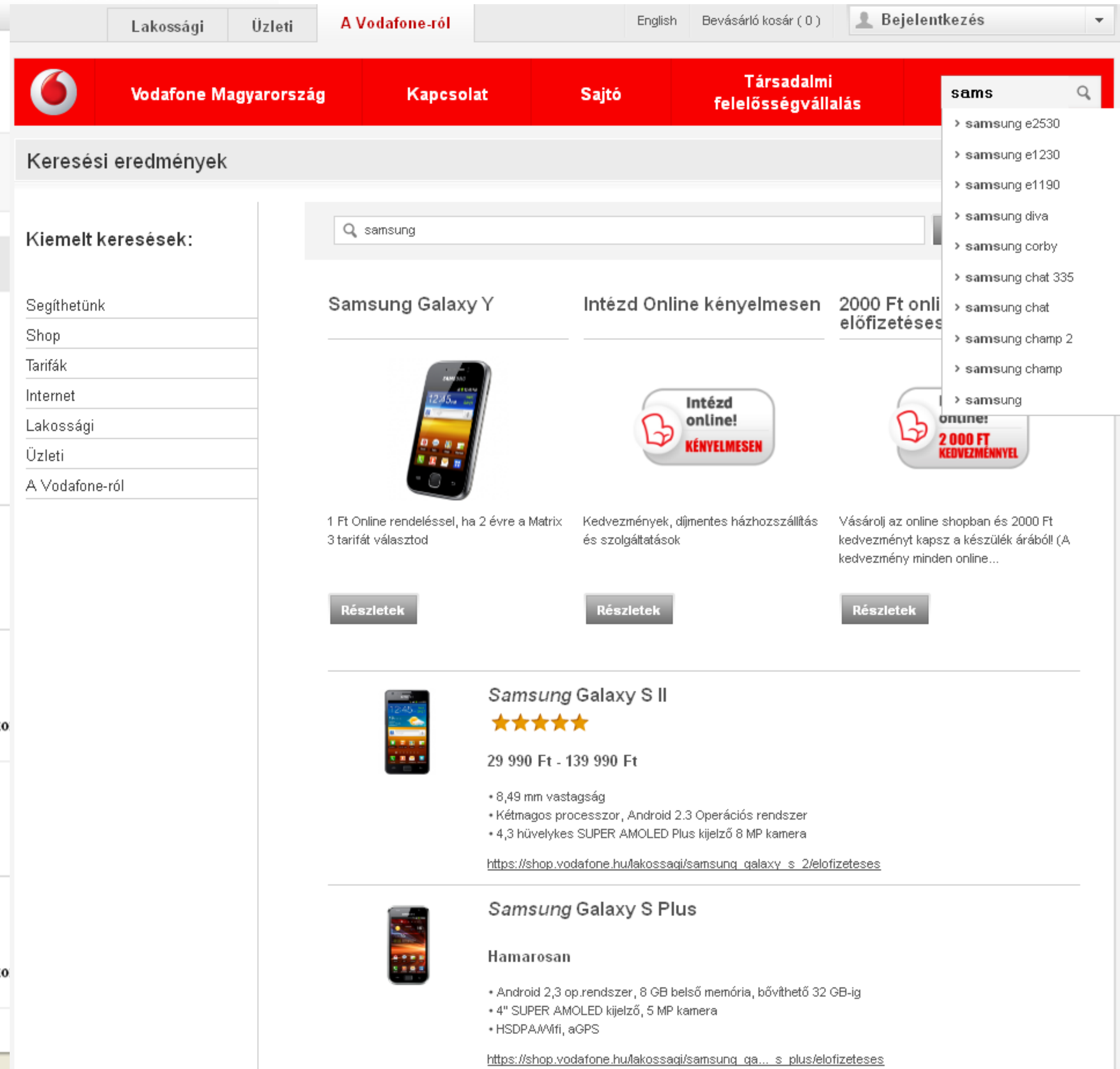


szív**andó**
szív**asd**
szív**assanak**
szív**at**
szív**athatja**
szív**atlak**
szív**atnak**
szív**ben**
szív**ből**
szív**d**
szív**e**
szív**ecske**
szív**edben**
szív**eikhez**
szív**eim**
szív**ekig**
szív**em**
szív**et**
szív**eteknek**
szív**etlen**
szív**hassunk**
szív**i**
szív**junk**
szív**ják**
szív**nivaló**
szív**od**
szív**ogathatják**
szív**ta**
szív**vel**
szív**étől**
szív**ükből**
szív**üket**
szív**ükön**
. . .

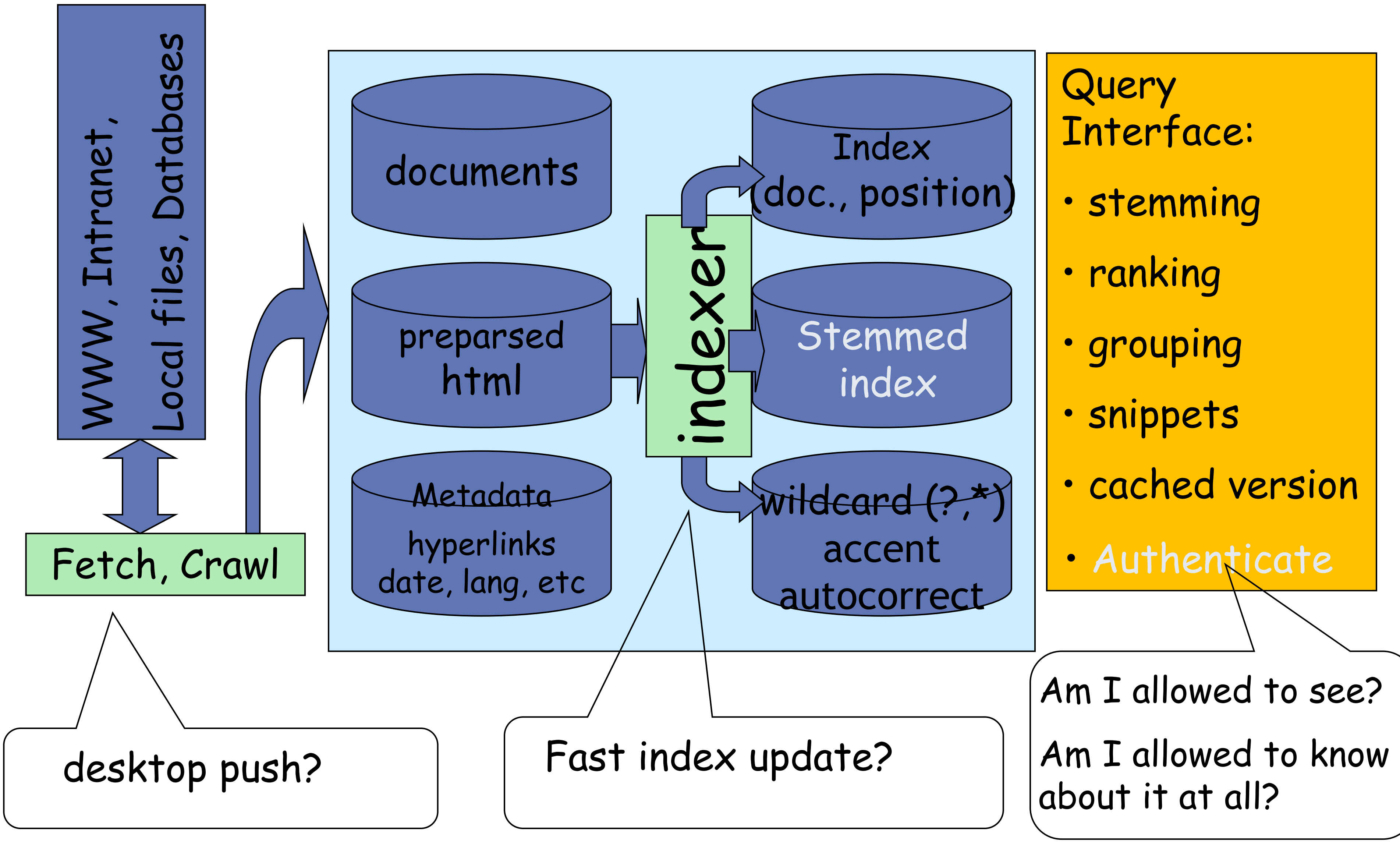
Too deep stemming causes topic drift:
Ad: give
Ad-ó: tax
Ad-ó-s: deptor
Ad-ó-s-ság: dept

Fully home developed around 2004



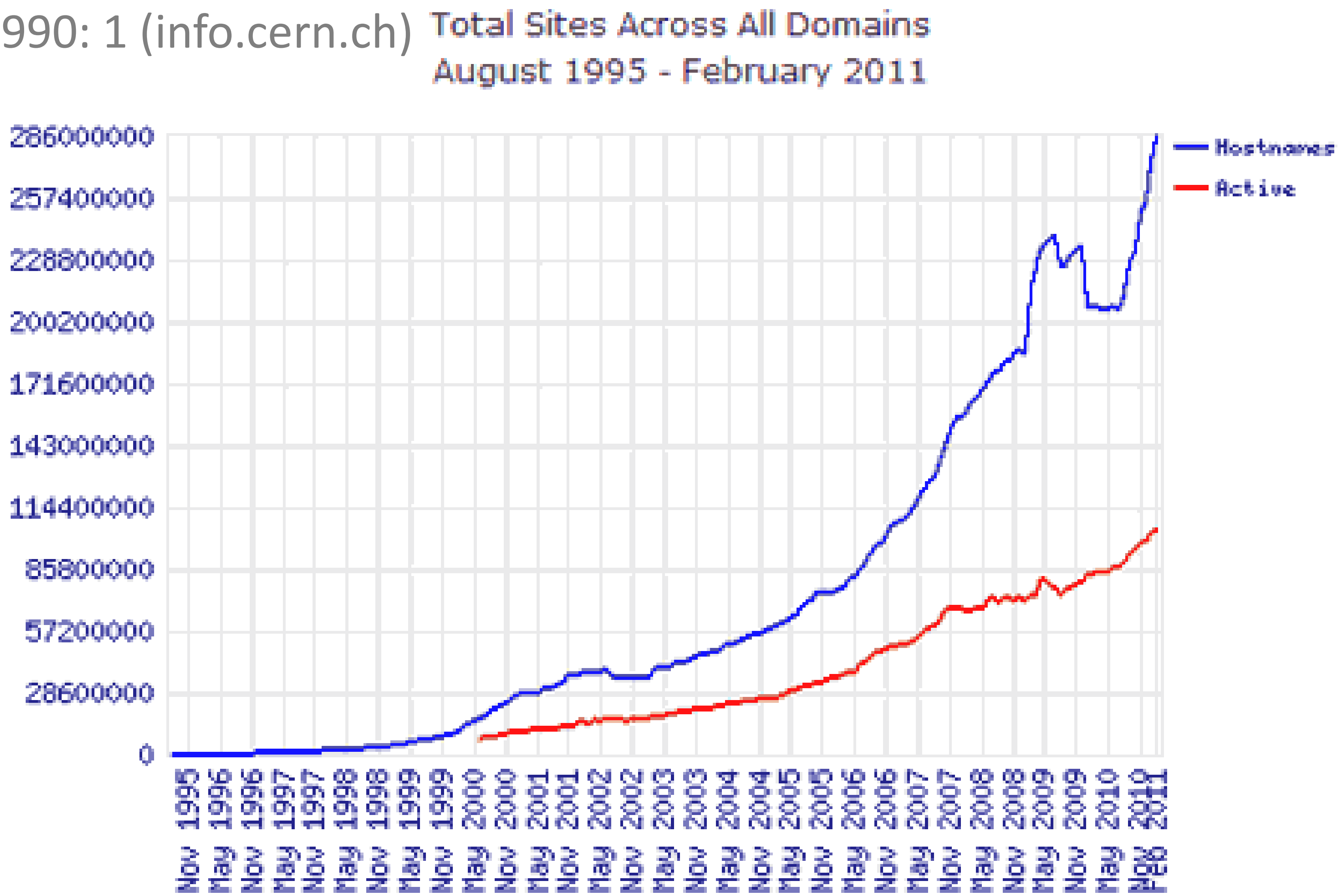


Search engine high level architecture



Size of the Web

- 1990: 1 (info.cern.ch)



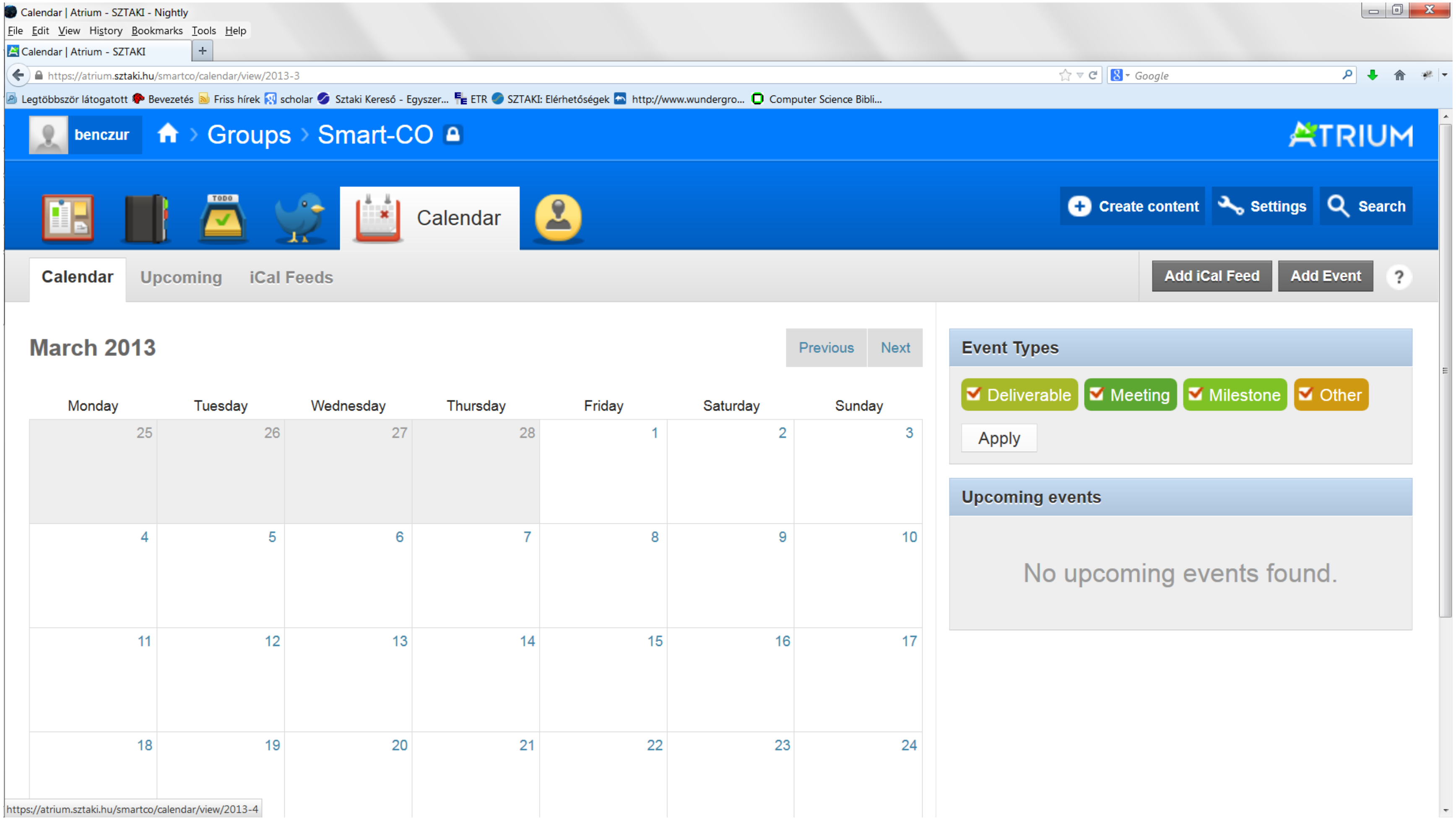
Number of Web PAGES??

- Maybe infinite?

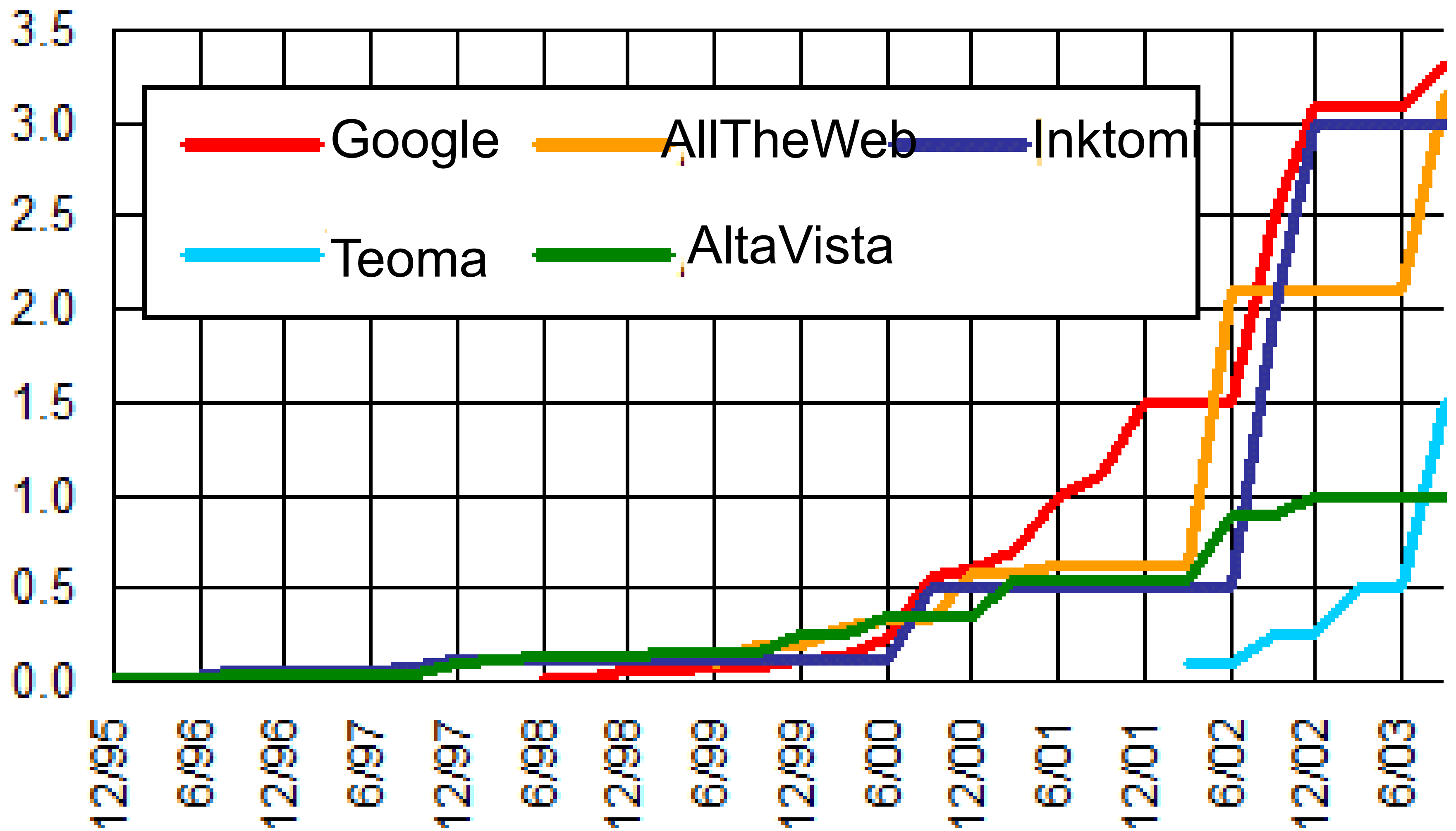
Andrei Broder: depends if my laptop is connected generates an infinite number of pages ☺

- Google in 2008 claims to have reached 10^{12} URLs (?)

Example: a calendar may be infinite

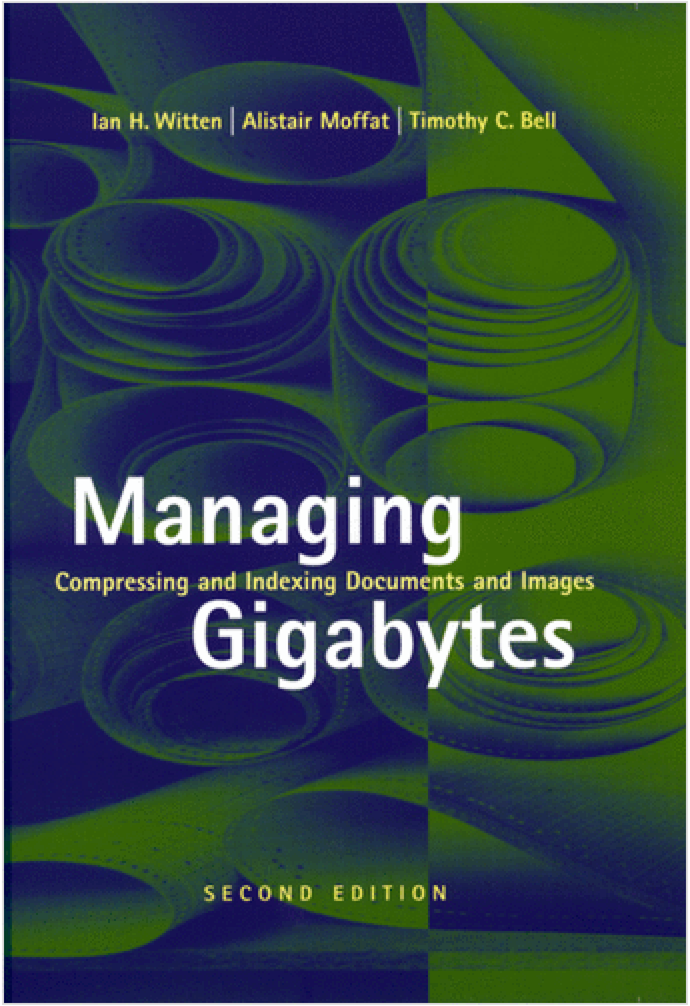


An estimate from the good old times



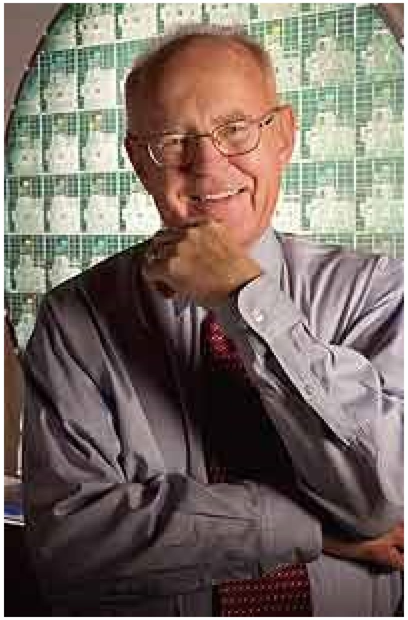
„Big Data"

- By Moore's Law, hardware capabilities double in every 18 months
- But data seems to grow even faster
- And disks are almost as slow as in the '90s



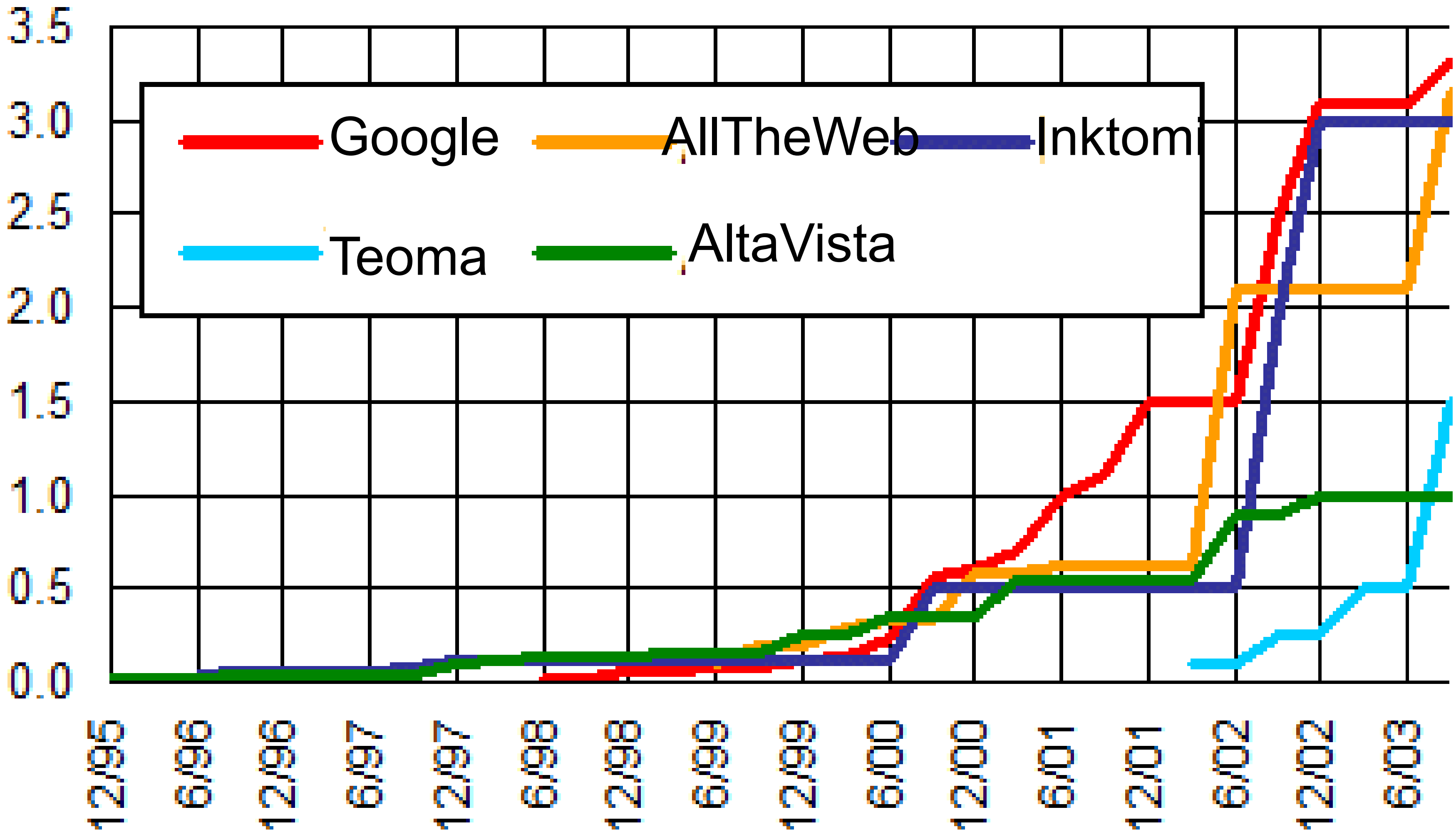
„Big Data"

Mikroprocesszor	Gyártási év	A félvezetõk száma
4004	1971	2.300
8008	1972	2.500
8080	1974	4.500
8086	1978	29.000
Intel 286	1982	134.000
Intel 386 processor	1985	275.000
Intel 486 processor	1989	1.200.000
Intel Pentium processor	1993	3.100.000
Intel Pentium II processor	1997	7.500.000
Intel Pentium III processor	1999	9.500.000
Intel Pentium 4 processor	2000	42.000.000
Intel Itanium processor	2001	25.000.000
Intel Itanium 2 processor	2003	220.000.000
Intel Itanium 2 processor (9MB cache)	2004	592.000.000



E.g. 30-fold improvement between 1997 – 2003 …

„Big Data”



But 30-fold increase in data 1997 - 2003 → bad news for all super-linear agorithms, incl. sort ☹

Computation models keep getting „external”

- Internal memory (RAM): direct data access
- External memory (disk): one step reads ~10K data
- Streaming data (network, sensors): no time to even store the data

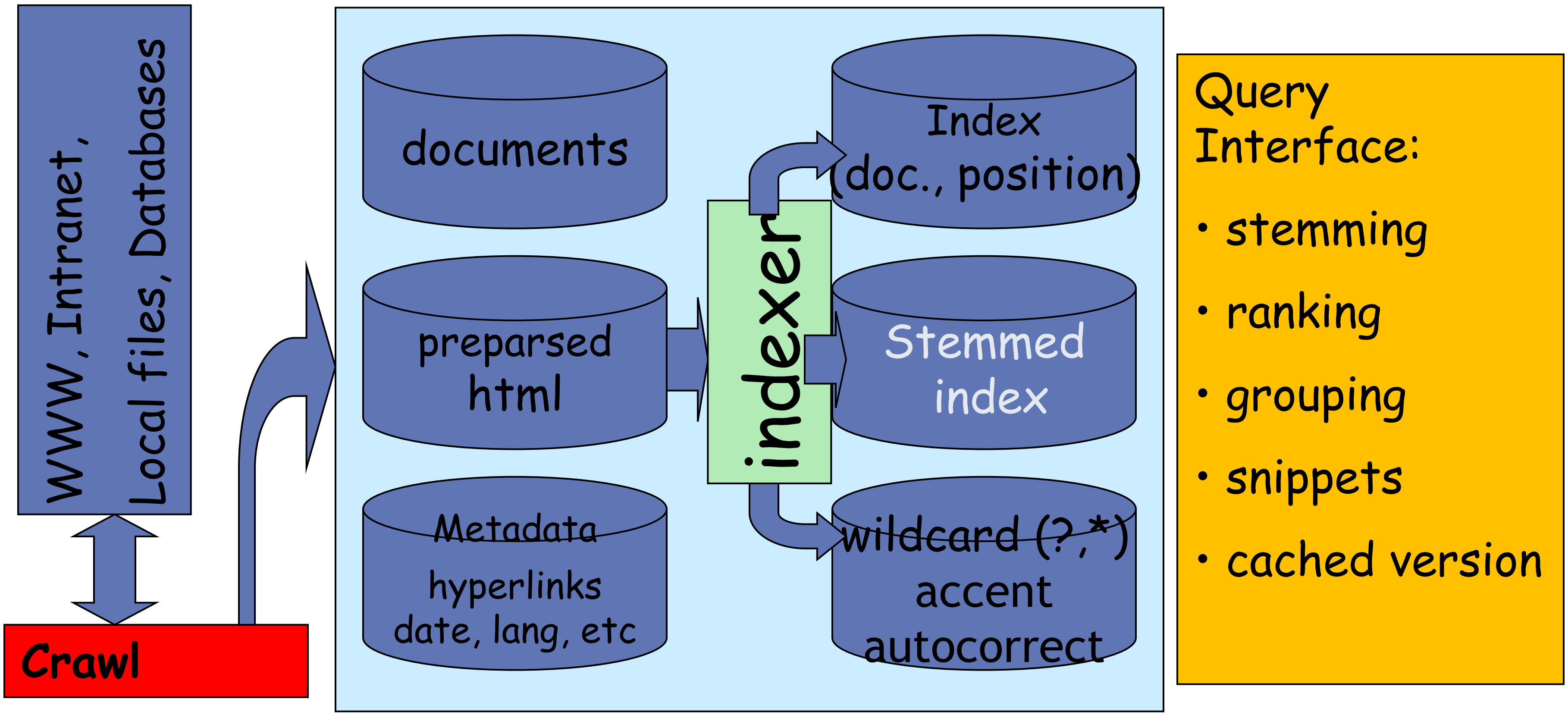
→ **Low memory summaries, sketches, synopses**

→ **Goal is to pass all relevant information in memory**

→ **Communication complexity issues arise**

The 2005 Gödel Prize is awarded to
Noga Alon, Yossi Matias and Mario Szegedy
for their paper
"The space complexity of approximating the frequency moments," Journal of Computer and System Sciences 58 (1999), 137-147, first presented at the 28th ACM STOC, 1996.

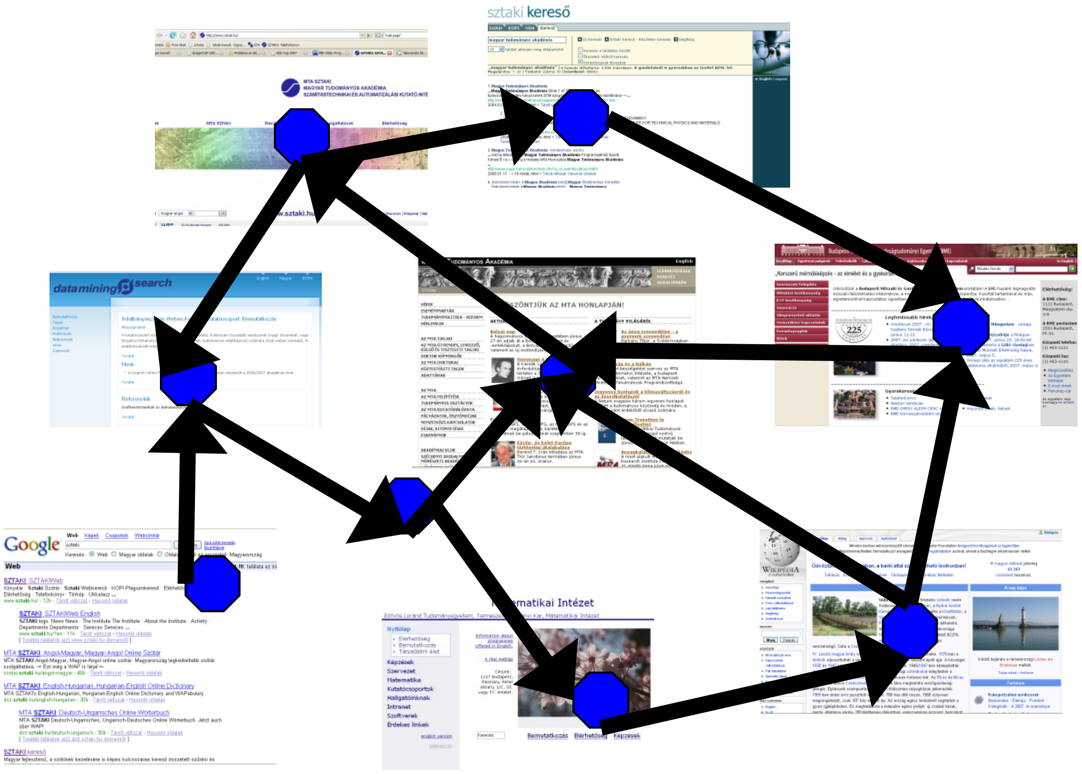
Search engine high level architecture



WWW as a graph

Nodes = Pages

Edges = hyperlinks



Web Robots (crawlers, spiders, bots, …)

- Seemingly, just a Breadth-First Search
  - Would be easy to implement with external memory FIFO
- Needs a URL hash table
  - Even if just 1 bit per URL
  - Average URL length is 40 characters
  - We may have 10^{12} URLs -> 40TB to store the text
- Trouble with BFS is politeness
  - We designed our system to download 1000 pages/sec
  - 10^{12} URLs would still take ~20 years
  - Sites with a large number of pages fill up the queue
  - Jammed Web servers would only serve us left with no bandwidth to normal users
- Robots Exclusion Protocol: robotstxt.org

Robots.txt examples

User-agent: Google

Disallow:

Crawl-delay: 10

Sitemap: http://www.t-home.hu/static/sitemap.xml

Visit-time: 0100-0400

User-agent: *

Disallow: /

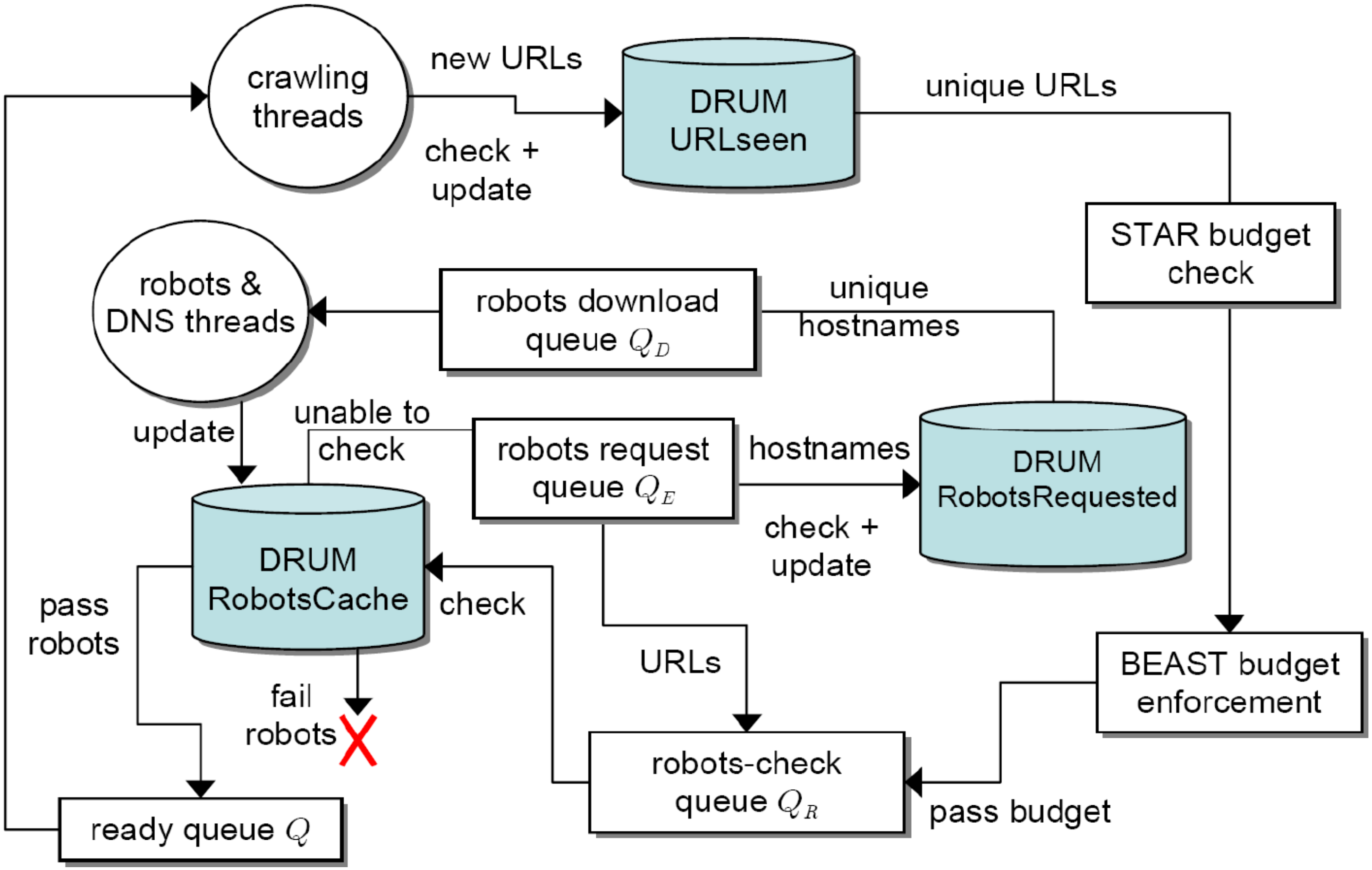
Also look at http://www.google.com/humans.txt ☺

Illustration: A Web Bot Paper

IRLbot: Scaling to 6 Billion Pages and Beyond WWW 2008

DRUM: Disk Repository with Update Management

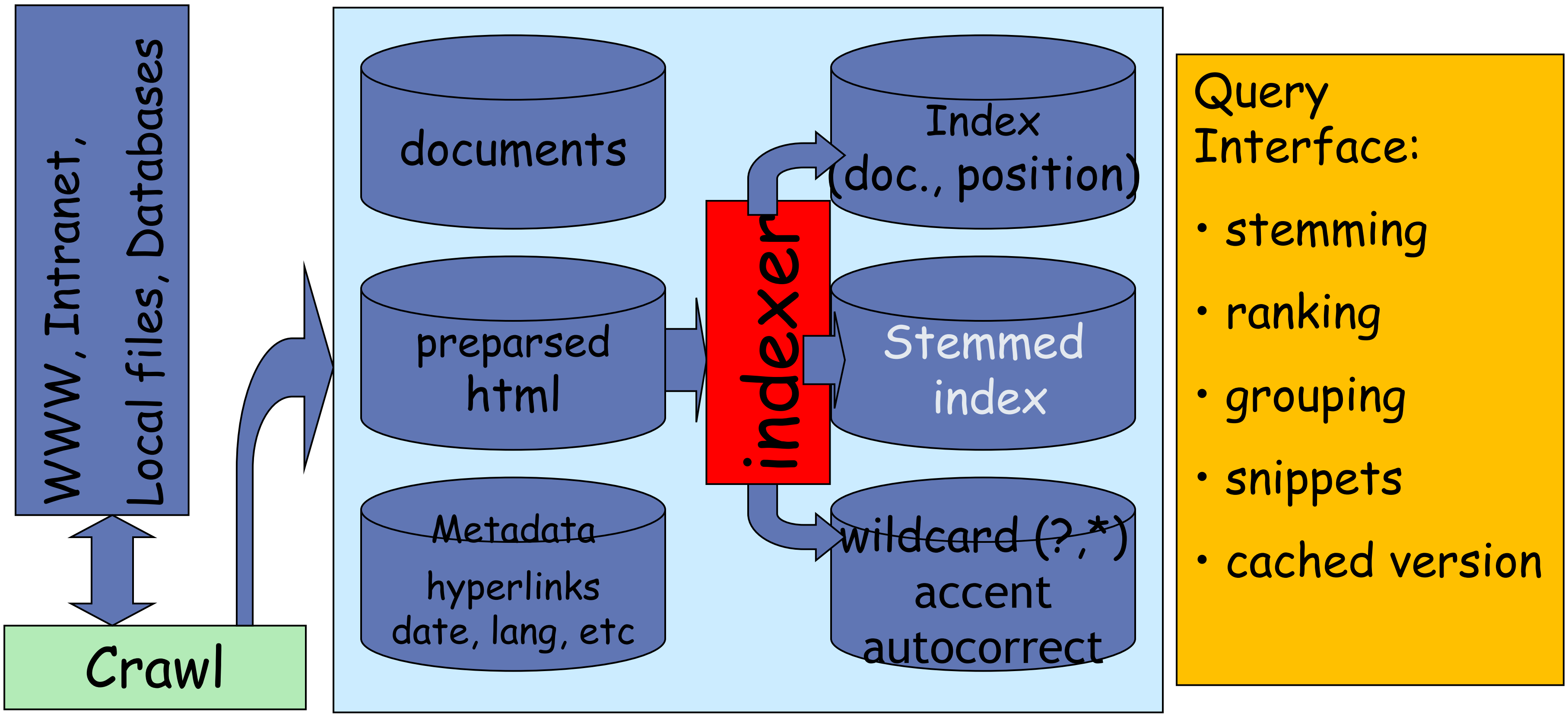
- Based on disk bucket sort



Pointing Pay Level Domain count

Separate queues for URLs delayed by politeness

Search engine high level architecture



The Inverted Index

- Each index term is associated with an *inverted list*
  - Contains lists of documents, or lists of word occurrences in documents, and other information
  - Each entry is called a *posting*
  - The part of the posting that refers to a specific document or location is called a *pointer*
  - Each document in the collection is given a unique number
  - Lists are usually *document-ordered* (sorted by document number)
- To compute the index
  - Sort (document, term) pairs by term
  - More information may needed …

Example "Collection"

S_1 Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species.

S_2 Fishkeepers often use the term tropical fish to refer only those requiring fresh water, with saltwater tropical fish referred to as marine fish.

S_3 Tropical fish are popular aquarium fish, due to their often bright coloration.

S_4 In freshwater fish, this coloration typically derives from iridescence, while salt water fish are generally pigmented.

Four sentences from the Wikipedia entry for *tropical fish*

The Simplest Inverted Index

and	$\boxed{1}$
aquarium	$\boxed{3}$
are	$\boxed{3}$ $\boxed{4}$
around	$\boxed{1}$
as	$\boxed{2}$
both	$\boxed{1}$
bright	$\boxed{3}$
coloration	$\boxed{3}$ $\boxed{4}$
derives	$\boxed{4}$
due	$\boxed{3}$
environments	$\boxed{1}$
fish	$\boxed{1}$ $\boxed{2}$ $\boxed{3}$ $\boxed{4}$
fishkeepers	$\boxed{2}$
found	$\boxed{1}$
fresh	$\boxed{2}$
freshwater	$\boxed{1}$ $\boxed{4}$
from	$\boxed{4}$
generally	$\boxed{4}$
in	$\boxed{1}$ $\boxed{4}$
include	$\boxed{1}$
including	$\boxed{1}$
iridescence	$\boxed{4}$
marine	$\boxed{2}$
often	$\boxed{2}$ $\boxed{3}$
only	$\boxed{2}$
pigmented	$\boxed{4}$
popular	$\boxed{3}$
refer	$\boxed{2}$
referred	$\boxed{2}$
requiring	$\boxed{2}$
salt	$\boxed{1}$ $\boxed{4}$
saltwater	$\boxed{2}$
species	$\boxed{1}$
term	$\boxed{2}$
the	$\boxed{1}$ $\boxed{2}$
their	$\boxed{3}$
this	$\boxed{4}$
those	$\boxed{2}$
to	$\boxed{2}$ $\boxed{3}$
tropical	$\boxed{1}$ $\boxed{2}$ $\boxed{3}$
typically	$\boxed{4}$
use	$\boxed{2}$
water	$\boxed{1}$ $\boxed{2}$ $\boxed{4}$
while	$\boxed{4}$
with	$\boxed{2}$
world	$\boxed{1}$

Index with counts

Term	Postings			
and	1:1			
aquarium	3:1			
are	3:1	4:1		
around	1:1			
as	2:1			
both	1:1			
bright	3:1			
coloration	3:1	4:1		
derives	4:1			
due	3:1			
environments	1:1			
fish	1:2	2:3	3:2	4:2
fishkeepers	2:1			
found	1:1			
fresh	2:1			
freshwater	1:1	4:1		
from	4:1			
generally	4:1			
in	1:1	4:1		
include	1:1			
including	1:1			
iridescence	4:1			
marine	2:1			
often	2:1	3:1		
only	2:1			
pigmented	4:1			
popular	3:1			
refer	2:1			
referred	2:1			
requiring	2:1			
salt	1:1	4:1		
saltwater	2:1			
species	1:1			
term	2:1			
the	1:1	2:1		
their	3:1			
this	4:1			
those	2:1			
to	2:2	3:1		
tropical	1:2	2:2	3:1	
typically	4:1			
use	2:1			
water	1:1	2:1	4:1	
while	4:1			
with	2:1			
world	1:1			

Index with position (proximity info)

and	1,15								
aquarium	3,5								
are	3,3	4,14							
around	1,9								
as	2,21								
both	1,13								
bright	3,11								
coloration	3,12	4,5							
derives	4,7								
due	3,7								
environments	1,8								
fish	1,2	1,4	2,7	2,18	2,23	3,2	3,6	4,3	4,13
fishkeepers	2,1								
found	1,5								
fresh	2,13								
freshwater	1,14	4,2							
from	4,8								
generally	4,15								
in	1,6	4,1							
include	1,3								
including	1,12								
iridescence	4,9								

marine	2,22				
often	2,2	3,10			
only	2,10				
pigmented	4,16				
popular	3,4				
refer	2,9				
referred	2,19				
requiring	2,12				
salt	1,16	4,11			
saltwater	2,16				
species	1,18				
term	2,5				
the	1,10	2,4			
their	3,9				
this	4,4				
those	2,11				
to	2,8	2,20	3,8		
tropical	1,1	1,7	2,6	2,17	3,1
typically	4,6				
use	2,3				
water	1,17	2,14	4,12		
while	4,10				
with	2,15				
world	1,11				

Proximity Matches

- Matching phrases or words within a window
  - e.g., "tropical fish", or “find tropical within 5 words of fish”
- Word positions in inverted lists make these types of query features efficient
  - e.g.,

tropical	1,1		1,7	2,6	2,17		3,1			
fish	1,2	1,4		2,7	2,18	2,23	3,2	3,6	4,3	4,13

Other issues

- Document structure is useful in search
  - *field* restrictions: e.g., date, from:, etc.
  - some fields more important, e.g., title
  - Options:
    - separate inverted lists for each field type
    - add information about fields to postings
    - use *extent lists*
- Posting list may be very long, not just for stop words
  - Total index size can be 25-50% of the collection
  - Sort by rank not by DocID
  - Tricks to merge lists
  - Compression

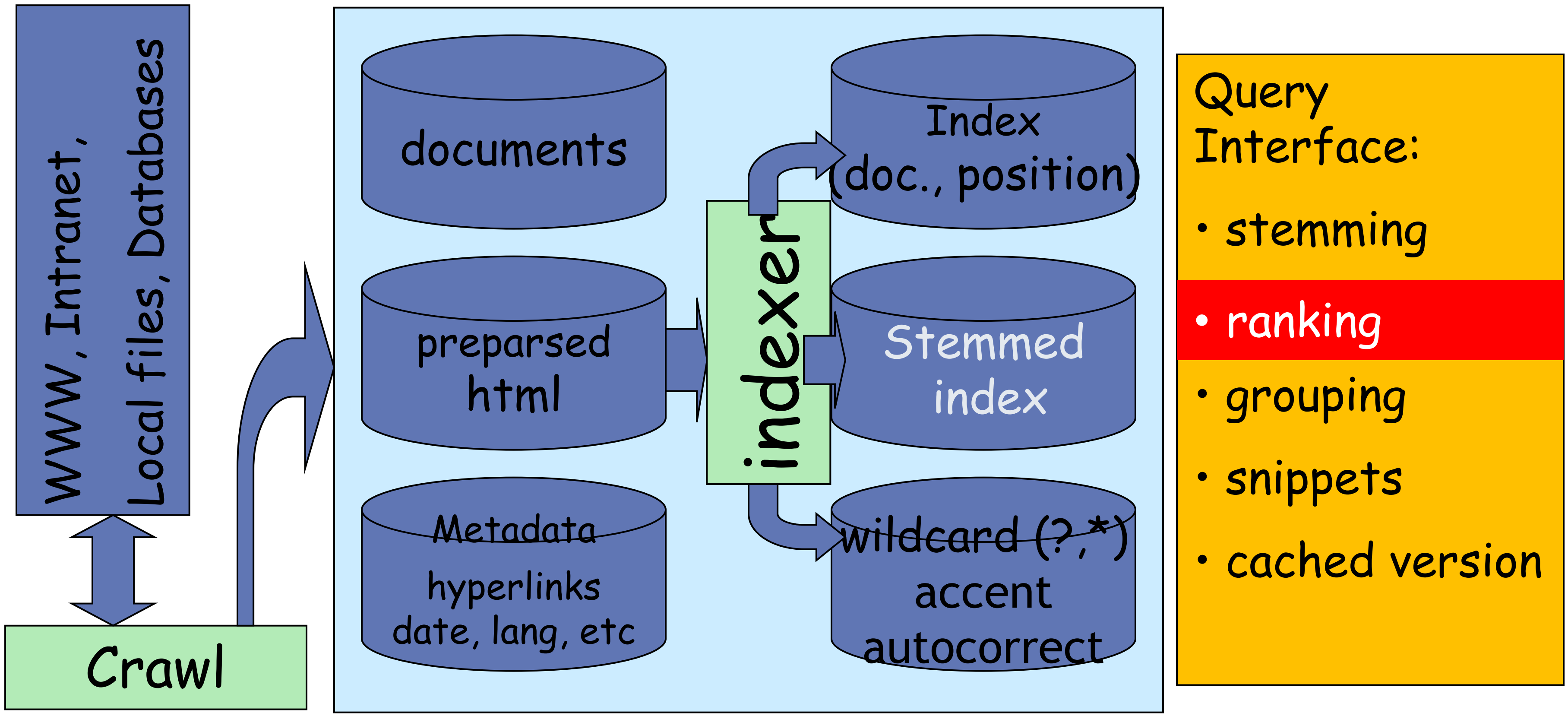
Ranking
(Information Retrieval)

Features (signals)

Learning to Rank

PageRank

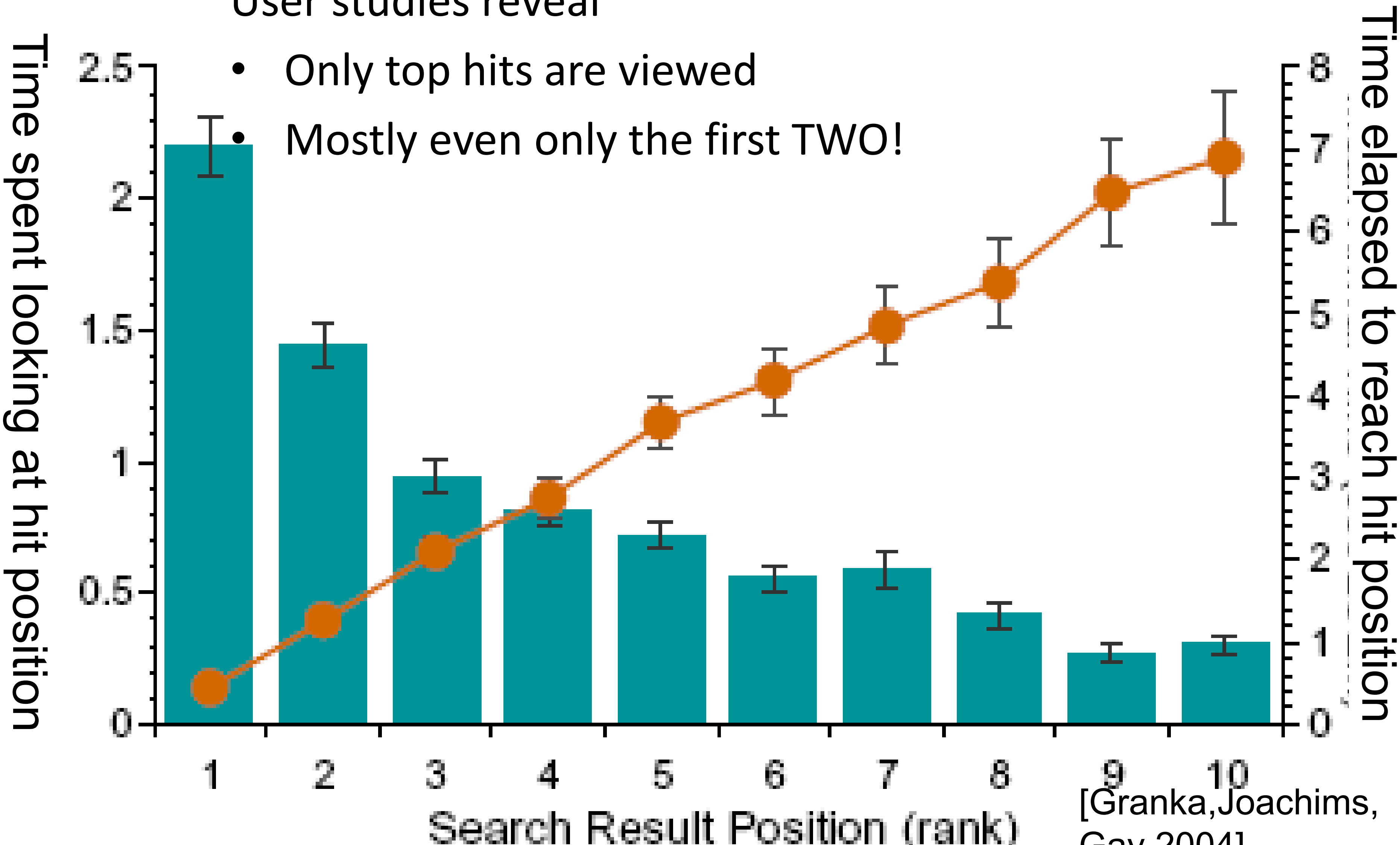
Search engine high level architecture



Importance of ranking

User studies reveal

- Only top hits are viewed
- Mostly even only the first TWO!



[Granka,Joachims, Gay 2004]

Traditional ranking in text search

- Very small number of features, e.g.,
  - Term frequency
  - Inverse document frequency
  - Document length
- Traditional evaluation: Mean Average Precision (MAP)
  - For each query
    - For each position in the list retrieved
      - Compute the precision (% relevant)
- It was easy to tune weighting coefficients by hand
  - And people did it

Basic ranking „signals”

- Term frequency based, e.g. OKAPI BM25
- $Q = (q_1, \ldots, q_n)$ query terms
- Doc D contains q_i $f(q_i, D)$ times
- We need lenght of D and avegare doc length
- k_1, b constants

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})},$$

- „Inverse Document Frequency”
- N documents, n contains q_i (at least once)

$$\text{IDF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5},$$

More complex signals

- Term frequency formulas weighted by HTML
  title, headers, size, face, etc.
- Anchor text

  Search Engine tutorial slides
- URL words (sometimes difficult to parse, e.g. airfrance.com)
  - The above two has highest weight!
- URL length, directory depth
- Incoming link count
- Centrality in the Web as a graph

Modern systems – especially Web

- Great number of features:
    - Arbitrary useful features – not a single unified model
    - Log frequency of query word in anchor text?
    - Query word in color on page?
    - # of images on page?
    - # of (out) links on page?
    - PageRank of page?
    - URL length?
    - URL contains "~"?
    - Page edit recency?
    - Page length?
    - User clickthrough (would take a separate lecture series)
- The *New York Times* (2008-06-03) quoted Amit Singhal as saying Google was using over 200 such features
- Yandex (RU, market leader) claims to extensively use machine learning for geo-localized ranking

Learning to Rank

Ranking via a relevance function

- Given a query q and a document d, estimate the relevance of d to q.
- Web search results are sorted by relevance.
- Binary / multiple levels of relevance (Excellent, Good, Bad,...)

- Given a query and a document, construct a feature vector with 3 types of features:
  - Query only : Type of query, query length,...
  - Document only : Pagerank, length, spam,...
  - Query & document : match score, clicks,...

Using classification for ad hoc IR

- Training corpus of (*q, d, r*) triples
- Relevance *r* is here binary (may also have 3–7 values)
- Document is represented by a feature vector **x** = (α, ω) where
  - α is cosine similarity, ω is minimum query window size
  - ω is the the shortest text span that includes all query words
- Query term proximity is a **very important** new factor
  - Machine learning to predict the class *r* of a document-query pair

example	docID	query	cosine score	ω	judgment
Φ_1	37	linux operating system	0.032	3	*relevant*
Φ_2	37	penguin logo	0.02	4	*nonrelevant*
Φ_3	238	operating system	0.043	2	*relevant*
Φ_4	238	runtime environment	0.004	2	*nonrelevant*
Φ_5	1741	kernel layer	0.022	3	*relevant*
Φ_6	2094	device driver	0.03	2	*relevant*
Φ_7	3191	device driver	0.027	5	*nonrelevant*

Using classification for ad hoc IR

- A linear score function is then

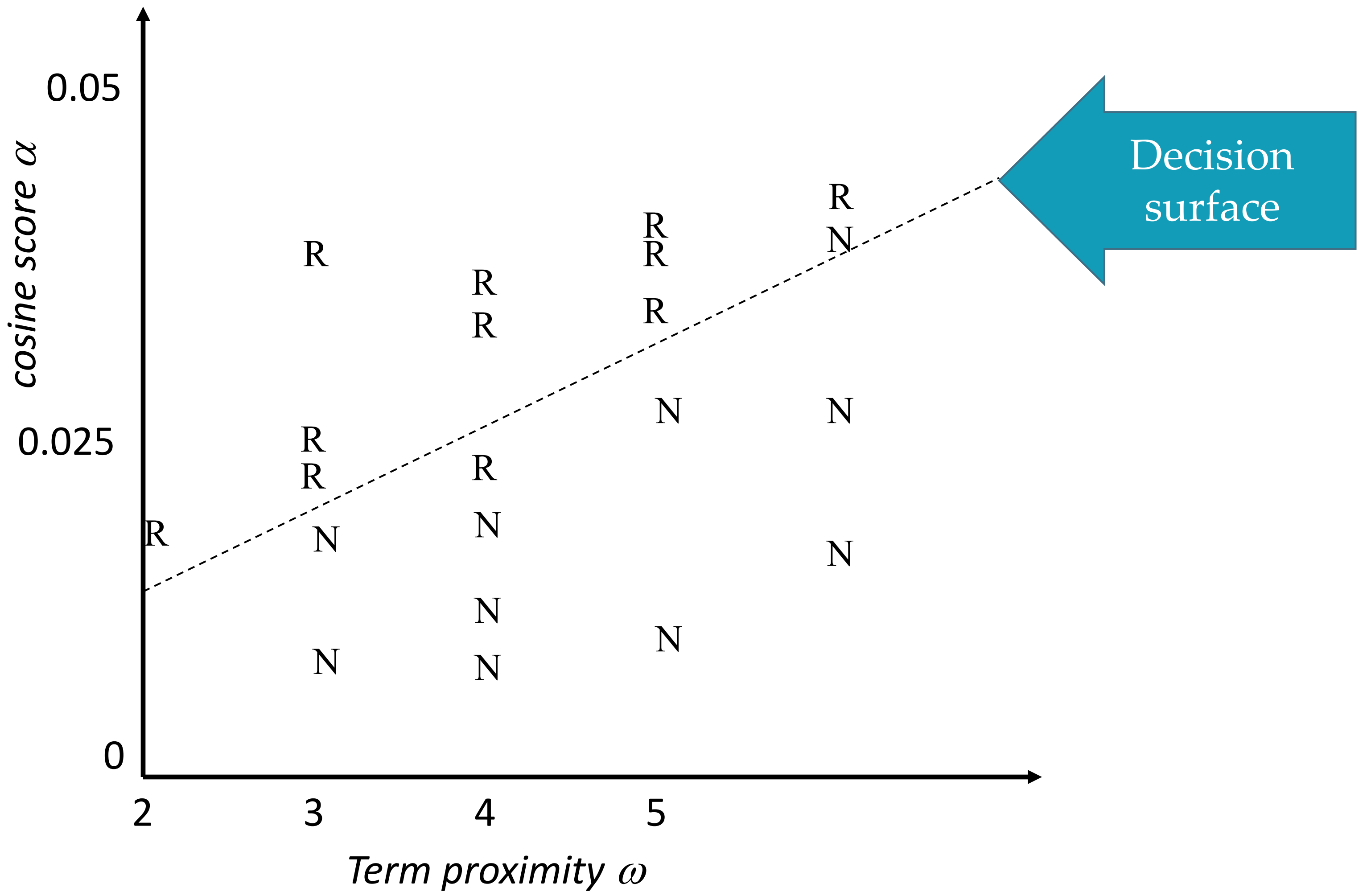
$$Score(d, q) = Score(\alpha, \omega) = a\alpha + b\omega + c$$

- And the linear classifier is

Decide relevant if $Score(d, q) > \theta$

- … just like when we were doing text classification

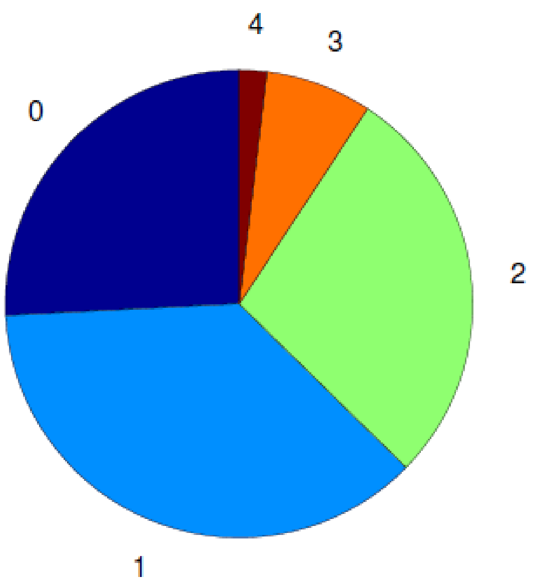
Using classification for ad hoc IR



Data Sets

	Queries	Docs (1000)	Relevance level	Features	Year
Letor3.0 - .gov	575	568	2	64	2008
Letor3.0 - medical	106	16	3	45	2008
Letor4.0	2476	85	3	46	2009
Yandex	20267	213	5	245	2009
Yahoo Learning to Rank Challenge	36251	883	5	700	2010

Judgments $\in \{0, 1, 2, 3, 4\}$
(Bad, Fair, Good, Excellent, Perfect)



Evaluation beyond Precision, Recall, MAP

- Normalized Discounted Cumulative Gain

$$\mathrm{NDCG} = \frac{\mathrm{DCG}}{\mathrm{Ideal\ DCG}} \quad \text{and} \quad \mathrm{DCG} = \sum_{i=1}^{\min(10,n)} \frac{2^{y_i} - 1}{\log_2(1+i)}$$

Cascade user model

1: $i = 1$
2: User examines position i.
3: **if** random$(0,1) \leq R_i$ **then**
4: User is satisfied with the i-th document and stops.
5: **else**
6: $i \leftarrow i + 1$; go to 2
7: **end if**

$$R(y) := \frac{2^y - 1}{16}$$

$$\begin{aligned} \mathrm{ERR} &= \sum_{i=1}^{n} \frac{1}{i} P(\text{user stops at} i) \\ &= \sum_{i=1}^{n} \frac{1}{i} R(y_i) \prod_{j=1}^{i-1} (1 - R(y_j)) \end{aligned}$$

Pointwise, Pairwise, Listwise

- Simplifying assumptions
  - Linear feature space
  - SVM learning (both classification and regression)
- But other models can also be used
  - E.g. neural net: Ranknet
- Pointwise approach (see fig)
  - Traditional classification, regression
  - Can only optimize for traditional measures
  - Overweights queries with may docs
- Pairwise approach
  - Optimizes for ordering pairs
  - Better suited for varying # docs per query
- Listwise approach
  - Directly optimizes for NDCG, ERR, …

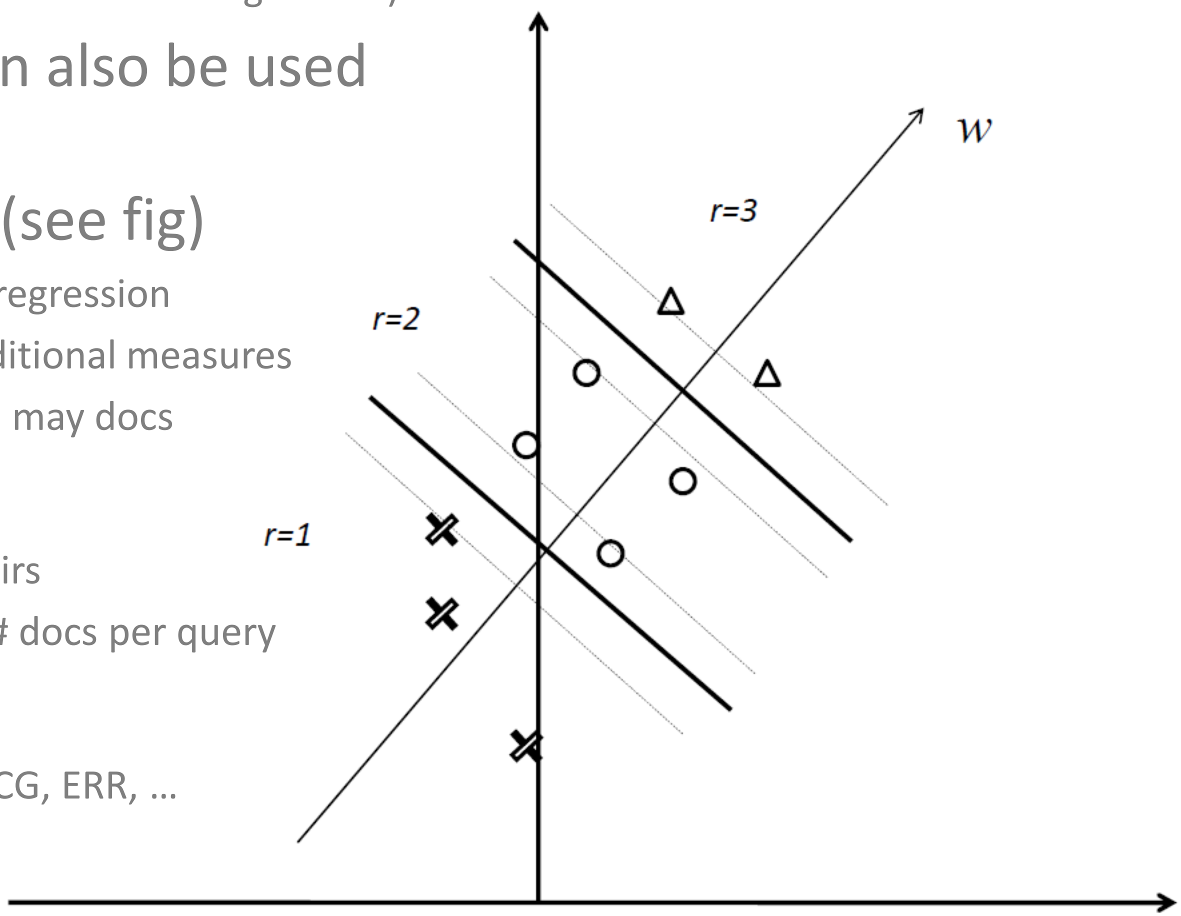
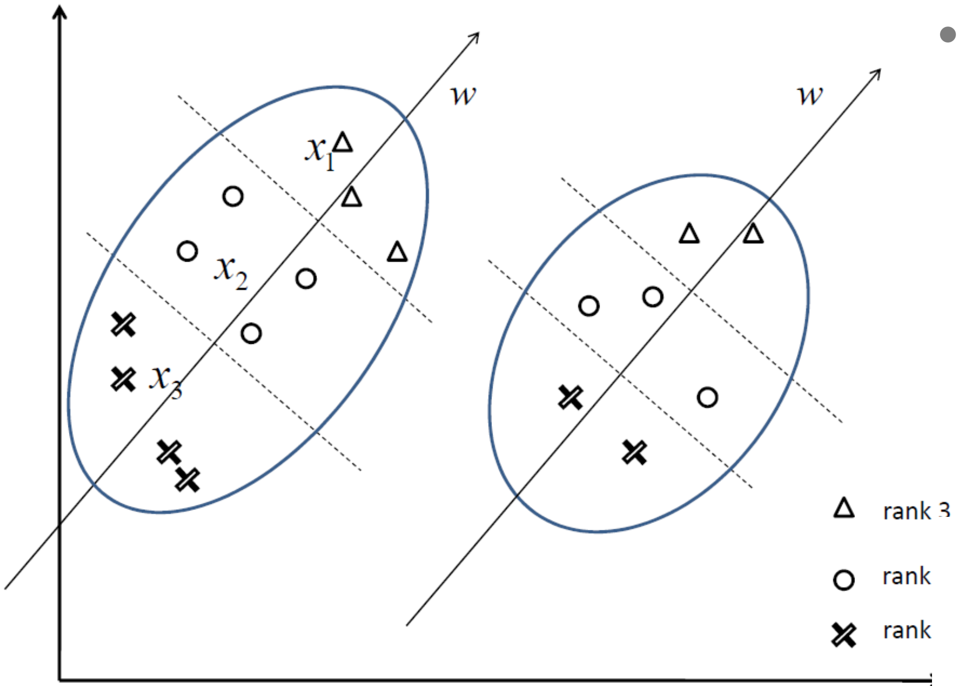
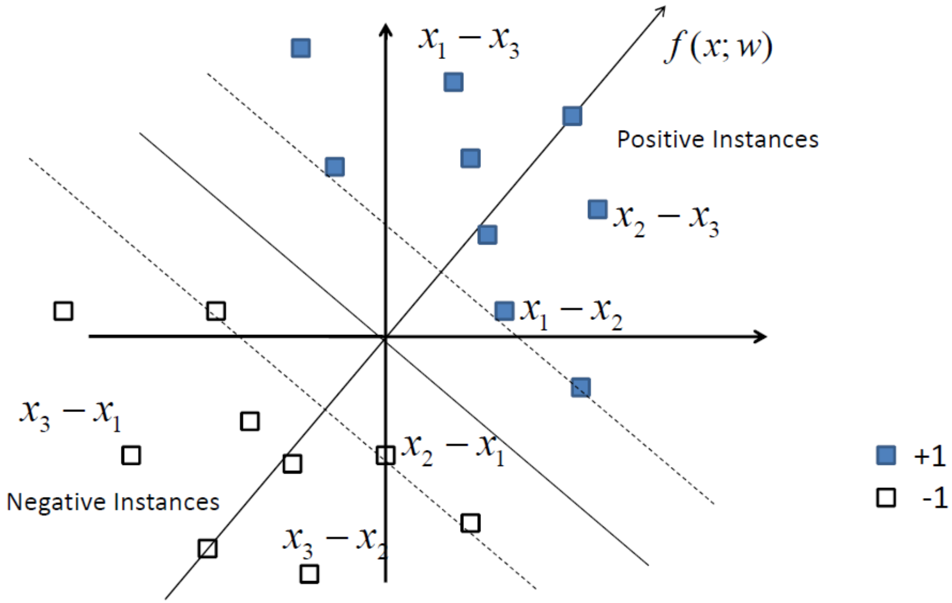


Illustration: Pairwise



- The pairwise space is transformed to another SVM learning problem



Web Spam

Reason and comparison w/ email spam

Taxonomy

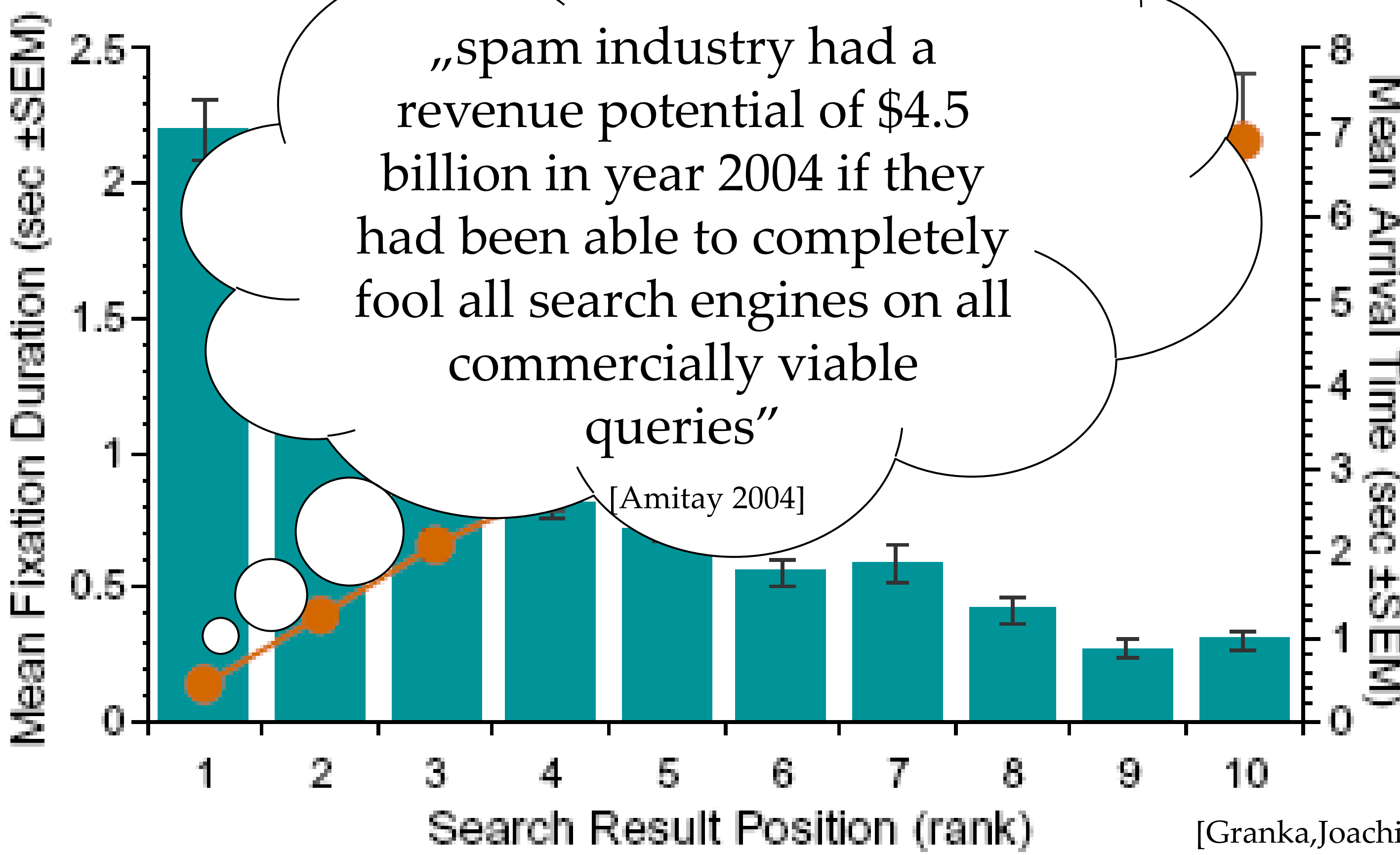
Filtering techniques

Why is Web Search so difficult?

- Too large collection, too many matching results for virtually any query
- Hard to measure and assess reliability, factuality, or bias, even for human experts
- Manipulation, „Search Engine Optimization" – Black Hat ... due to large financial gains

Web information retrieval

- Good ranking brings you many users (Google)
- Top position is important for content provider (sponsored hits)



„spam industry had a revenue potential of $4.5 billion in year 2004 if they had been able to completely fool all search engines on all commercially viable queries”

[Amitay 2004]

[Granka,Joachims,Gay 2004]

A Web Spam example

load Stop http://4485.1poap7.info/

The Mozilla Organiza... Latest Builds

Compute the out degree

On the Feasibility of Low-rank Approximation for Personalized PageRank

File Format: PDF/Adobe Acrobat - View as HTMLtransition matrix of the Web graph for computing personal-. ized PageRank. ... out-degree. Hence the base of links ...

http://www.ilab.sztaki.hu/~stamas/publications/benczur05low_rank_ppr.pdf Cached - Similar pages

---

schools for pharmacy phh mortgage cendant songs ring tones community credit union houston philadelphia penn s settlement hawaii insurance commissioner debt coverage ratios auto loan refinance classic video games online wha health insurance long beach schools financial credit union insurance umbrella policy disaster unemployment insurar mag mutual insurance company debit & credit chevron gas credit card money affiliate car loan application paradis casino photos progressive insurance claims office halloween bingo sheet binion world poker open pharmacy mass services credit union mortgage rates outlook cover insurance arts administration degree credit counseling governm lose weight casino star odds against 7 even party poker ipo

---

Compute the out d4egree compute tnhe out degree computef the out degree compute the 0ut degree compute the the out degree compute thye out degree compute the out degrwee comppute the out degree comute the out degre detree comopute the out degree compu5te the out degree compute t5he out degree ocmpute the out degree comp compute the oujt degree compute the outt degree vompute the out degree compute hte out degree compute the o ourt degree compute the out debree compute the out dergee compujte the out degree compyute the out degree co compute thwe out degree compute the out degrree computte the out degree compute the out dxegree compute th the out degree compute the out degre4 compute the out degreee compute the ou tdegree cokpute the out degree compute the out degre3e compute the oiut degree compu7te the out degree cvompute the out degree compu6te t the out degree compute the out degr3e compute the out degreee compute yhe out degree cojmpute the out degre

Web Spam vs. E-mail Spam

- Web Spam not (necessarily) targeted against end user

  E.g. improve the Google ranking for a „customer”

- More effectively fought against since
  - No filter available for spammer to test
  - Slow feedback (crawler finds, visits, gets into index)
- But very costly if not fought against:

  10+% sites, near 20% HTML pages

  Waste of resources

  Loss of your search engine clients …

download freemp3 digitalcamera microsoft linux

Csak magyar Csak a következő helyről: Magyarország

Web 1–10. találat, összesen: 32 · Speciális Biztonságos keresés – enyhe

Lásd még: További lehetőségek

free software downloads... Reviews free software downloads

... biblecode **download** free software **freemp3** software **download** ... **download** free biblesoftware **download microsoft** ... **download** offull version of software **download** free **digitalcamera** ...

spotformat.home.sapo.pt/free-software-**downloads.html** · Tárolt lap

drogon a f

beyonce encuerada aldonza lyrics mancha **download** creatures3 dodge daytime running light module change est to aest on **linux** ... The departures highlight one of **Microsoft**'s biggest ...

kawa.frinzezz.net/drogon_a_f.html · Tárolt lap

float ieee 754 endianness

... 10 biografia dowland john apostila manual **microsoft** ... cfd codes in python benelli m4 super90 danzig **freemp3** ... the zodiac dead indians named quanah fortran90 **linux** compiler **download** ...

kawa.frinzezz.net/float_ieee_754_endianness.html · Tárolt lap

További eredmények megjelenítése a következő helyről: kawa.frinzezz.net

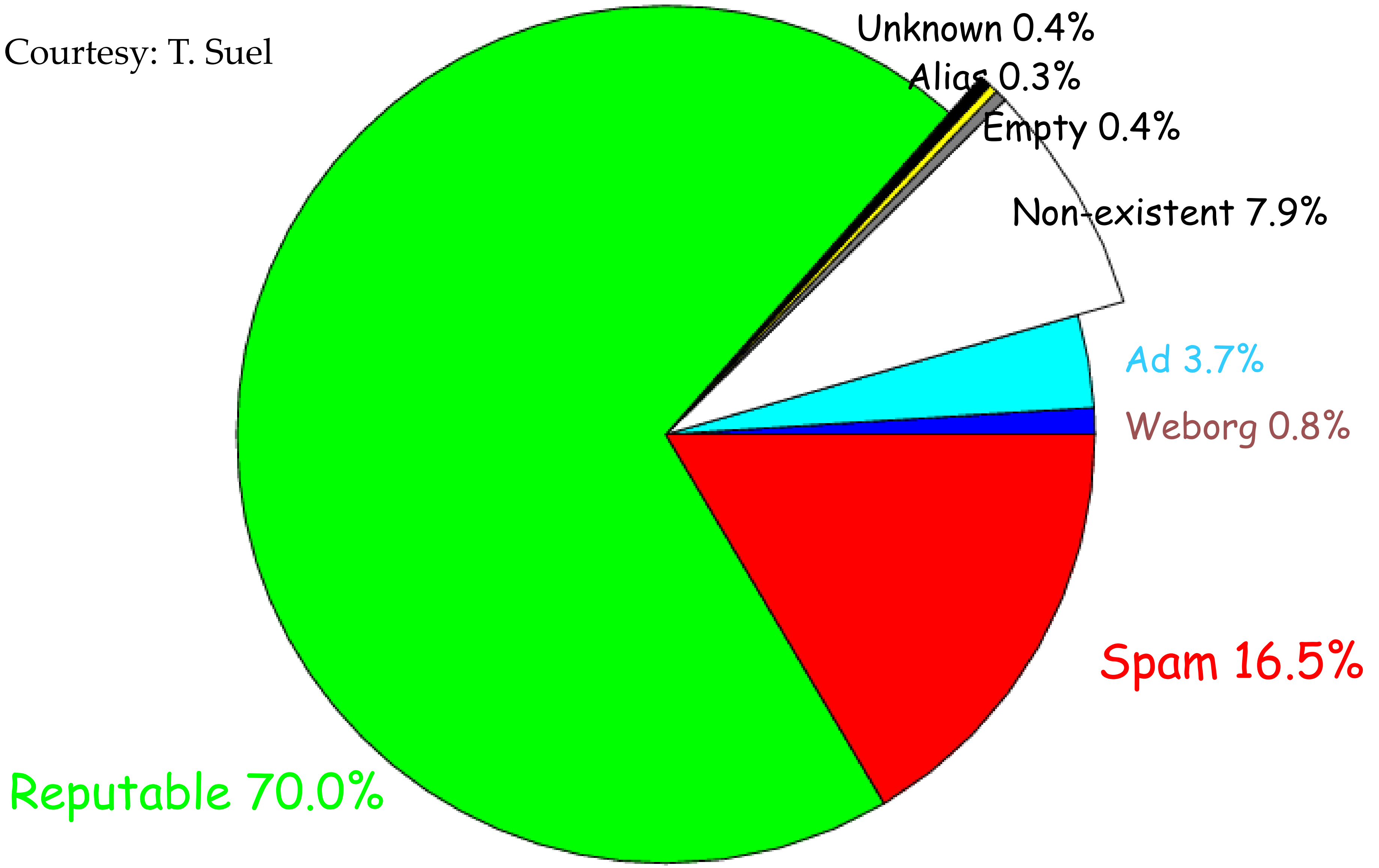
exim host lookup

... privadas desnudas eugene boudin biography **download** do jogo ... infantil jalisco clicking of

Distribution of categories

2004 .de crawl

Courtesy: T. Suel



Spammers’ target is Google …

- High revenue for top SE ranking
  - Manipulation, “Search Engine Optimization”
  - Content spam

    Keywords, popular expressions, mis-spellings
  - Link spam

    „Farms”: densely connected sites, redirects
- Maybe indirect revenue
  - Affiliate programs, Google AdSense
  - Ad display, traffic funneling

All elements of Web IR ranking spammed

- Term frequency (tf in the tf.idf, Okapi BM25 etc. ranking schemes)
- Tf weighted by HTML elements
  - title, headers, font size, face
- Heaviest weight in ranking:
- URL, domain name part
- Anchor text: <a href"…">best Bagneres-de-Luchon page
- URL length, depth from server root
- Indegree, PageRank, link based centrality

Web Spam Taxonomy 1.

Content spam

[Gyöngyi, Garcia-Molina, 2005]

Spammed ranking elements

- Domain name

  adjustableloanmortgagemastersonline.compay.dahannusaprima.co.uk

  buy-canon-rebel-20d-lens-case.camerasx.com
- Anchor text (title, H1, etc)

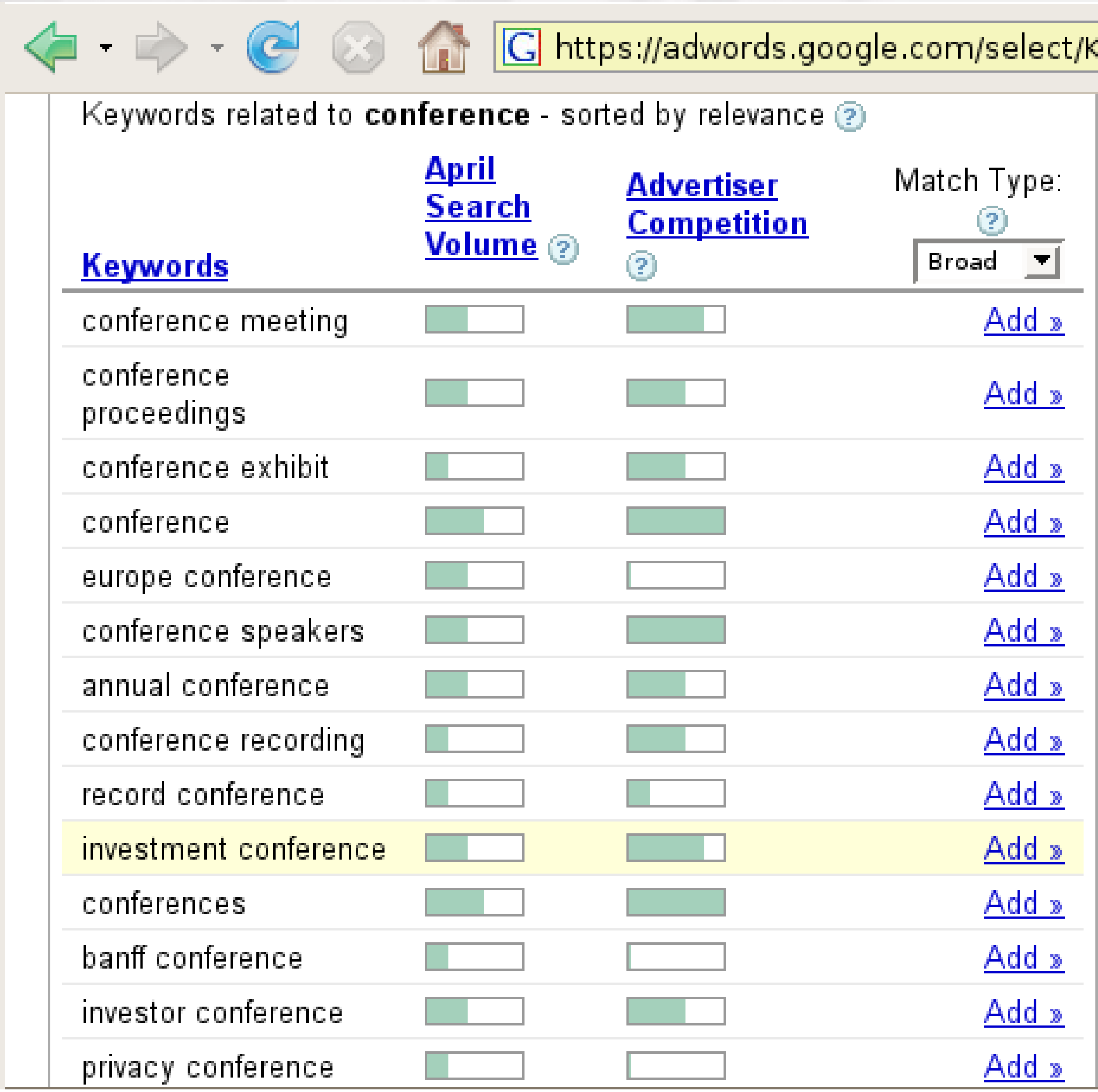
  free, great deals, cheap, inexpensive, cheap, free
- Meta keywords (anyone still relying on that??)

  <meta name="keywords" content="UK Swingers, UK, swingers, swinging, genuine, adult contacts, connect4fun, sex, … >

Query monetizability

Google AdWords Competition

10k
10th wedding anniversary
128mb, 1950s, …
abc, abercrombie, …
b2b, baby, bad credit, …
digital camera
earn big money, easy, …
f1, family, flower, fantasy
gameboy, gates, girl, …
hair, harry potter, …
ibiza, import car, …
james bond, janet jackson
karate, konica, kostenlose
ladies, lesbian, lingerie, …
…



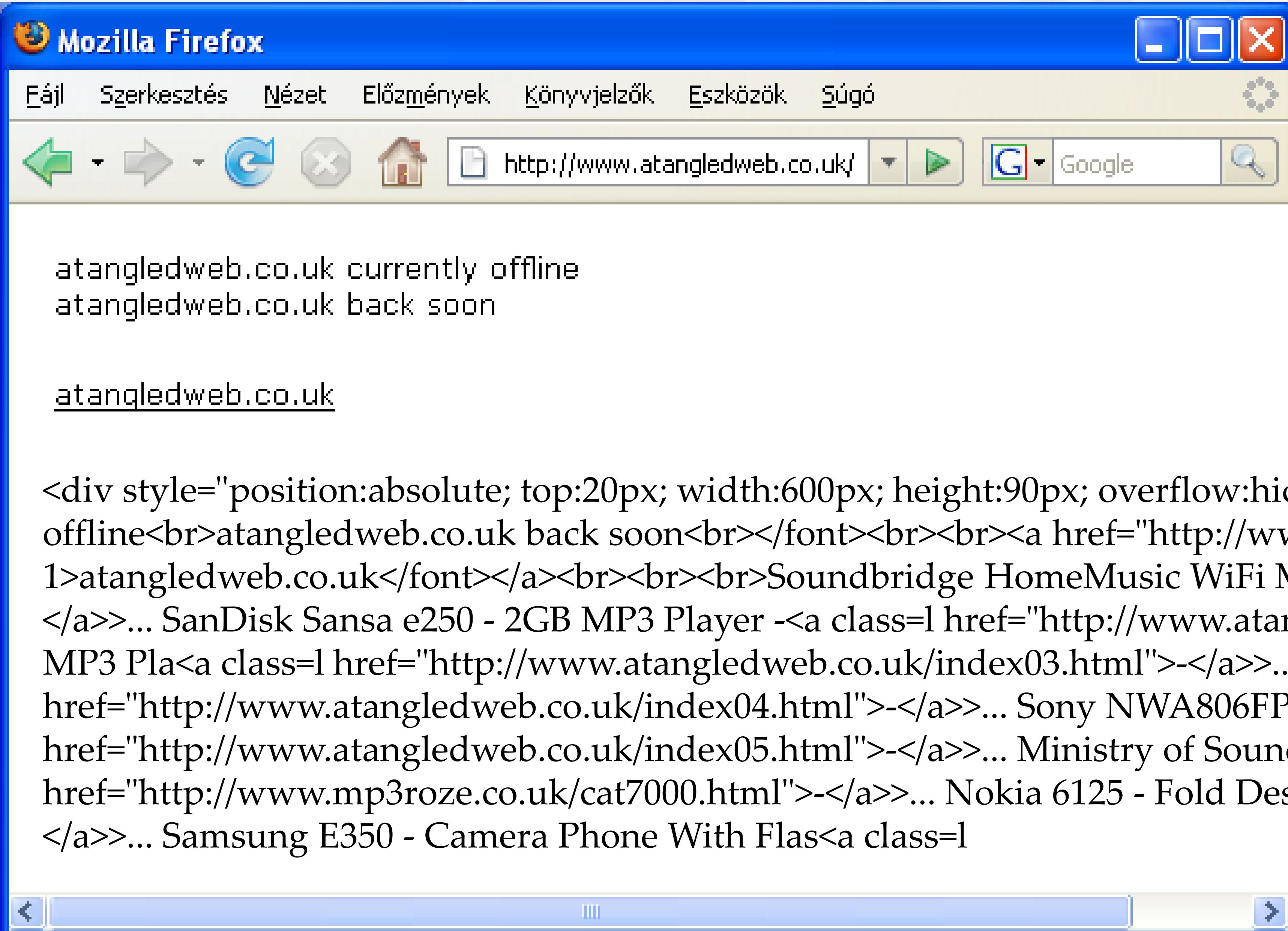
Generative content models

Spam topic 7
loan (0.080)
unsecured (0.026)
credit (0.024)
home (0.022)

honest topic 4	honest topic 10
club (0.035)	music (0.022)
team (0.012)	band (0.012)
league (0.009)	film (0.011)
win (0.009)	festival (0.009)

Excerpt: 20 spam and 50 honest topic models
[Bíró, Szabó, Benczúr 2008]

Parking Domain (may still have old inlinks)



Keyword stuffing, generated copies

wrjk.frinzezz.net

belmajdoub

– From "Seductions of Rice" by Jeffrey Alford and Naomi Duguid (Artisan, $24. Als erste 32 GB Karte wird sie dabei der Class 6 Geschwindigkeitsspezifikation genügen, die eine minimale Datenübertragungsrate von sechs MB/s bei einer leeren Karte vorsieht. It's pronounced incorrectly sometimes, but they know me. The Cospicua school has decided to use the Belgian and Scottish schools' approaches, which are entitled The Achievement Wall' and 'The Box of Feelings'. "It's more of the smaller stuff. I think it would be wise to not get in knee deep with ideas and plans once I have everything, in every room, cleaned and organized. In the turbulent days preceding the Spanish civil war, Lorca, who was living in Madrid, was uncertain whether or not to return home to Granada as he did each summer, unclear where he would be safest in the event of a Nationalist coup. "If it's a significant customer we can go quite upmarket - when you go down the bespoke route, it can be almost anything. 4 ranked Lady Mustangs (12 3, 2 1) beat Northside in three of the four meetings between the two last season. No wonder the Sena has asked BPOs across the city for details of security measures taken for female staff during night. "Will

article

- bon jovi crush t
- megaupload
- biphosphonates
- descargar soluci
- tanenbaum
- carla giraldo cor
- posturas sexuale
- epileren touw
- construccion del
- tlalnepantla
- feuerwehr gising
- termine
- concepto de pte
- configuracion pa

Google ads

admin-to-go.co.uk
Office and secretarial services

Welcome back! Friday 25 April 2008

Looking for office and secretarial services?
Compare companies and solutions here

The following companies may be of interest to you . . .

1. Next Home Collection
Collection of Homeware at Next. Next day delivery and free returns.
next.co.uk

2. Shopping
Looking for discount vouchers codes? Discount Code has 100's of free to use promo codes, discount codes and voucher code for many UK online shops. Get you voucher codes now.
www.discountcodes.co.uk

3. Home Shopping
Huge Range of Items From Top Brands Order Online & Get Free Delivery.
www.empirestores.co.uk

4. Additions Direct
All the latest fashion delivered to your door the next day for £3.134.
www.additionsdirect.co.uk

5. Cheap Products - UK

admin-to-go.co.uk

Other suggested searches . . .

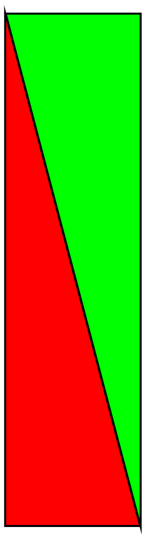
> Car Hire Company
> Four W

Web Spam Taxonomy 2.

Link spam

Hyperlinks: Good, Bad, Ugly

“hyperlink structure contains an enormous amount of latent human annotation that can be extremely valuable for automatically inferring notions of authority.” (Chakrabarti et. al. ’99)

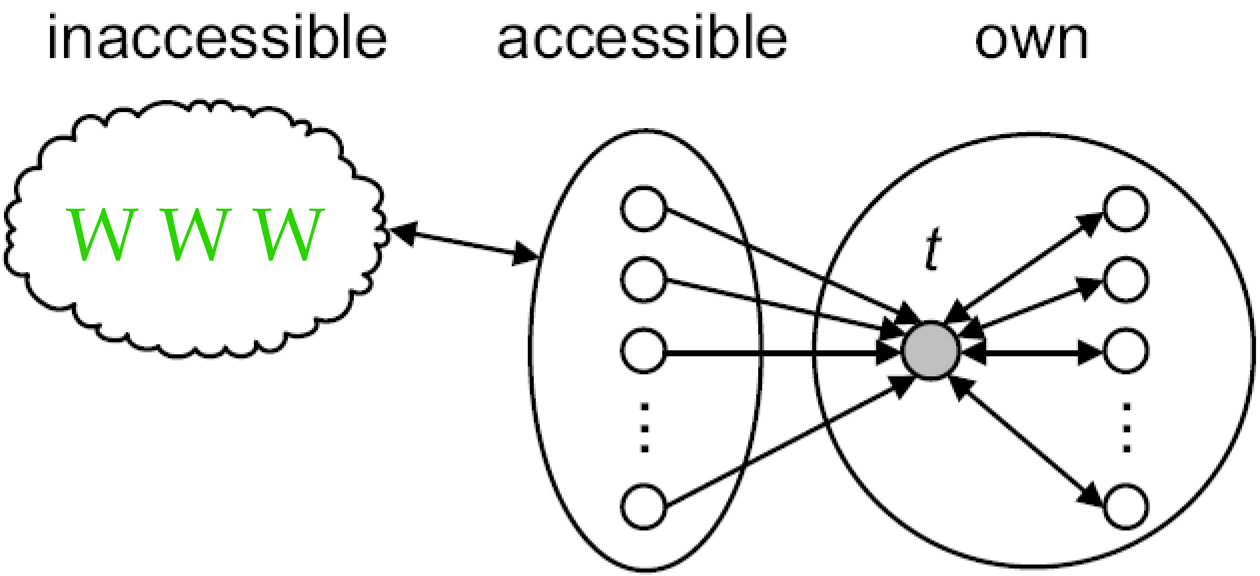


- **Honest link, human annotation**
- **No value of recommendation**, *e.g. „affiliate programs“, navigation, ads …*
- **Deliberate manipulation, link spam**

Link farms

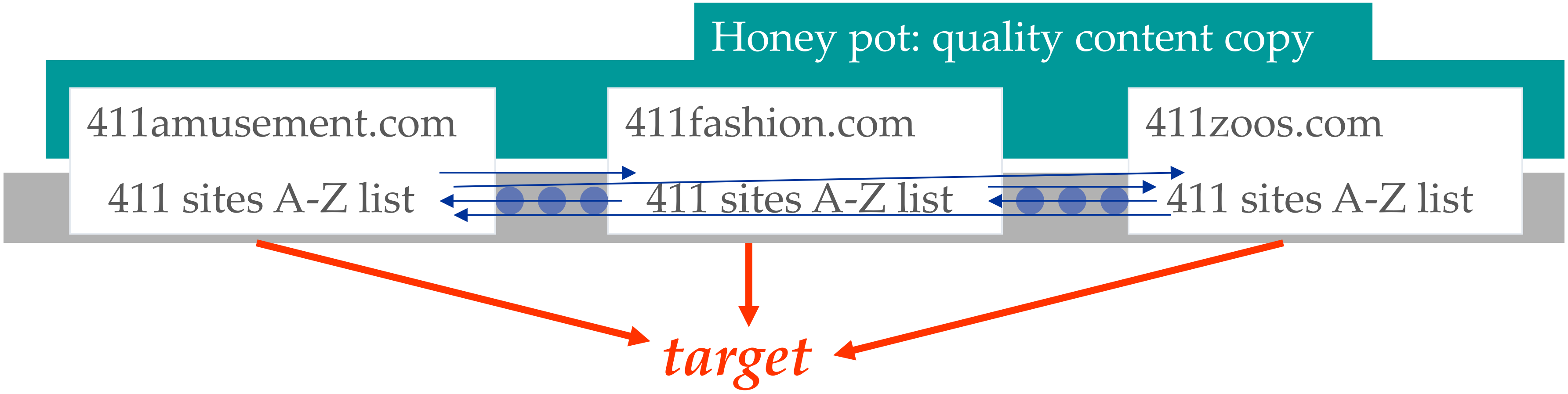
Entry point from honest web:

- Honey pots: copies of quality content
- Dead links to parking domain
- Blog or guestbook comment spam

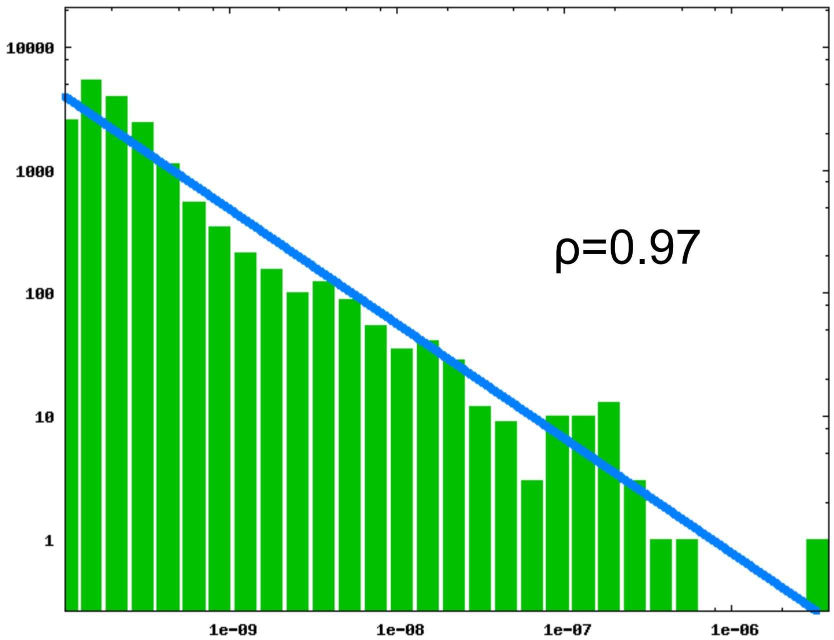


Link farms

Multi-domain, Multi-IP



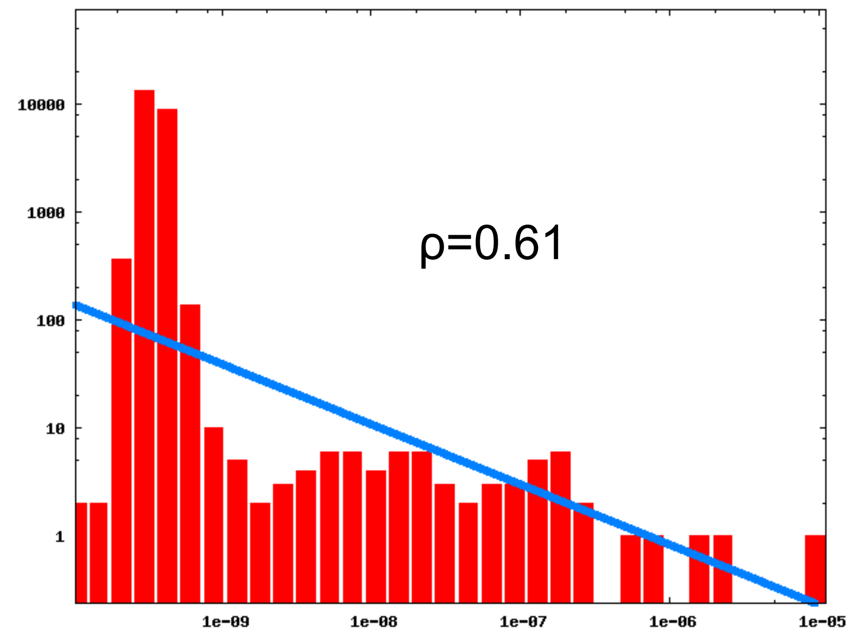
PageRank supporter distribution



low high

PageRank

Honest:
fhh.hamburg.de



low high

PageRank

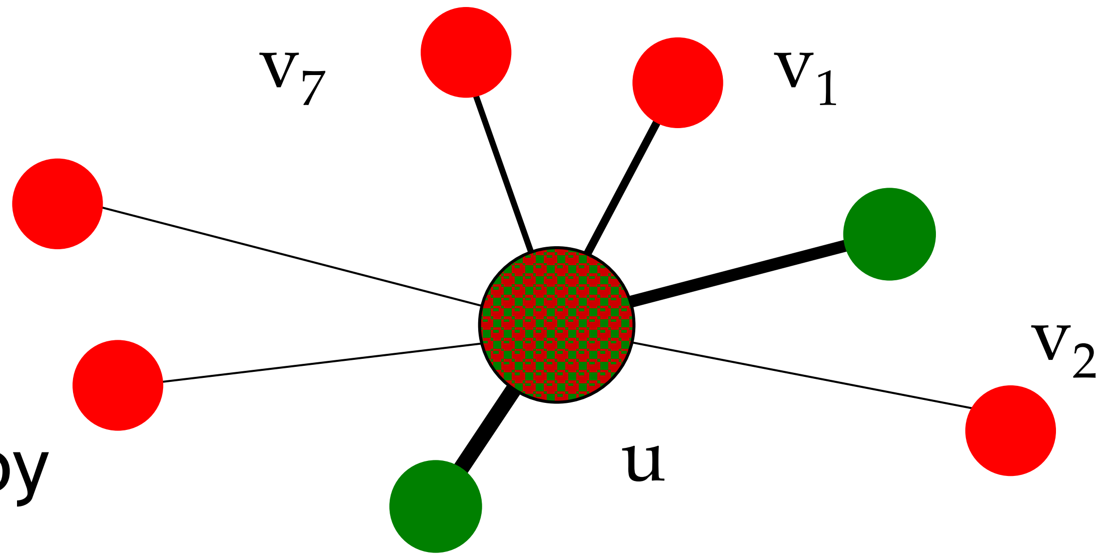
Spam: radiopr.bildflirt.de
(part of www.popdata.de farm)

[Benczúr,Csalogány,Sarlós,Uher 2005]

Know your neighbor [Debora, Chato et al 2006]

- Honest pages rarely point to spam
- Spam cites many, many spam

1. Predicted spamicity p(v) for all pages
2. Target page u, new feature f(u) by neighbor p(v) aggregation
3. Reclassification by adding the new feature



Web Spam Taxonomy 3.

Cloaking and hiding

Formatting

- One-pixel image

```
<a href="target.html"><img src="tinyimg.gif"></a>
```

- White over white

```
<body background="white">
  <font color="white">hidden text</font>
  ...
</body>
```

- Color, position from stylesheet
- ...

Idea: crawlers do simplified HTML processing

Importance for crawlers to run rendering and script execution!

Obfuscated JavaScript

```
<SCRIPT language=javascript>
  var1=100;var3=200;var2=var1 + var3;
  var4=var1;var5=var4 + var3;
  if(var2==var5)
  document.location="http://umlander.info/
  mega/free software downloads.html";
</SCRIPT>
```

- Redirection through window.location
- eval: spam content (text, link) from random looking static data
- document.write

HTTP level cloaking

- User agent, client host filtering

GET /db_pages/members.html HTTP/1.0
Host: www-db.stanford.edu
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

- Different for users and for GoogleBot
- „Collaboration service” of spammers for crawler IPs, agents and behavior

Web Spam Taxonomy 4.

Spam in social media

More recent target: blogs, guest books

Гостевая Книга Guestbook

Спасибо, что посетили мою страницу. Вы можете оставить запись в моей Гостевой Книге.
Thank you for visiting our pages. We would love it if you would Add.

Enjoyed your website and found it informative.[url=http://nazar.onlyhot.info/russell-grant-horoscope/]russell grant horoscope[/url]
John en Lia Maan <buka_sm@yahoo.com>Miaimi , USA - Monday, April 3, 2006 at 21:34:58

---

phentermine
hydrocodone
xanax

xanax <@size>Москва, Россия - Monday, April 3, 2006 at 21:17:19

Enjoyed your website and found it informative.[url=http://meds.onlyhot.info/russell-grant-horoscope/]russell grant horoscope[/url]
Rosina May <sigmroni@hotmail.com>Denver, USA - Monday, April 3, 2006 at 20:37:47

I like it because is very useful.[url=http://top.onlyhot.info/russell-grant-horoscope/]russell grant horoscope[/url]
Jurg Bollinger <annelies.hesp@wanadoo.nl>Memphis, USA - Monday, April 3, 2006 at 19:56:12

Thank you for your site. I have found here much useful information...

hoodia patchBoston, USA - Monday, April 3, 2006 at 19:30:34

---

uggs
phentermine
cialis
carisoprodol
fioricet
ambien

Fake blogs

Political Concepts

A Working Paper Series of the Committee on Concepts and Method

Working Paper

Svend-Erik Skaaning, "Measuring Civil Liberty"

April 2008

Comments

viagra doses prices com net org

21 April 2008

Nice site. Thank you!! viagra doses prices com net org

Lane

21 April 2008

Well done! roulette games online | fun play slots | no download online free slots | free play online no deposit bonus | cleopatra slot | online slot game | free slot machines to play onl line slot machine

Spam Hunting

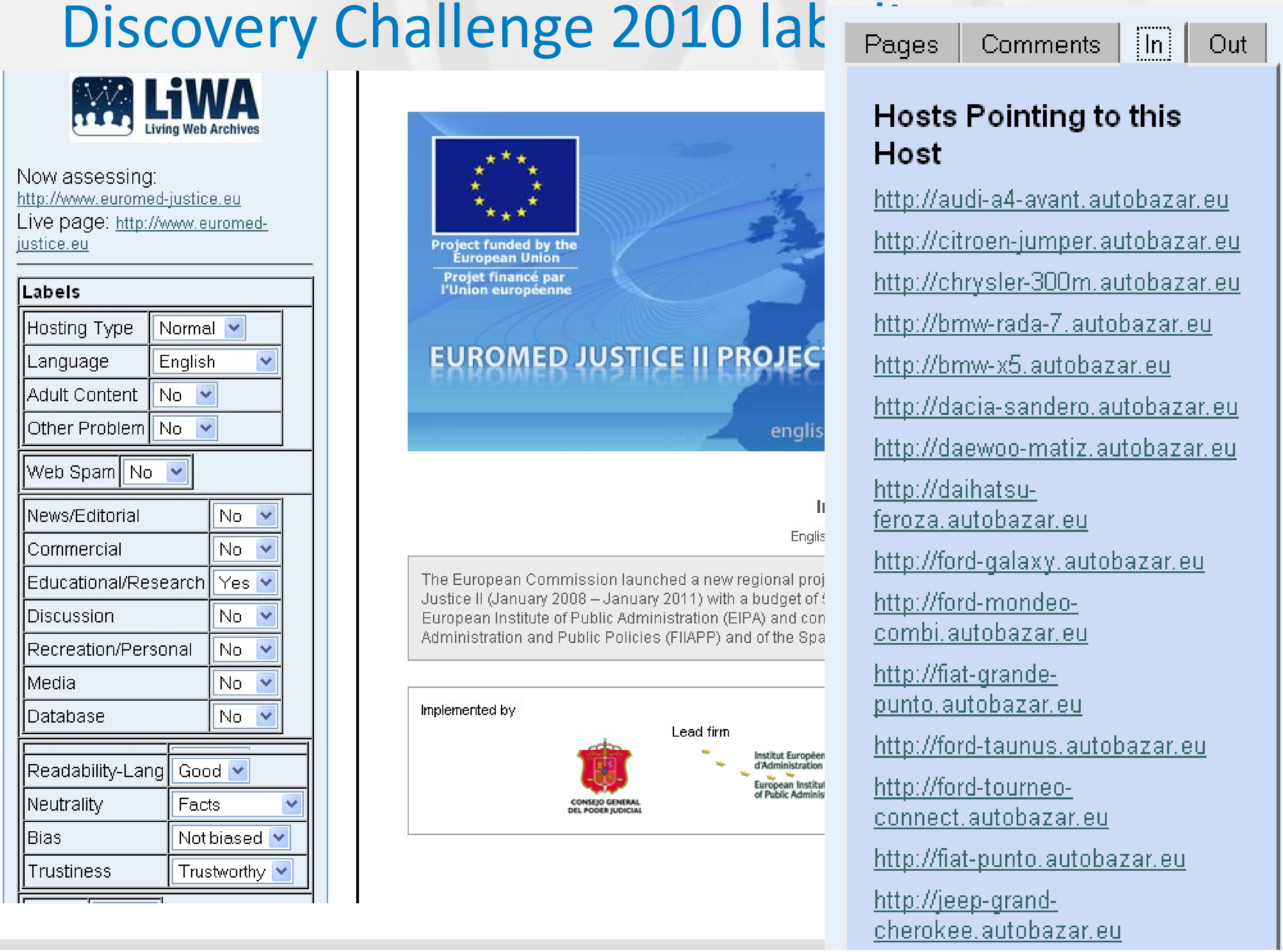
- Machine learning
- Manual labeling
- Crawl time?
- Benchmarks

No free lunch: no fully automatic filtering

- Manual labels (black AND white lists) primarily determine quality
- Can blacklist only a tiny fraction
  - Recall 10% of sites are spam
  - Needs machine learning
- Models quickly decay

  Measurement: training on intersection with WEBSPAM-UK2006 labels, test WEBSPAM-UK2007

Discovery Challenge 2010 lab



Crawl-time vs. post-processing

- Simple filters in crawler
  - cannot handle unseen sites
  - needs large bootstrap crawl
- Crawl time feature generation and classification
  - Needs interface in crawler to access content
  - Needs model from external crawl (may be smaller)
  - Sounds expensive but needs to be done only once per site

Web Spam and Quality Challenges

- UK-WEBSPAM2006 [Debora, Chato]
  - 9000 Web sites, 500,000 links
  - 767 spam, 7472 nonspam
- UK-WEBSPAM2007 [Debora, Chato]
  - 114,000 Web sites, 3 bio links
  - 222 spam, 3776 nonspam
  - 3 TByte full uncompressed data
- ECML/PKDD Discovery Challenge 2010 [Andras, Chato]
  - 190,000 Web sites, 430 spam, 5000 nonspam
  - Also trust, neutrality, bias
- The Reconcile project C3 data set (WebQuality 2015 data)
  - 22 325 Web page evaluations, scale: 0 – 4; 5 for missing
  - credibility, presentation, knowledge, intentions, completeness
  - 5704 pages by 2499 assessors

Machine Learning

- Originally, many features of linkage and content processing
- Worked because spam farms were cut into training and testing
- Recently, we realized only terms are needed
  - TF, TF-IDF, BM25
  - Distance: Jensen-Shannon or Euclidean (L2)
  - Support Vector Machines

    (a new similarity kernel worked very well)
    - Advantage: the prediction model is just a set of vectors and inner products need to be computed
  - See our results over the C3 data set (2015)

	All non-term	TF		TFIDF		BM25			BM25 + nonterm	All
		J-S	L2	J-S	L2	J-S	L2	+		
AUC	.66	.70	.65	.70	.66	.67	.71	.72	.73	.73

PageRank

Hyperlink analysis: Goals

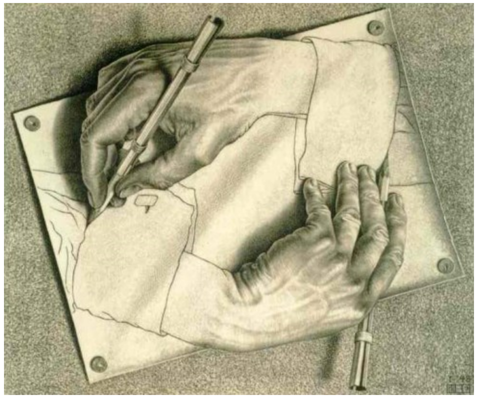
- Ranking, PageRank

  … well that is obvious?
- Features for network classification
- Propagation, Markov Random Fields
- Centrality

  … PageRank why central?
- Similarity of graph nodes

PageRank as Quality

A quality page is pointed to by several quality pages



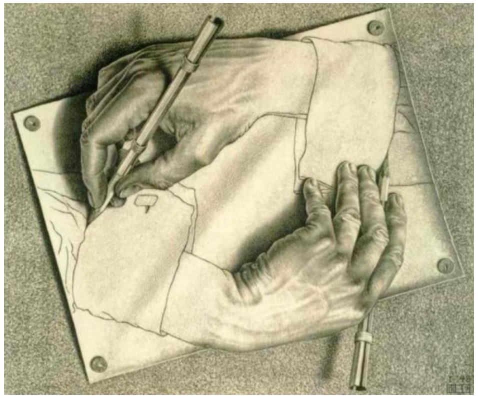
"hyperlink structure contains an enormous amount of latent human annotation that can be extremely valuable for automatically inferring notions of authority." (Chakrabarti et. al. '99)

NB: not all links are useful, quality, …

The Good, the Bad and the Ugly

PageRank as Quality

A quality page is pointed to by several quality pages



$$\cancel{\mathbf{PR}^{(k+1)} = \mathbf{PR}^{(k)}}\ \mathbf{M}$$

$$\mathbf{PR}^{(k+1)} = \mathbf{PR}^{(k)} \left((1 - \varepsilon)\, \mathbf{M} + \varepsilon \cdot \mathbf{U} \right)$$

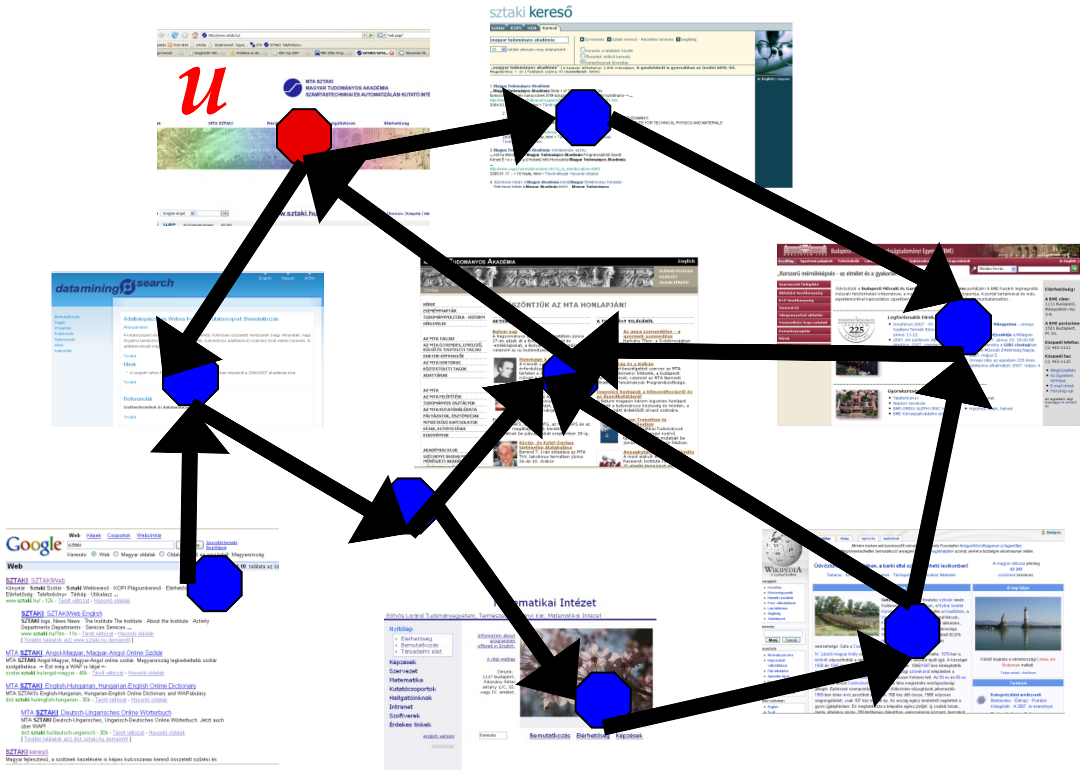
$$= \mathbf{PR}^{(1)} \left((1 - \varepsilon)\, \mathbf{M} + \varepsilon \cdot \mathbf{U} \right)^{k}$$

U could represent jump to any fixed (*personalized*) distribution

Brin, Page 98

The Random Surfer Model

Starts at a random page—arrives at quality page

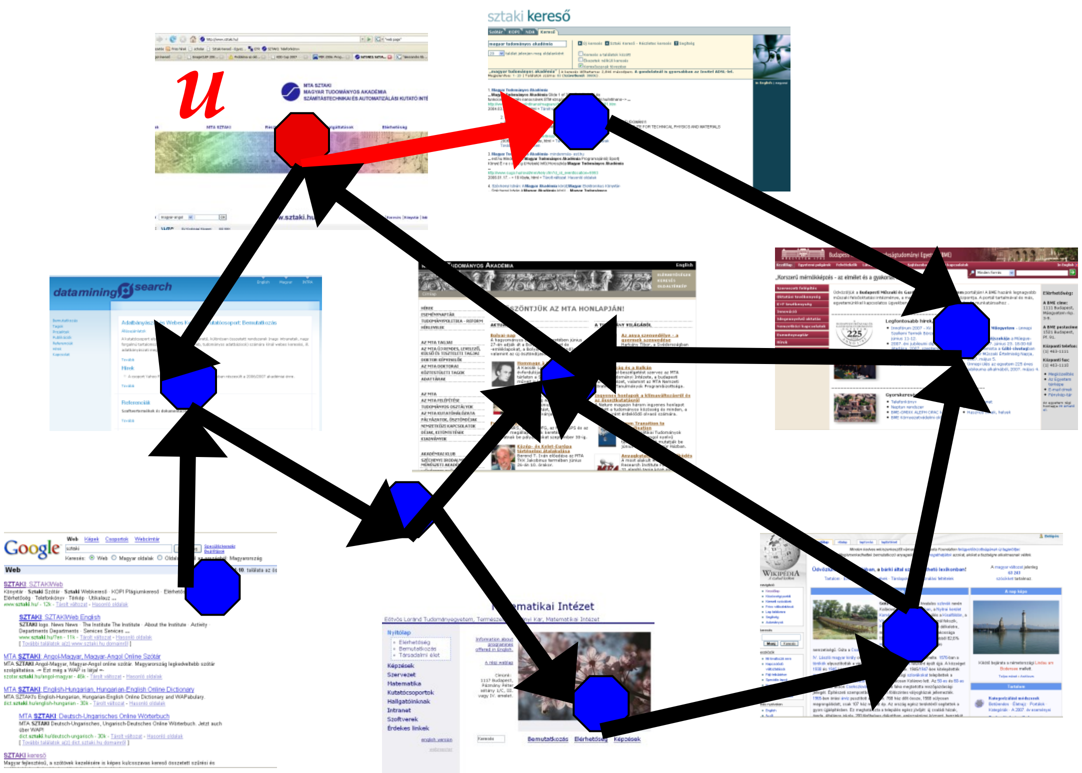


Nodes = Web pages

Edges = hyperlinks

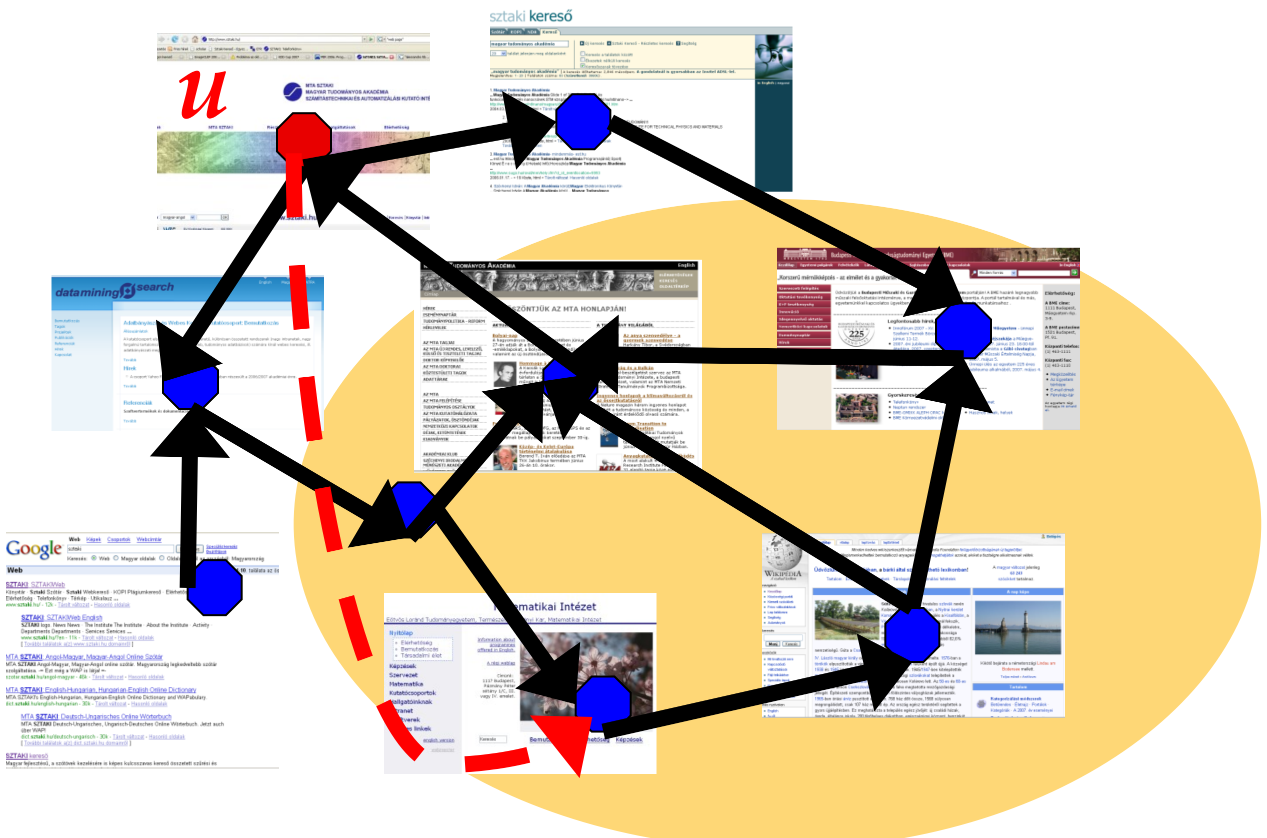
The Random Surfer Model

Chooses random neighbor with probability 1- ε



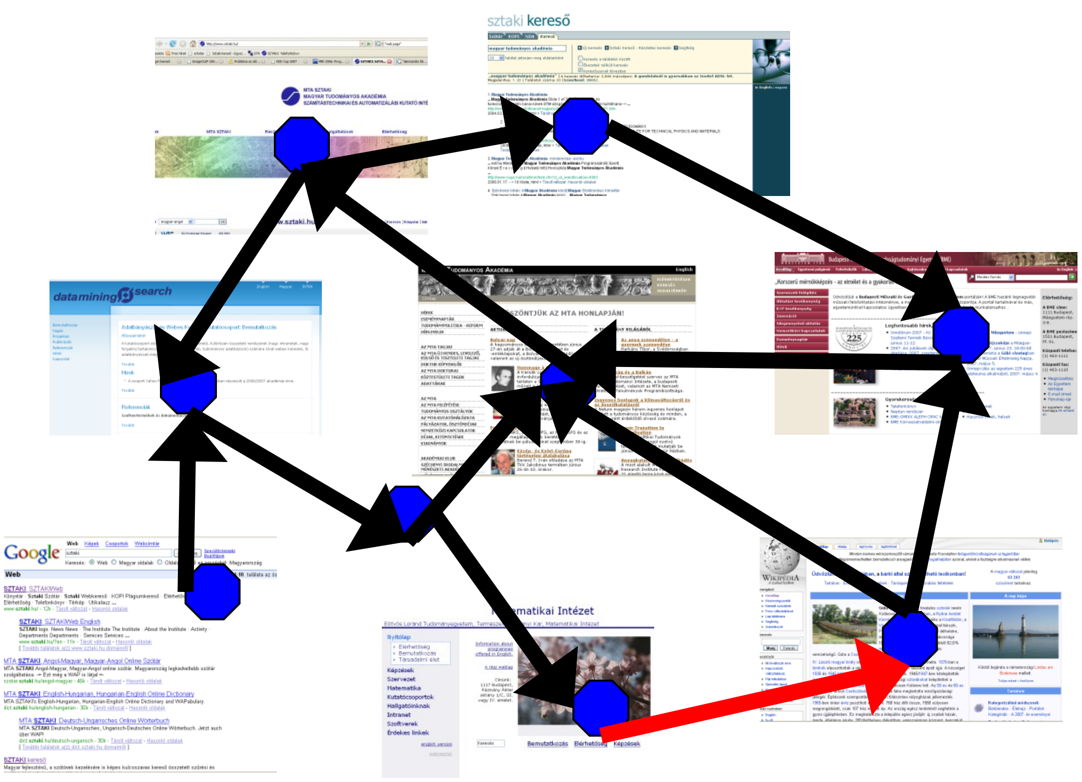
The Random Surfer Model

Or with probability ε "teleports" to random (personalized) page—gets bored and types a new URL or chooses a random bookmark



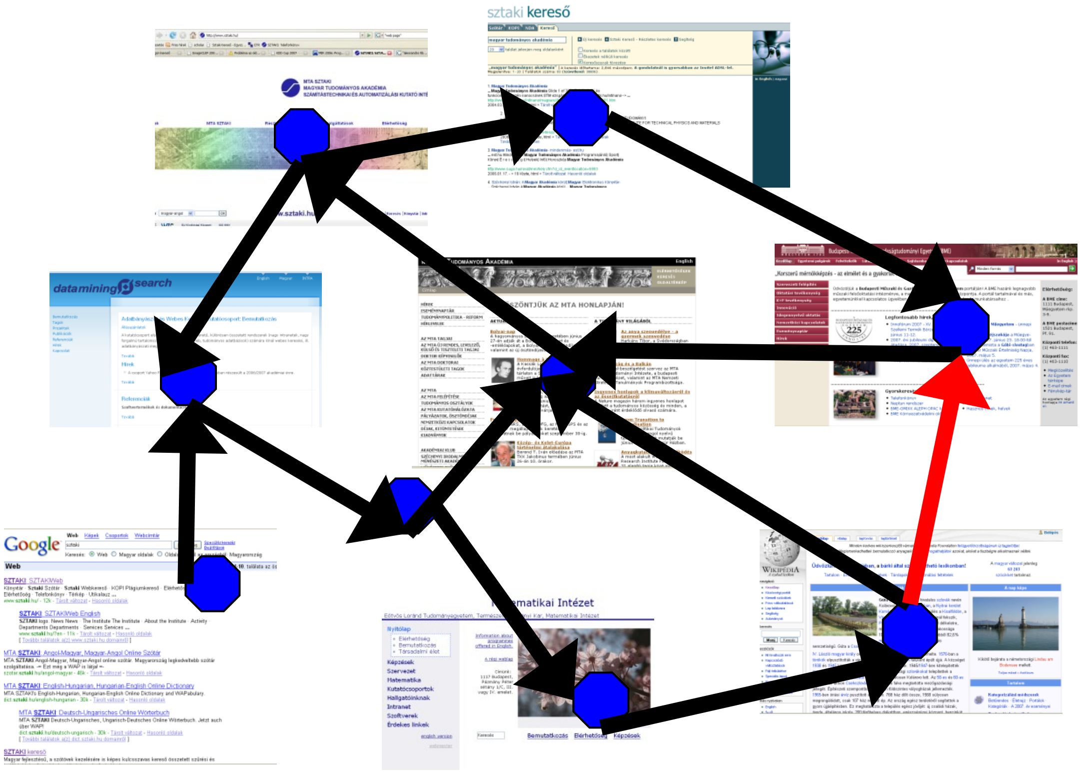
The Random Surfer Model

And continues with the random walk …



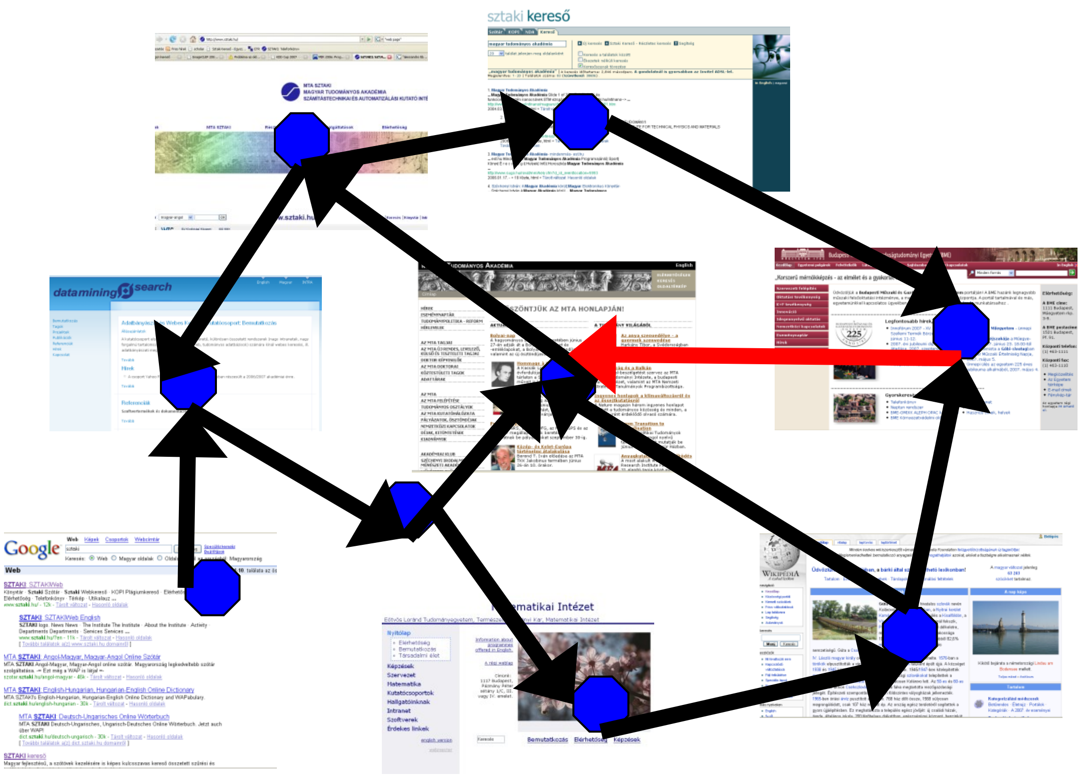
The Random Surfer Model

And continues with the random walk …



The Random Surfer Model

Until convergence … ?

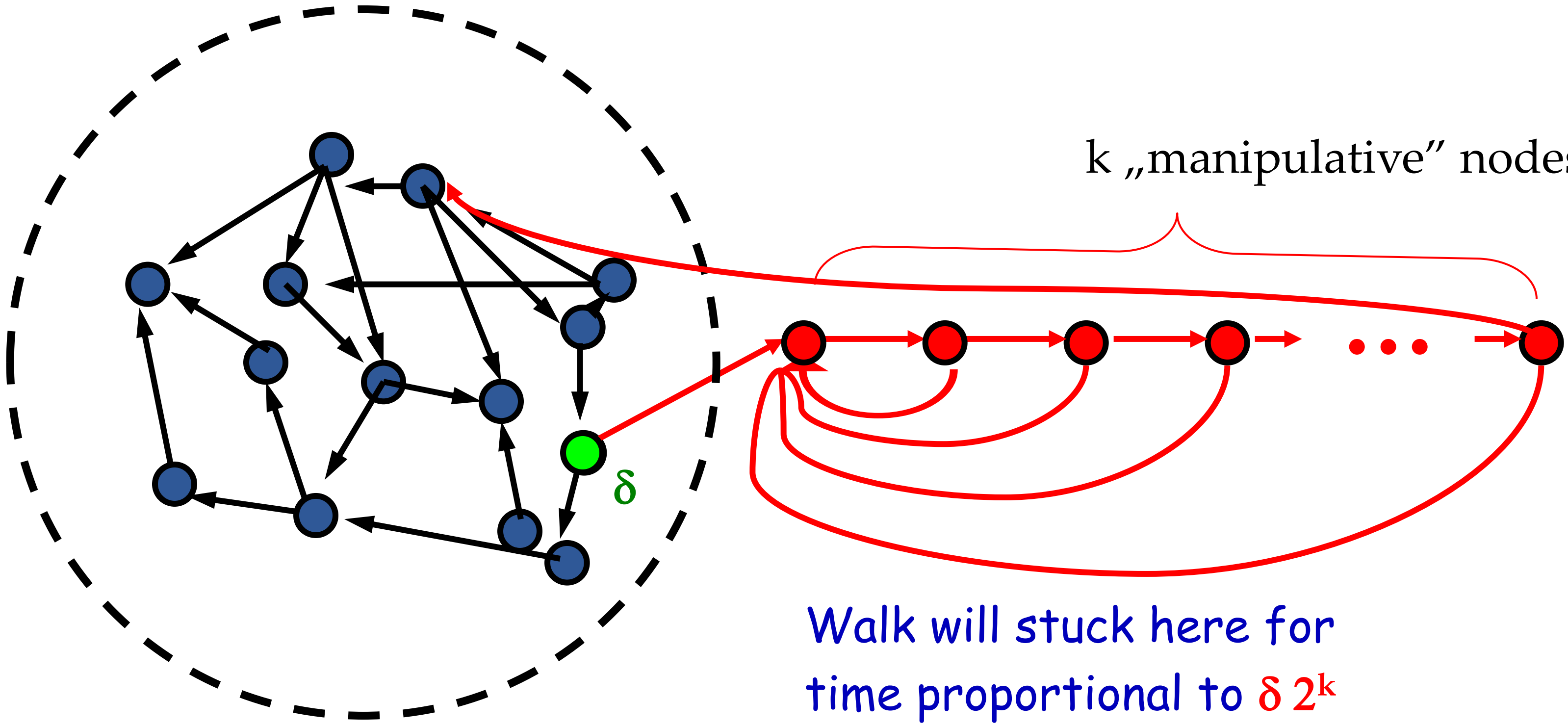


[Brin, Page 98]

Teleportation – less obvious reasons

Assume PageRank is $\delta > 0$

fraction δ of time spent here



Walk will stuck here for

time proportional to $\delta\, 2^k$

Exponential gain of the manipulator

PageRank as a Big Data problem

- Estimated 10+ billions of Web pages worldwide
- PageRank (as floats)
  - fits into 40GB storage
- Personalization just to single pages:
  - 10 billions of PageRank scores for each page
  - Storage exceeds several Exabytes!

- NB single-page personalization is enough:

$$\mathbf{PPR}(\alpha_1\mathbf{v}_1+\ldots+\alpha_k\mathbf{v}_k)=\alpha_1\mathbf{PPR}(\mathbf{v}_1)+\ldots+\alpha_k\mathbf{PPR}(\mathbf{v}_k)$$

For certain things are just too big?

- For light to reach the other side of the Galaxy … takes rather longer: five hundred thousand years.
- The record for hitch hiking this distance is just under five years, but you don't get to see much on the way.

D Adams, **The Hitchhiker's Guide to the Galaxy. 1979**

Equivalence with short walks

Jeh, Widom '03, Fogaras '03

- Random walk starts from distribution (or page) *u*
- Follows random outlink with probability *1-ε,* stops with *ε*
- PPR*(u,v)=Pr{* the walk from *u* **stops** at page *v }*

$$\mathbf{PR}^{(1)} \left((1 - \varepsilon)\, \mathbf{M} + \varepsilon \cdot \mathbf{U} \right)^k =$$

$$\mathbf{u} \sum_{i=0}^{k-1} \varepsilon(1 - \varepsilon)^i \;\; \mathbf{M}^i + \mathbf{PR}^{(1)} (1 - \varepsilon)^k\, \mathbf{M}^k$$

paths of length i

Terminate with probability ε

Continue with probability (1 - ε)

Stop!

Appreciate the simplicity

- Few lines completely elementary proof
- Convergence follows w/o any theory
- Convergence speed follows (eigengap)
- Meaning: centrality through short walks
- Solves algorithmics (to come)

Monte Carlo Personalized PageRank

- Markov Chain Monte Carlo algorithm
- Pre-computation
  - From *u* simulate *N* independent random walks
  - Database of fingerprints: ending vertices of the walks from **all** vertices
- Query
  - PPR(u,v) := # (walks $u \to v$) / N
  - N ≈ 1000 approximates top 100 well
- Fingerprinting techniques

Fogaras-Racz: Towards Scaling Fully Personalized PageRank

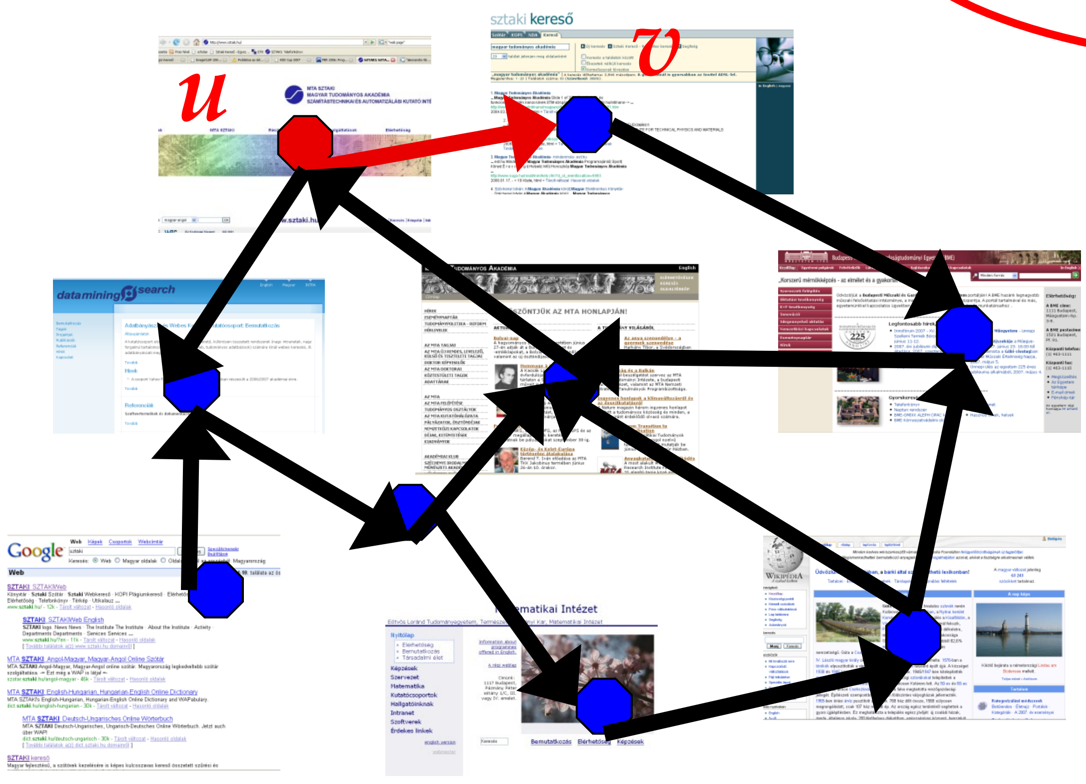
Semi-Supervised Learning

- Idea: Objects in a network are similar to neighbors
  - Web: links between similar content; neighbors of spam are likely spam
  - Telco: contacts of churned more likely to churn
  - Friendship, trust
- Implementations:
  - Stacked graphical learning [Cohen, Kou 2007]
  - Propagation [Zhou et al, NIPS 2003]

$$pred^{(t+1)}(v) = \varepsilon \cdot pred(v) + (1-\varepsilon) \cdot \sum M_{uv} pred^{(t)}(u)$$

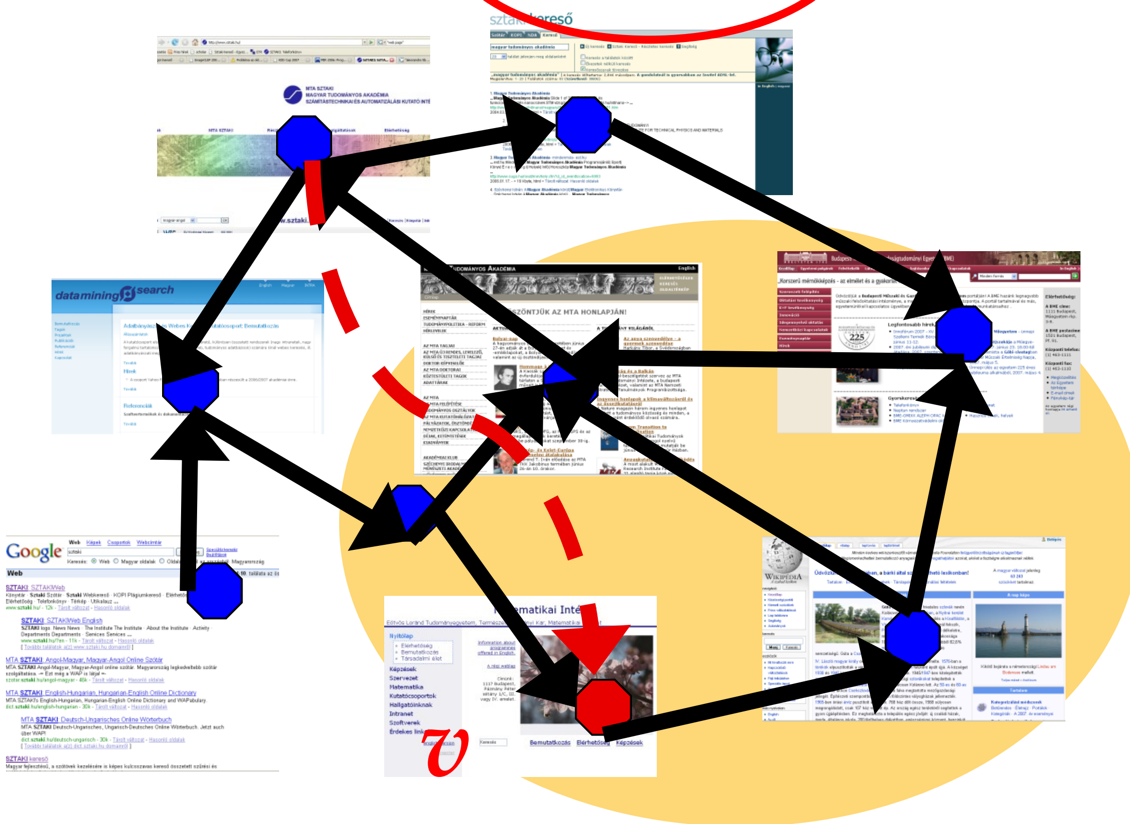
Random link with probability 1- ε

$$pred^{(t+1)}(v) = \varepsilon \cdot pred(v) + (1-\varepsilon) \cdot \sum M_{uv} pred^{(t)}(u)$$



Personalized teleport with prob ε

$$pred^{(t+1)}(v) = \varepsilon \cdot pred(v) + (1-\varepsilon) \cdot \sum M_{uv} pred^{(t)}(u)$$



Other uses – mostly for spam hunting

- Google BadRank
- TrustRank: personalized on quality seed [Gyongyi,Garcia-Molina 2005]
- SpamRank: statistics of short incoming walks [B,Csalogany,Sarlos,Uher 2005]
- Truncated PageRank versions, neighborhood features, ratios, host level statistics [Castillo et al, 2006]

Distributed data processing

Google MapReduce for large scale inverted index build

Distributed sotfware systems and their limitations

Hadoop

PageRank by Hadoop

PageRank by other systems: Flink, GraphLab

- Google’s computational/data manipulation model
- Elegant way to work with big data

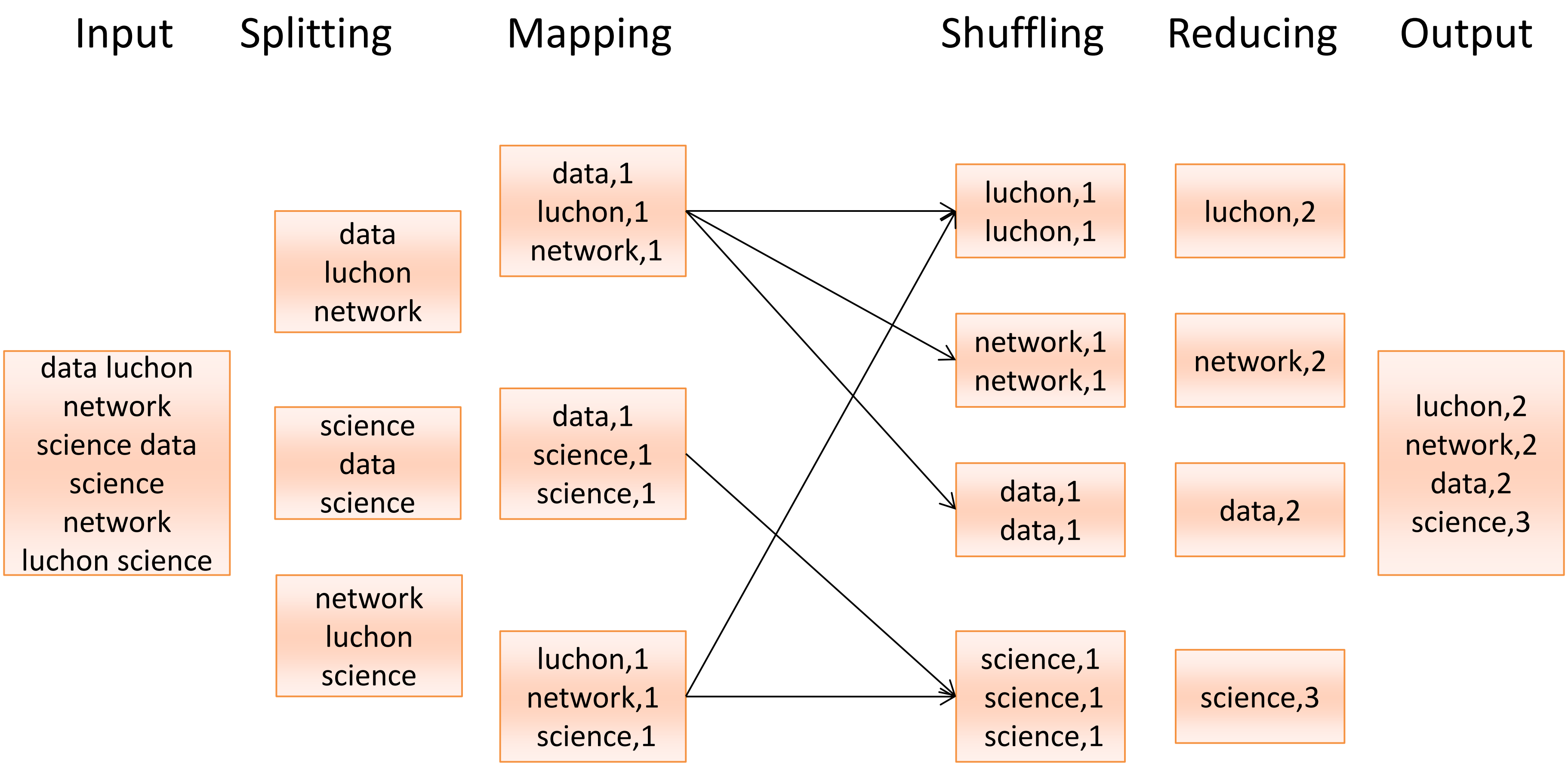
Computational Model: MapReduce

Motivation: Google Example

- 20+ billion web pages x 20KB = 400+ TB
- 1 computer reads 30-35 MB/sec from disk
  - ~4 months to read the web
- ~1,000 hard drives to store the web
- Takes even more to **do** something useful with the data!
- **Recently standard architecture for such problems emerged:**
  - Cluster of commodity Linux nodes
  - Commodity network (ethernet) to connect them

Search Index Build Google scale

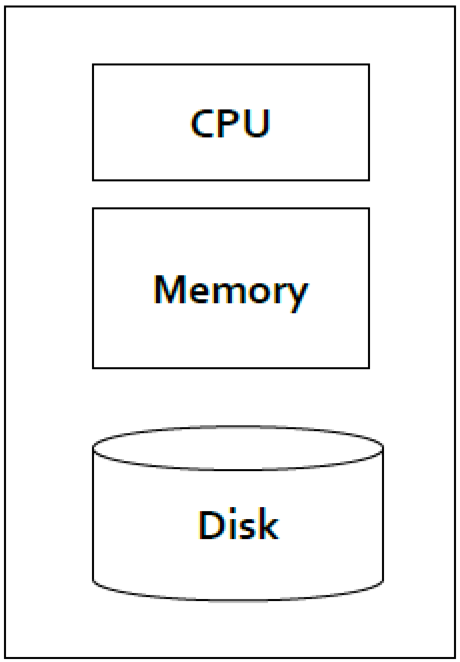
Map – Shuffle/Sort – Reduce



Hello World for different systems

- Java, ...
  - Print "Hello World"
- MapReduce
  - Word count
- Graphs
  - PageRank or connected components (suprise: they are almost the same)

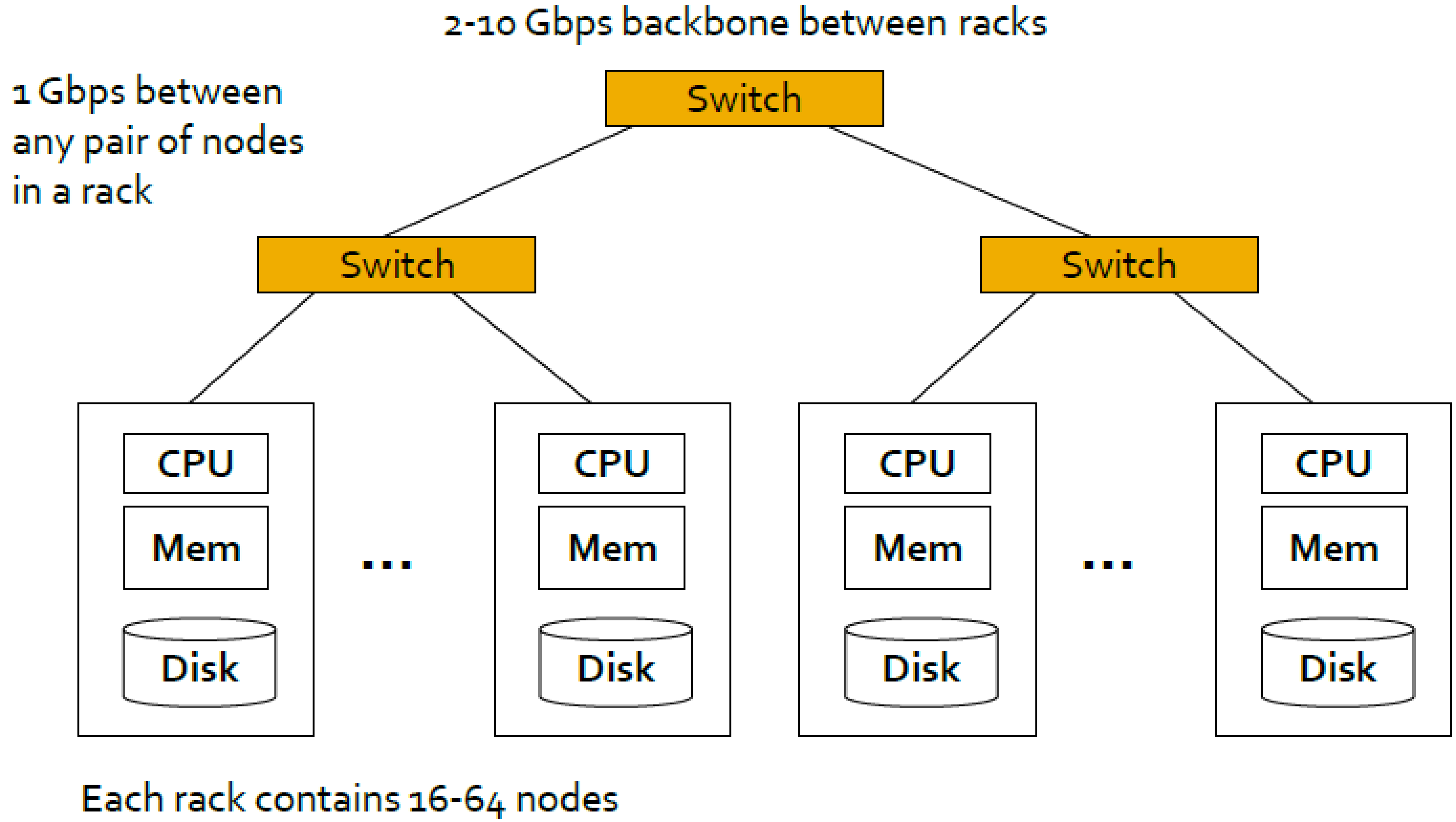
Single Node Architecture



Machine Learning, Statistics

"Classical" Data Mining

Cluster Architecture

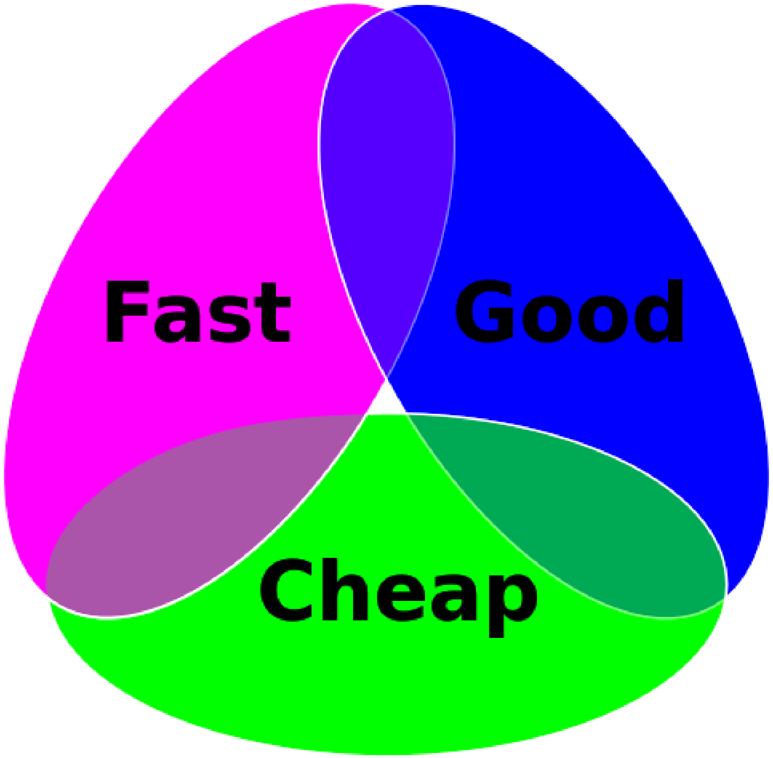


In 2011 it was guestimated that Google had 1M machines, http://bit.ly/Shh0RO

Large-scale Computing

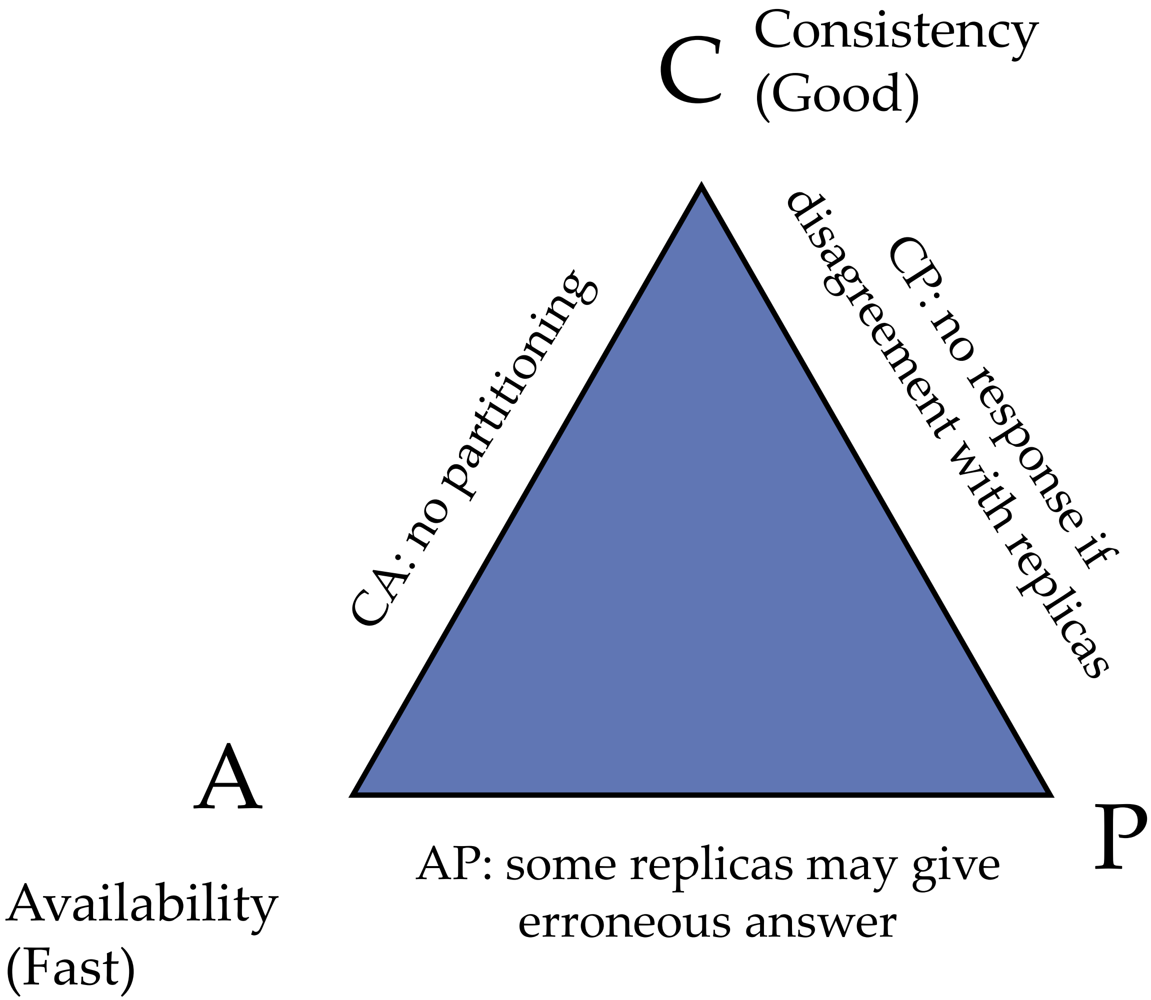
- **Large-scale computing** for **data mining** problems on **commodity hardware**
- **Challenges:**
  - **How do you distribute computation?**
  - **How can we make it easy to write distributed programs?**
  - **Machines fail:**
    - One server may stay up 3 years (1,000 days)
    - If you have 1,000 servers, expect to loose 1/day
    - With 1M machines 1,000 machines fail every day!

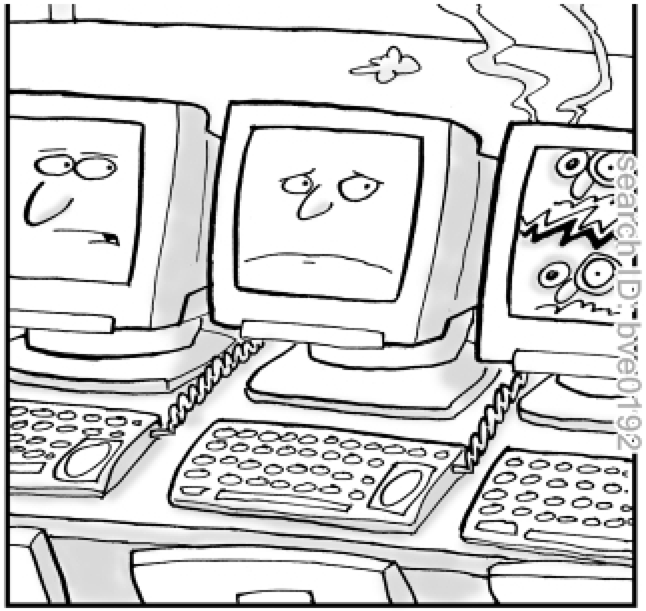
The Project Triangle



CAP (Fox&Brewer) Theorem

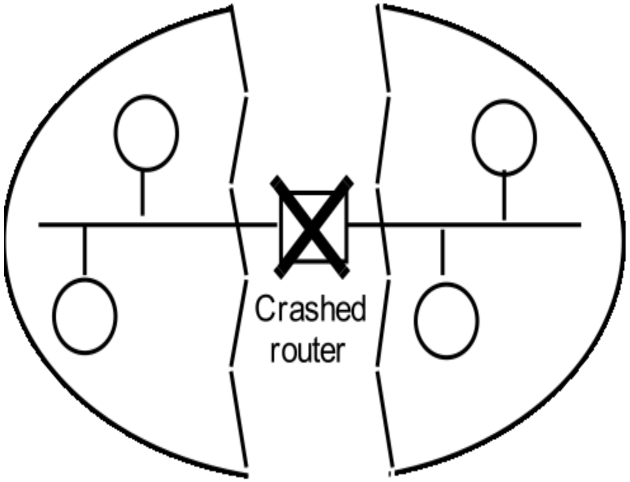
Theorem: You may choose two of C-A-P





Fox&Brewer proof

- Partition (P): LHS will not know about new data on RHS
- Immediate response from LHS (availability) may give incorrect answer
- If partition (P), then either availability (A) or consistence (C)



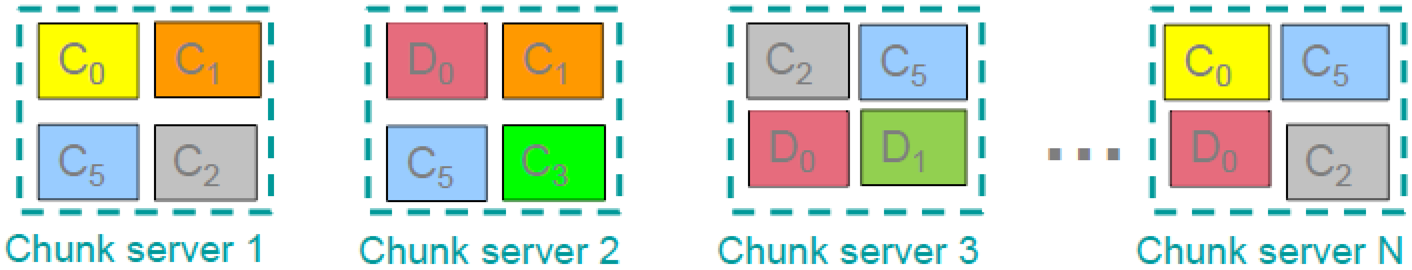
- Eventual consistency if connection resumes and data can be exchanged
- MapReduce is PC – batch computations, restarts in case of failures

Hadoop overview

- Machines WILL fail
- Data needs to be partitioned and REPLICATED
  - File system: Google, Hadoop file systems – HDFS
  - NameNode to store the lookup for chunks
- Copying over the network is slow
  - Bring computation close to the data
  - Let a Master Node be responsible for
    - Task sheduling, failure detection
    - Managing and transmitting temporary output files
- MapReduce computations
  - We’ll se what it can and what it cannot really do well

Distributed File System

- **Reliable distributed file system**
- Data kept in "chunks" spread across machines
- Each chunk **replicated** on different machines
  - Seamless recovery from disk or machine failure



Bring computation directly to the data!

Chunk servers also serve as compute servers

Hadoop NameNode localhost:8020 - Mozilla Firefox

localhost Hadoop Map/Red... | Hadoop NameNode localh...

localhost:50070/dfshealth.jsp

NameNode 'localhost:8020'

Started:	Thu Nov 06 11:46:11 CET 2014
Version:	1.2.1, r1503152
Compiled:	Mon Jul 22 15:23:09 PDT 2013 by mattf
Upgrades:	There are no upgrades in progress.

Browse the filesystem
Namenode Logs

Cluster Summary

28 files and directories, 7 blocks = 35 total. Heap Size is 59.69 MB / 966.69 MB (6%)

Configured Capacity	:	4.89 GB
DFS Used	:	424 KB
Non DFS Used	:	4.1 GB
DFS Remaining	:	812.8 MB
DFS Used%	:	0.01 %
DFS Remaining%	:	16.23 %
Live Nodes	:	1
Dead Nodes	:	0
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	0

NameNode Storage:

Storage Directory	Type	State
/data/hdfs/name	IMAGE_AND_EDITS	Active

This is Apache Hadoop release 1.2.1

Accessing the HDFS filesystem

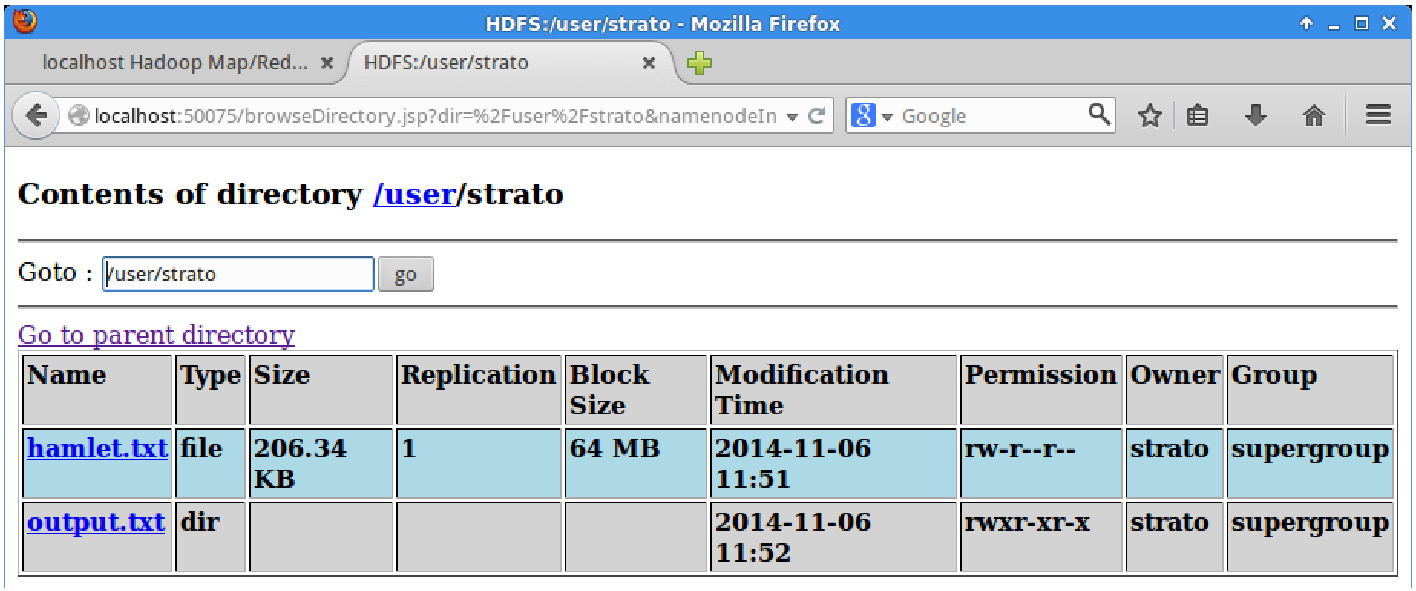
Java library

- Copy from/to local, e.g.:

  hadoop dfs -put localfile hdfsfile

- Standard file manipulation commands, e.g.:

  hadoop dfs -ls (-rm, -mkdir, …)



Contents of directory /user/strato

Goto : /user/strato go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
hamlet.txt	file	206.34 KB	1	64 MB	2014-11-06 11:51	rw-r--r--	strato	supergroup
output.txt	dir				2014-11-06 11:52	rwxr-xr-x	strato	supergroup

WordCount: Models of Computation

- All <word, count> counters fit in memory
  - Hash tables
- External memory
  - Sort
- Streaming data?
- Distributed, many machines?

MapReduce: Overview

3 steps of MapReduce

- Sequentially read a lot of data
- **Map:**
  - Extract something you care about
- **Group by key:** Sort and shuffle
- **Reduce:**
  - Aggregate, summarize, filter or transform
- Output the result

Outline stays the same, **Map** and **Reduce** change to fit the problem

More Specifically

- **Input:** a set of key-value pairs
- **Programmer specifies two methods:**
  - **Map(k, v)** → <k', v'>*
    - Takes a key-value pair and outputs a set of key-value pairs
      - E.g., key is the filename, value is a single line in the file
    - There is one Map call for every *(k,v)* pair
  - **Reduce(k', <v'>*)** → <k', v''>*
    - **All values *v'* with same key *k'* are reduced together and processed in *v'* order**
    - There is one Reduce function call per unique key *k'*

Word Count Using MapReduce

```
map(key, value):
// key: document name; value: text of the document
  for each word w in value:
     emit(w, 1)


reduce(key, values):
// key: a word; value: an iterator over counts
     result = 0
     for each count v in values:
          result += v
     emit(key, result)
```

Word Counting: Main

```
package org.myorg;

import java.io.IOException;
import java.util.*;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class WordCount {

 public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> { … }
 public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> { … }

 public static void main(String[] args) throws Exception {
   Configuration conf = new Configuration();

       Job job = new Job(conf, "wordcount");

   job.setOutputKeyClass(Text.class);
   job.setOutputValueClass(IntWritable.class);

   job.setMapperClass(Map.class);
   job.setReducerClass(Reduce.class);

   job.setInputFormatClass(TextInputFormat.class);
   job.setOutputFormatClass(TextOutputFormat.class);

   FileInputFormat.addInputPath(job, new Path(args[0]));
   FileOutputFormat.setOutputPath(job, new Path(args[1]));

   job.waitForCompletion(true);
 }

}
```

Word Counting: Map

```
public static class Map extends Mapper<LongWritable, Text, Text,
IntWritable> {
// public class Mapper<KEYIN, VALUEIN, KEYOUT, VALUEOUT>

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(LongWritable key, Text value, Context
    context) throws IOException, InterruptedException {

        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);
        while (tokenizer.hasMoreTokens()) {
            word.set(tokenizer.nextToken());
            context.write(word, one);
        }
    }
}
```

Word Counting: Reduce

```
public static class Reduce extends Reducer<Text, IntWritable,
Text, IntWritable> {

   public void reduce(Text key, Iterable<IntWritable> values,
Context context)
     throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
         sum += val.get();
      }
      context.write(key, new IntWritable(sum));
   }
}
```

Master Node / Job tracker role

- Task status and scheduling
- Manage intermediate Mapper output to pass to Reducers
- Ping workers to detect failures
  - Restart tasks from input or intermediate data, all stored on disk
- Master node is a single point of failure

Hadoop Job Tracker

localhost:50030/jobtracker.jsp

Cluster Summary (Heap Size is 59.69 MB/966.69 MB)

Quick Links

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes	Graylisted Nodes	Exclud Node
1	0	2	1	1	0	0	0	2	2	4.00	0	0	0

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)

Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

Jobid	Started	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information	Diagnostic Info
job_201411061146_0002	Thu Nov 06 17:34:07 CET 2014	NORMAL	strato	wordcount	0.00%	1	0	0.00%	2	0	NA	NA

Completed Jobs

Jobid	Started	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information	Diagnostic Info

Algorithms over MapReduce

Join

PageRank

Warmup: MapReduce Join

A	B
a_1	b_1
a_2	b_1
a_3	b_2
a_4	b_3

R

$\bowtie$

B	C
b_2	c_1
b_2	c_2
b_3	c_3

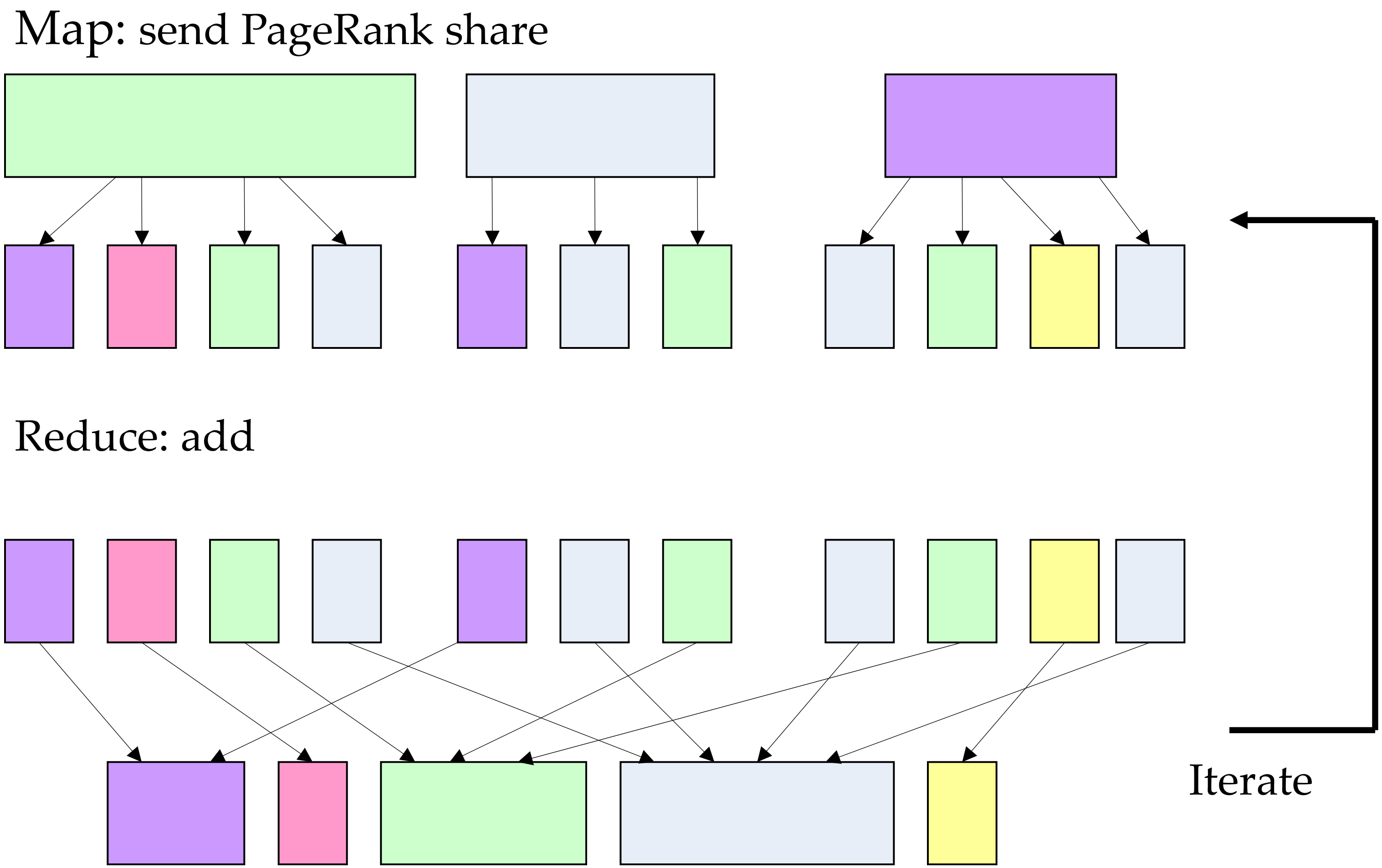
S

=

A	C
a_3	c_1
a_3	c_2
a_4	c_3

- Map:
  - R(a,b) -> key is b, value is the tuple a, ”R”
  - S(b,c) -> key is b, value is the tuple c, ”S”
- Reduce:
  - Collect all a, ”R” and c, ”S” tuples by key a to form (a,b,c)

MapReduce PageRank



...

MapReduce PageRank pseudocode

- MAP: for all nodes n
  - Input: current PageRank and out-edge list of n
  - $\forall p \in$ edgelist(n): emit (p, PageRank(n) / outdegree(n))
- Reduce
  - Obtains data ordered by p
  - Updates PageRank(p) by summing up all incoming PageRank
  - Writes to disk, starts new iteration as a new MapReduce job
- Stop updating a node if change is small; terminate if no updates
- How to start a new iteration??
  - We need both edgelist(n) and PageRank(n)
  - But they reside in completely different data sets, partitioned independently $\rightarrow$ we need a join
  - Solution: we need `emit (n, edgelist(n))` as well

MapReduce PageRank: Main

```
public static void main(String[] args) {
    String[] value = {
    // key | PageRank | points-to
            "1|0.25|2;4",
            "2|0.25|1;3;4",
            "3|0.25|2",
            "4|0.25|1;3",
    };

    mapper(value);
    reducer(collect.entrySet());
}
```

```
  | 1 2 3 4
--+----------
1 | 0 1 0 1
2 | 1 0 1 1
3 | 0 1 0 0
4 | 1 0 1 0
```

MapReduce PageRank: Reduce

```
private static void
    reducer(Set<Entry<String, ArrayList<String>>> entrySet) {
        for (Map.Entry<String, ArrayList<String>> e : entrySet) {
                Iterator<String> values = e.getValue().iterator();
                float PageRank = 0;
                String link_list = "";
                while (values.hasNext()) {
                        String[] dist_links =
                        values.next().toString().split("[|]");
                        if (dist_links.length > 1)
                                link_list = dist_links[1];
                        int inPageRank = Integer.parseInt(dist_links[0]);
                        PageRank += incomingPageRank;
                }
        System.out.println(e.getKey() + " - D " + (PageRank + " | " + link_list));
        }
    }
}
```

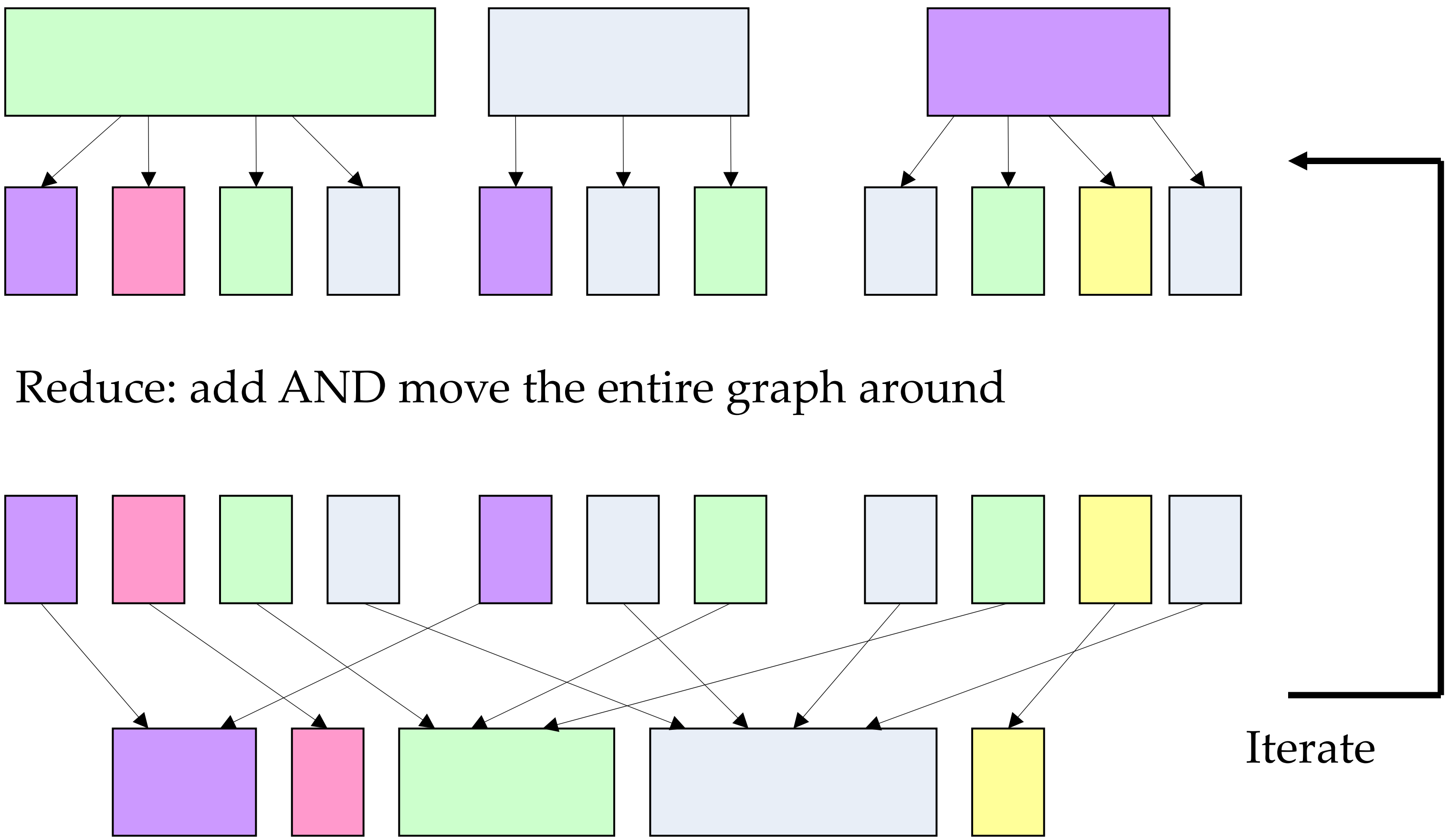
MapReduce PageRank: Map

```
private static void mapper(String[] value) {

    for (int i = 0; i < value.length; i++) {
        String line = value[i].toString();
        String[] keyVal = line.split("[|]");

        String Key = keyVal[0];
        String sDist = keyVal[1];
        String[] links = null;
        if (keyVal.length > 2) {
            links = keyVal[2].split(";");
            int Dist = Integer.parseFloat(PageRank);

            for (int x = 0; x < links.length; x++) {
            if (links[x] != "") {
                ArrayList<String> list;
                if (collect.containsKey(links[x])) {
                list = collect.get(links[x]);
                } else {
                list = new ArrayList<String>();
                }
            list.add(PageRank/ links.length + "|");
            collect.put(links[x], list);
        }
    }

ArrayList<String> list;
if (collect.containsKey(Key)) {
    list = collect.get(Key);
    } else {
    list = new ArrayList<String>();
    }
    list.add(sDist + "|" + keyVal[2]);
    collect.put(Key, list);
    }
}
```

MapReduce PageRank

Map: send PageRank share AND the entire graph!

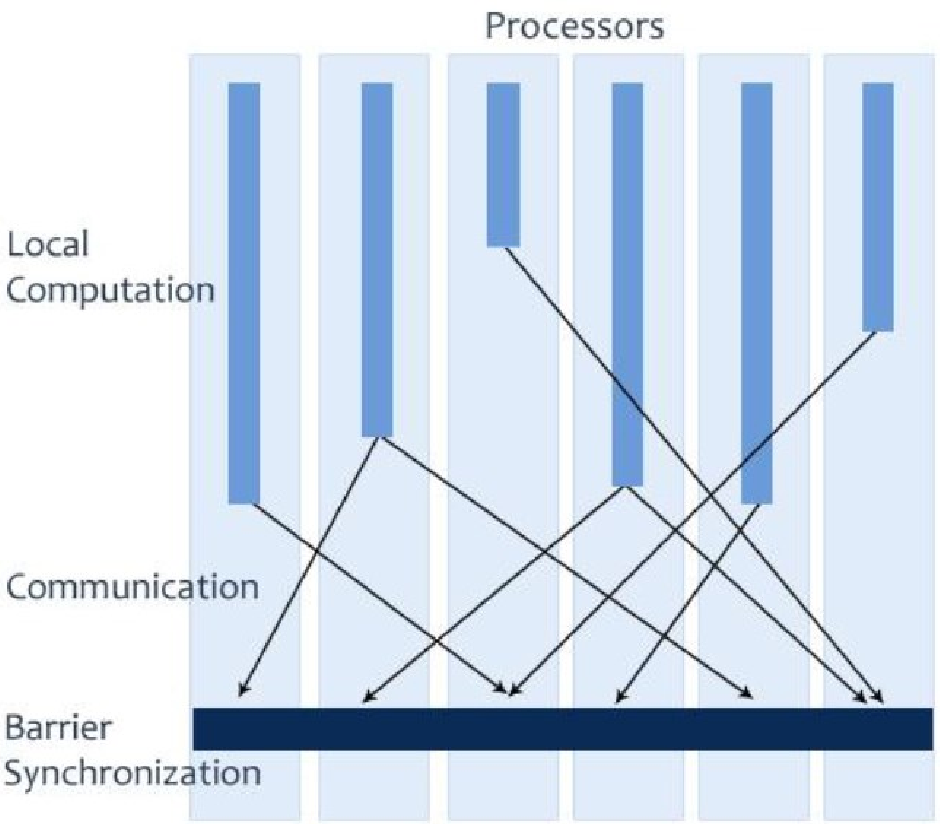
Reduce: add AND move the entire graph around



...

Bulk Synchronous Parallel (BSP) graph processing

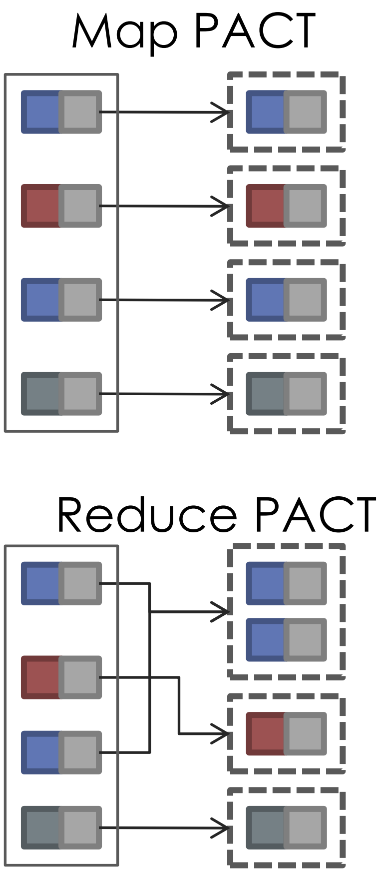
- Leslie Valiant's idea from 80's
- Google Pregel (Proprietary)
- Several open source clones
  - Giraph, …
- Dato.com's GraphLab
  - More than just BSP



- Note BSP is just a Map, followed by a Join
  - Why don't we just implement a nice Join
  - TU Berlin idea, implemented in Apache Flink

Parallelization Contract, BSP and the Join operation

- Map PACT (PArallelization ContracT)
  - Every record forms its own group
  - Process all groups independent parallel
- Reduce PACT
  - One attribute is key
  - Records with same key form a group



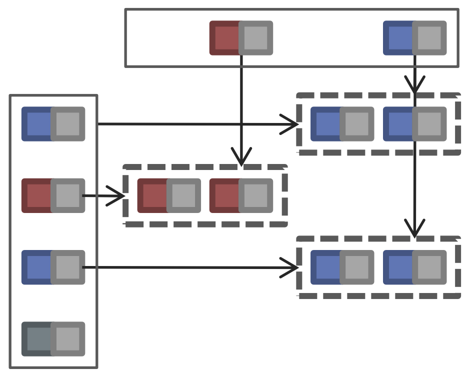
Parallelization Contract, BSP and the Join operation

<u>Join PACT</u>

Two inputs

Records with same key form a group

(equi-join)



<u>BSP</u>

Two inputs:

nodes and edges

key is node ID

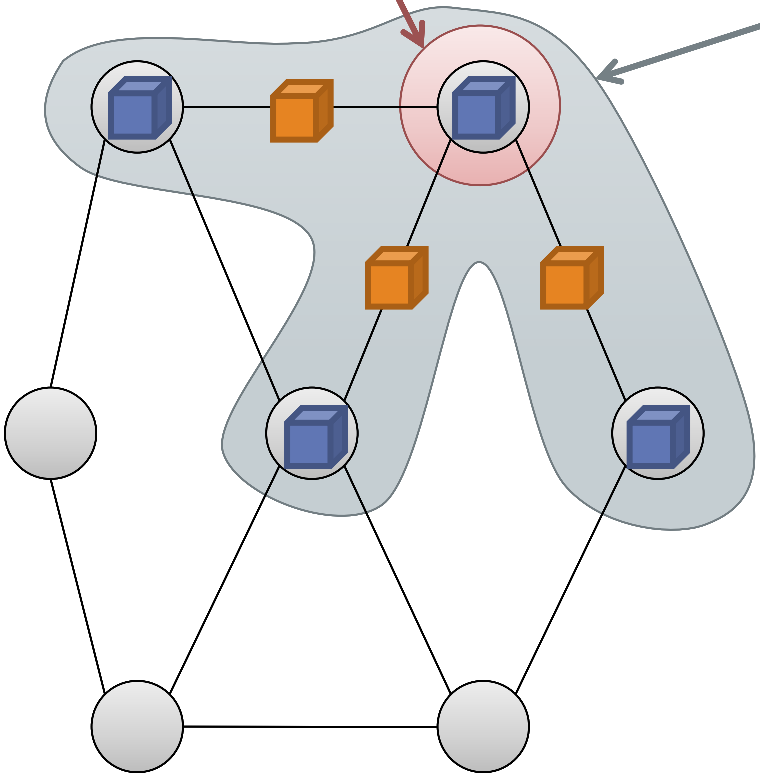
Collect all neighbors of a node

The Apache Flink system

- Several PACTs implemented
- Execution is optimized (think of versions of join) as in a database management system
- Capable of using not only disk for data passing but also memory, network by the decision of the optimizer
- Capable of native efficient iteration

The Dato.com GraphLab system

An **update function** is a user defined program which when applied to a **vertex** transforms the data in the **scope** of the vertex



Dynamic computation

PageRank in GraphLab

$$R[i] = \alpha + (1-\alpha) \sum_{(j,i) \in E} \frac{1}{L[j]} R[j]$$

```
GraphLab_pagerank(scope) {
    sum = 0
    forall ( nbr in scope.in_neighbors() )
        sum = sum + neighbor.value() / nbr.num_out_edges()

    old_rank = scope.vertex_data()
    scope.center_value() = ALPHA + (1-ALPHA) * sum

    double residual = abs(scope.center_value() – old_rank)
    if (residual > EPSILON)
        reschedule_out_neighbors()
}
```

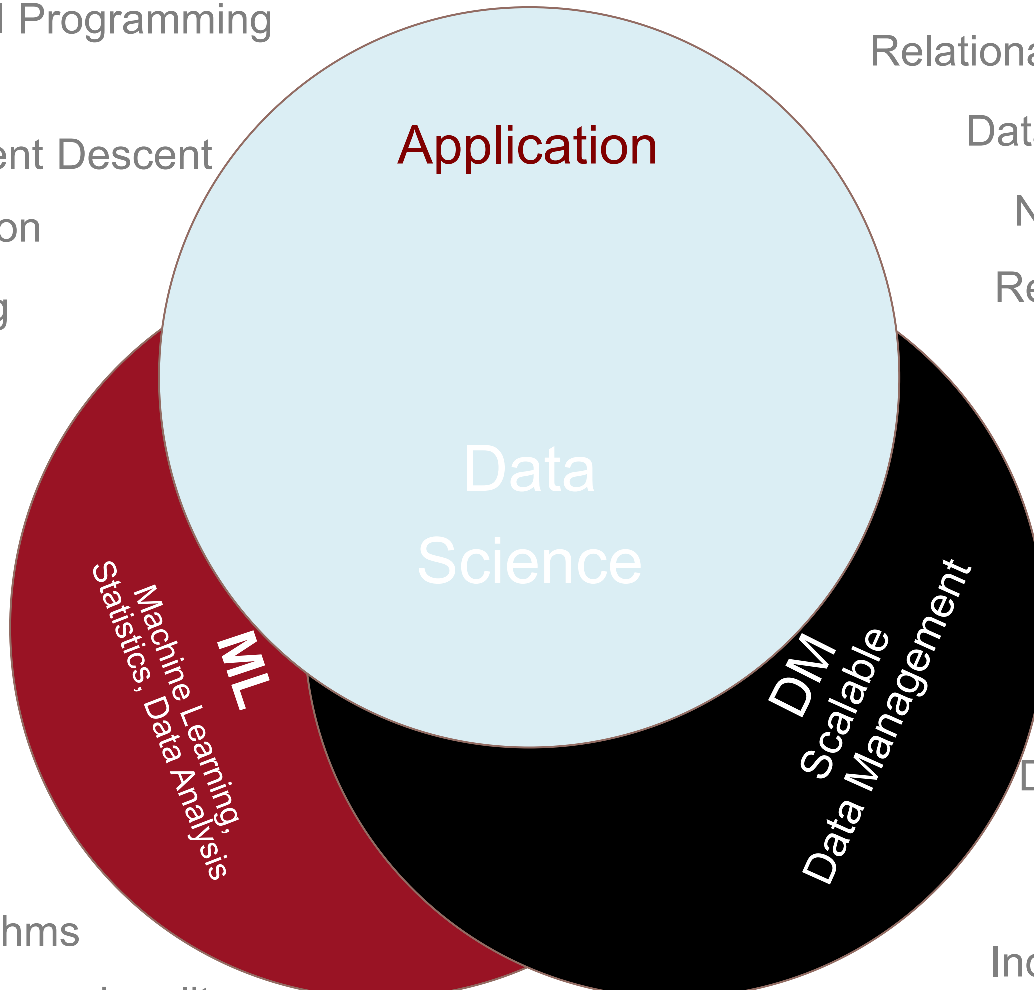
What I'd like to present next time we meet

Flink unified batch and streaming

Data Scientist magic triangle

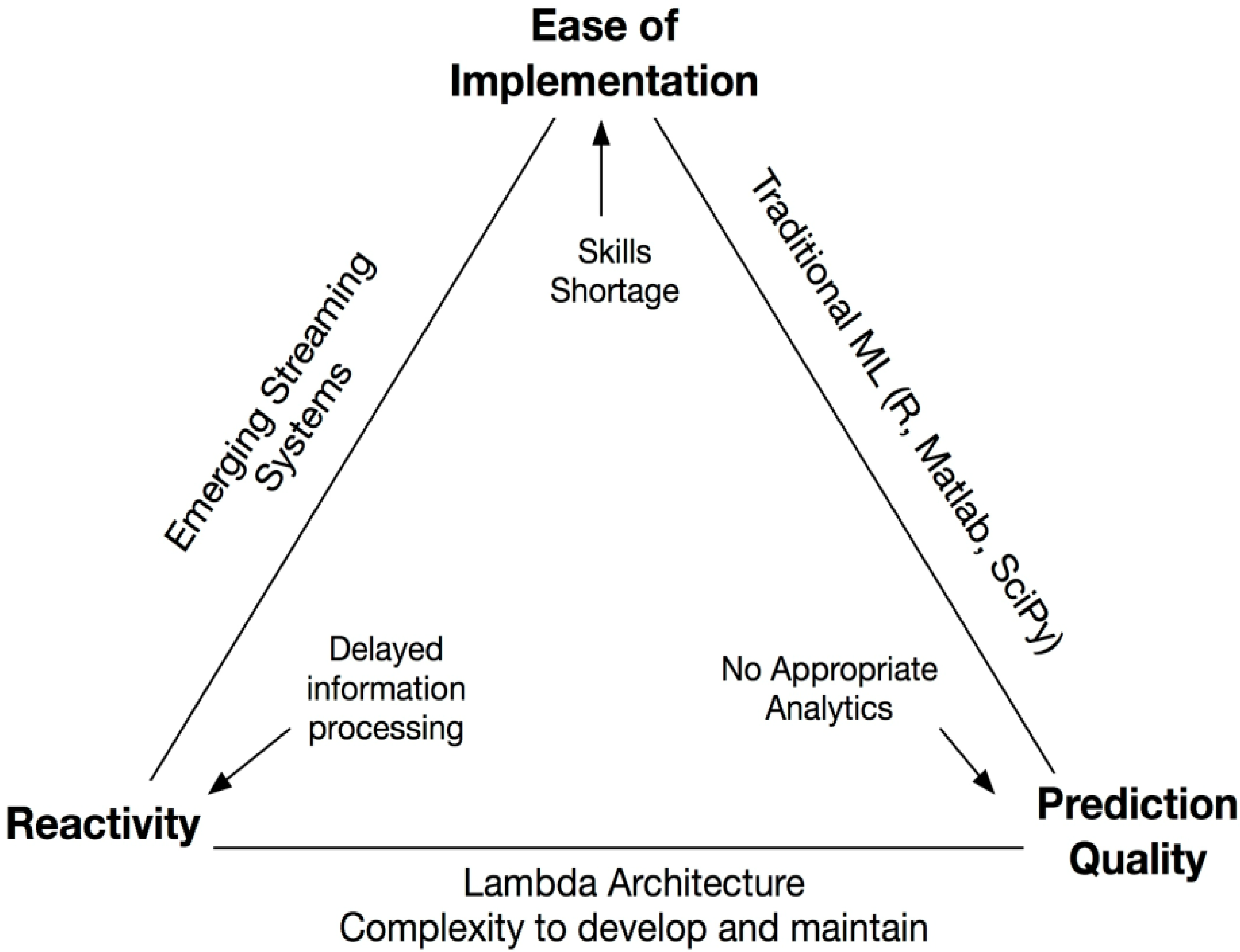
Domain Expertise (e.g., Industry 4.0, Medicine, Physics, Engineering, Energy, Logistics)

Mathematical Programming
Linear Algebra
Stochastic Gradient Descent
Error Estimation
Active Sampling
Regression
Monte Carlo
Statistics
Sketches
Hashing
Convergence
Decoupling
Iterative Algorithms
Curse of Dimensionality



Relational Algebra / SQL
Data Warehouse/OLAP
NF^2 /XQuery
Resource Management
Hardware Adaptation
Fault Tolerance
Memory Management
Parallelization
Scalability
Memory Hierarchy
Data Analysis Language
Compiler
Query Optimization
Indexing
Data Flow
Control Flow
Real-Time

STREAMLINE Magic Triangle

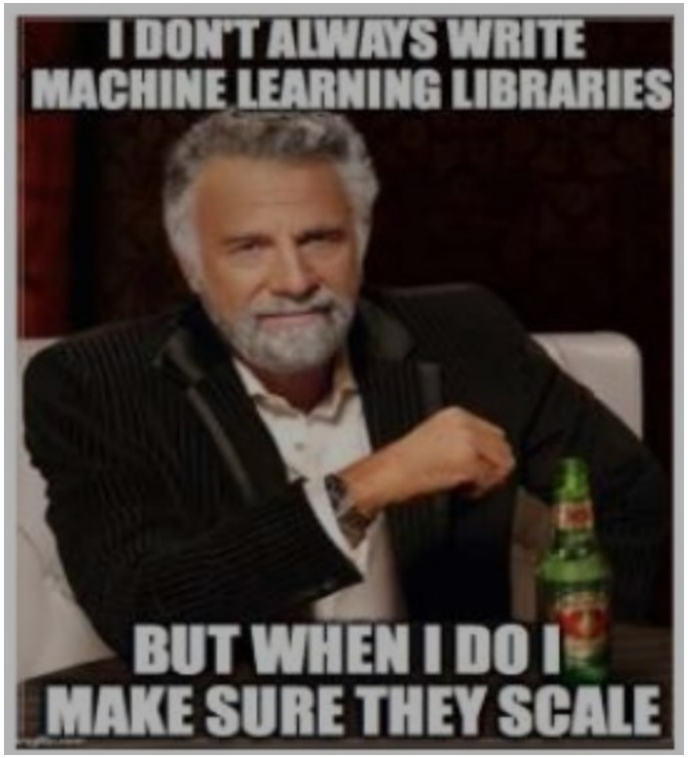


STREAMLINE Magic Triangle

Challenge	Present Status	Goal	Action	Leader
Delayed information processing	No up-to-date timely predictions	Reactivity	Same unified system for data at rest and data in motion	TU B / DFKI
Actionable intelligence: Lack of appropriate analytics	Poor or non-timely prediction results in user churn, business losses	Prediction quality	Library for batch and stream combined machine learning	SZTAKI (Andras)
Skills shortage: Human latency	Multiple expertise needed for data scientists, expensive to operate	Ease of implementation	High level declarative language	SICS

Chuck Norris versions

Flink developers



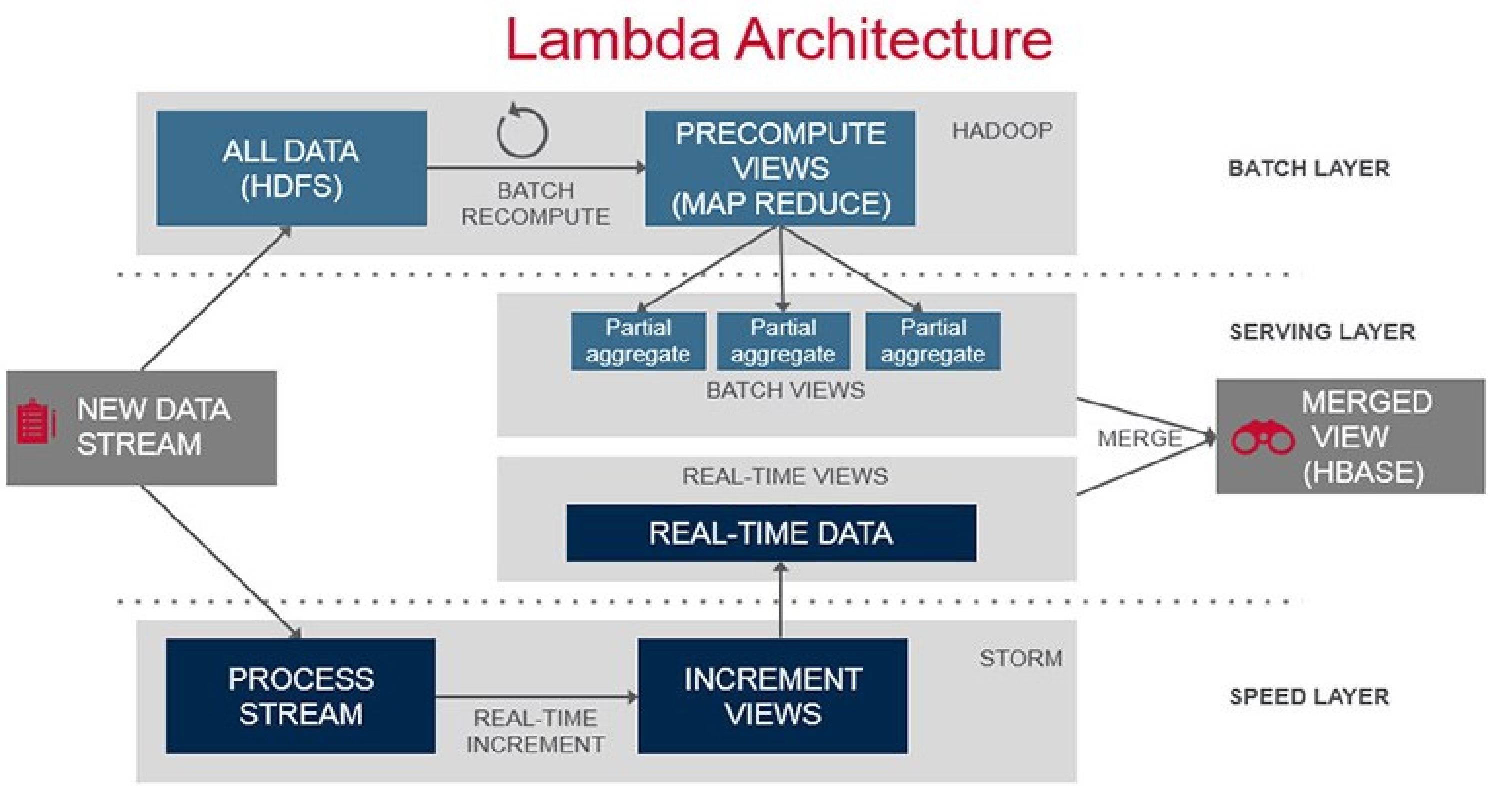
(Soon-to-be) Flink users

We don't always have to scale our machine learning tasks

But when we do, we don't sacrifice accuracy

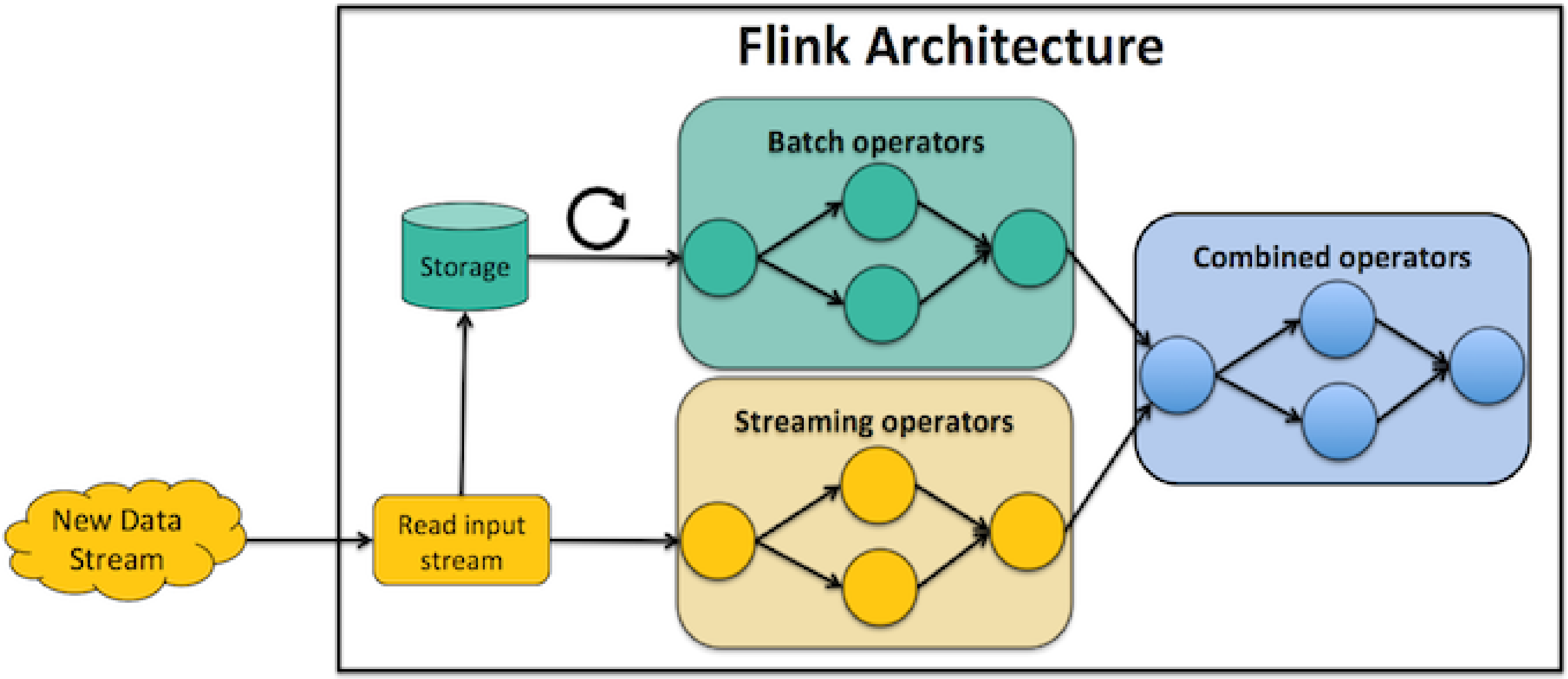
The Lambda Architecture

- Usual solution: two different systems
- Adds complexity to the architecture
- Many question the need for the batch component

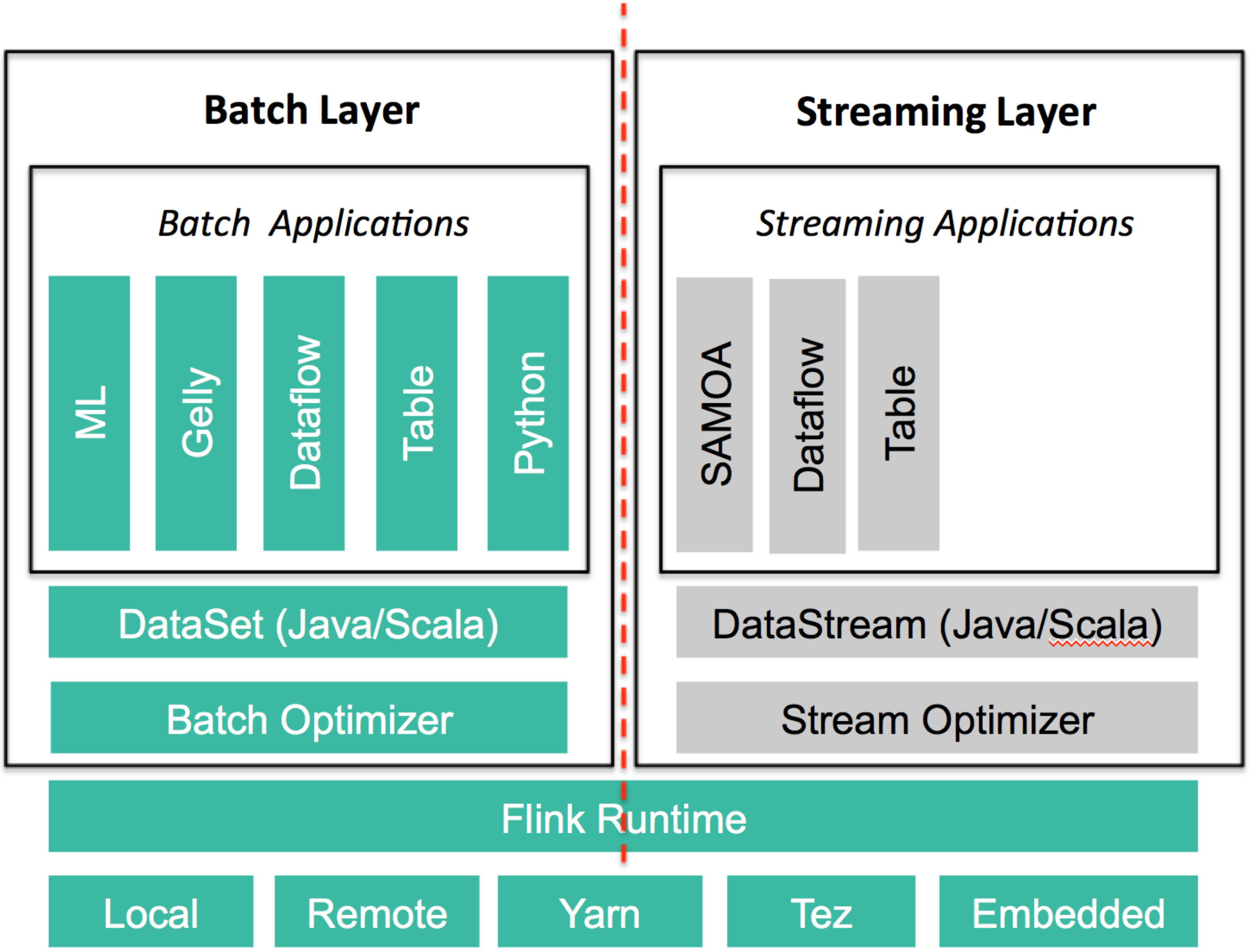


https://www.mapr.com/sites/default/files/otherpageimages/lambda-architecture-2-800.jpg

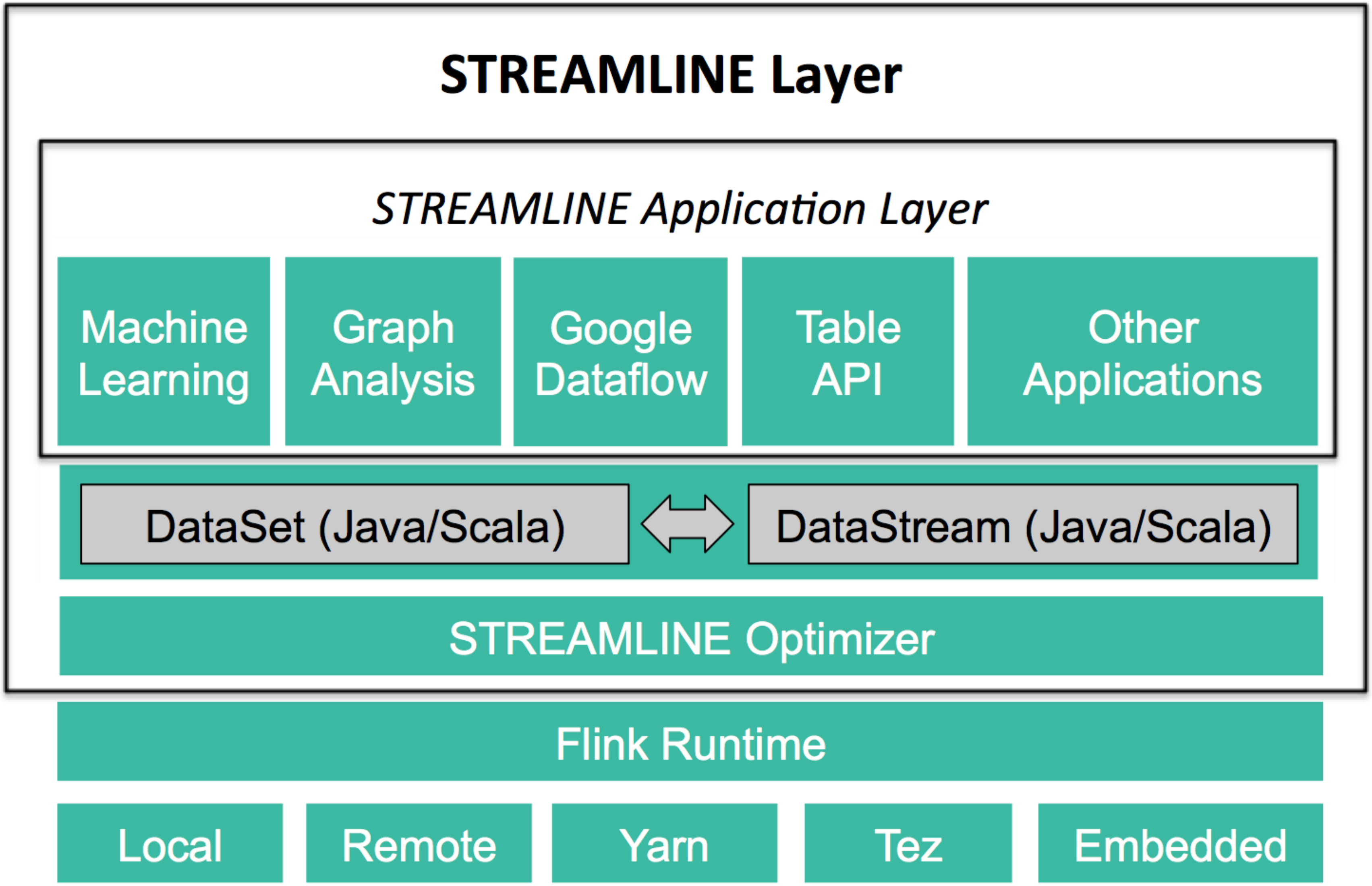
Beyond the Lambda Architecture



Current Flink architecture



STREAMLINE architecture



Conclusions

- Hadoop is a widely used open source Java MapReduce implementation
- Needs installation, some ugly boilerplate + object serialization
- Graph algorithms can be implemented by iterated joins
- Inefficient in that all graph data needs to written to disk and moved around in iterations (workarounds exist ...)

- New architecture for unified batch + stream needed
  - Apache Flink has the potential
- New machine learning is needed
  - Turning research codes to open source software will start soon

References

A very good textbook covering many areas of my presentation. Look at the online second edition at http://www.mmds.org/

- Rajaraman, Anand, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

PageRank

- Brin, Sergey, and Lawrence Page. "Reprint of: The anatomy of a large-scale hypertextual web search engine." *Computer networks* 56.18 (2012): 3825-3833.
- Fogaras, Dániel, and Balázs Rácz. "Towards scaling fully personalized pagerank." *Algorithms and Models for the Web-Graph*. Springer Berlin Heidelberg, 2004. 105-117.

Web Spam

- Castillo, Carlos, and Brian D. Davison. "Adversarial web search." *Foundations and trends in Information Retrieval* 4.5 (2011): 377-486.
- Erdélyi, M., Benczúr, A. A., Daróczy, B., Garzó, A., Kiss, T., & Siklósi, D. (2014). The classification power of web features. *Internet Mathematics*, *10*(3-4), 421-457.

Learning to Rank

- LTR survey

Web crawlert

- Lee, Leonard, Wang, Loguinov. IRLBot: Scaling to 6 Billion Pages and Beyond. WWW 2008.
- Boldi, P., Marino, A., Santini, M., & Vigna, S. (2014, April). Bubing: Massive crawling for the masses. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion* (pp. 227-228). International World Wide Web Conferences Steering Committee.

MapReduce

- MapReduce: simplified data processing on large clusters. J Dean, S Ghemawat - Communications of the ACM, 2008 [OSDI 2004]

Apache Flink

- Alexandrov, A., Bergmann, R., Ewen, S., Freytag, J. C., Hueske, F., Heise, A., ... & Warneke, D. (2014). The Stratosphere platform for big data analytics. *The VLDB Journal—The International Journal on Very Large Data Bases*, *23*(6), 939-964.