SearchLand:
search quality for beginners

Valeria de Paiva
Santa Clara University
Nov 2010

Check http://www.parc.com/event/934/adventures-in-searchland.html

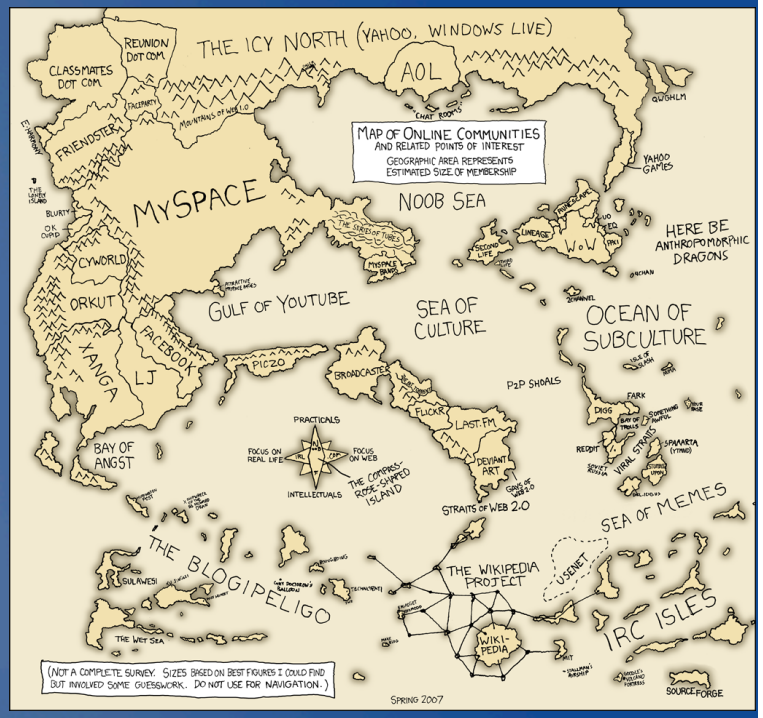

cuil
Search
Search 127 billion pages
About Cuil | Preferences | Add Cuil to Firefox
Privacy Policy | © 2010 Cuil, Inc.

Outline

- SearchLand
- Search engine basics
- Measuring search quality
- Conclusions...
- and Opportunities

SearchLand?

"Up until now most search engine development has gone on at companies with little publication of technical details. This causes search engine technology to remain largely a black art and to be advertising oriented.'' Brin and Page, "The anatomy of a search engine", 1998



Disclaimer: This talk presents the guesswork of the author.
It does not reflect the views of my previous employers or practices at work.

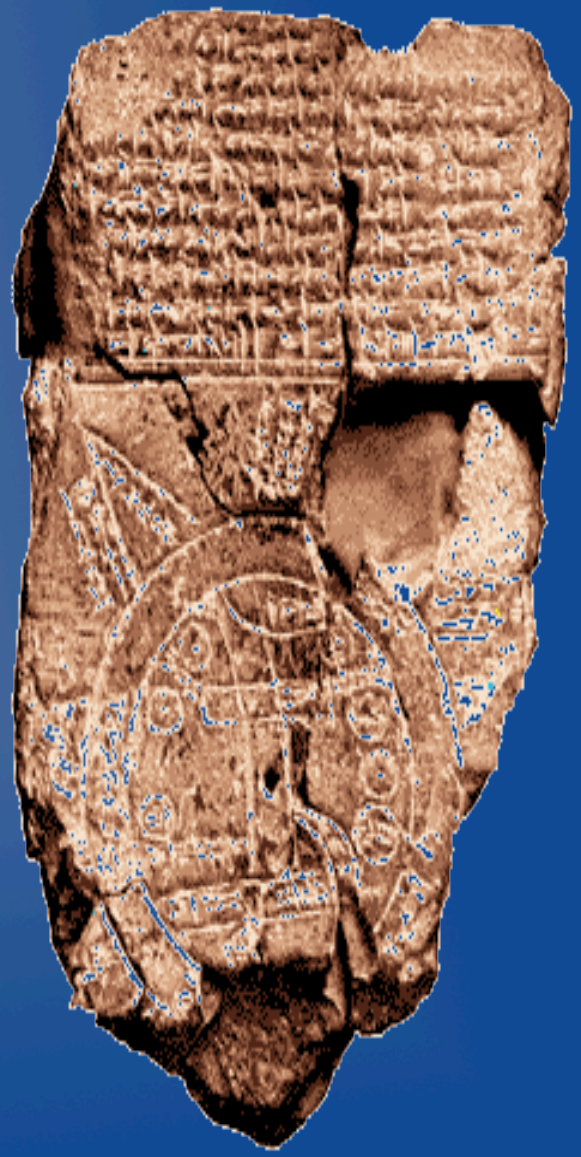
Thanks kxdc2007!

SearchLand

- Twelve years later the complaint remains…
- Gap between research and practice widened
- Measuring SE quality is `adversarial computing'
- Many dimensions of quality: pictures, snippets, categories, suggestions, timeliness, speed, etc…

SearchLand: Draft Map

Based on slides for Croft, Metzler and Strohman's "Search Engines: Information Retrieval in Practice", 2009 the tutorial **`Web Search Engine Metrics'** by Dasdan, Tsioutsiouliklis and Velipasaoglu for WWW09/10 and Hugh Williams slides for ACM SIG on Data Mining



THANKS GUYS!!…

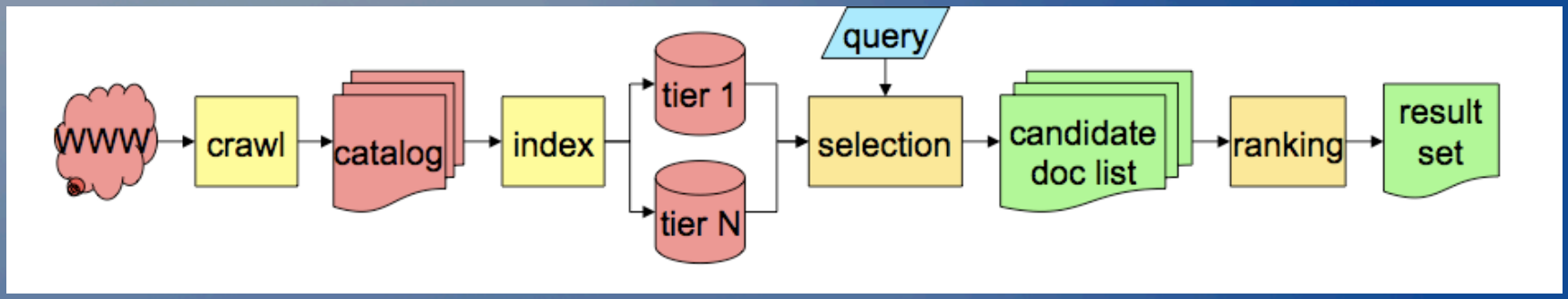
Search Engine Basics...

Web search engines don't search the web:

- They search a *copy* of the web
- They *crawl* documents from the web
- They *index* the documents, and provide a search interface based on that index
- They present short *snippets* that allow users to judge relevance
- Users click on links to visit the actual web document

Search Engines Basics

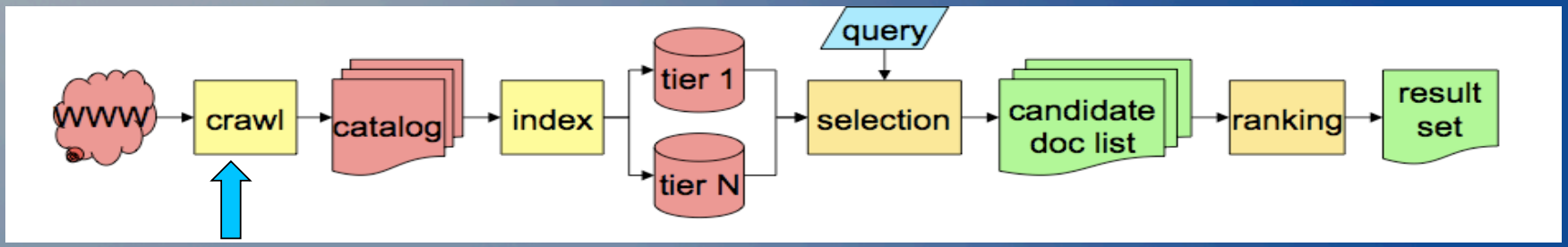
Basic architecture



From **Web Search Engine Metrics for Measuring User Satisfaction**
Tutorial at WWW conference, by Dasdan et al 2009,2010

Crawling

"you don't need a lot of thinking to do crawling; you need bandwidth"



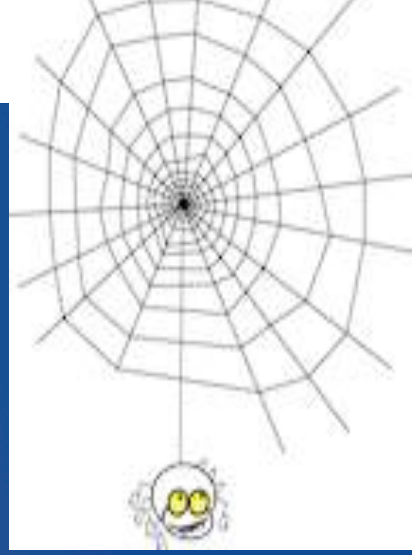
CRAWLERS
Fetch new resources from new domains or pages
Fetch new resources from existing pages
Re-fetch existing resources that have changed

Prioritization is essential:
There are far more URLs than available fetching bandwidth
For large sites, it's difficult to fetch all resources
Essential to balance re-fetch and discovery
Essential to balance new site exploration with old site exploration

Snapshot or incremental? How broad? How seeded?

Writing/running a crawler isn't straightforward….

Crawler Challenges



Crawlers shouldn't overload or overvisit sites
Must respect robots.txt exclusion standard

Many URLs exist for the same resource

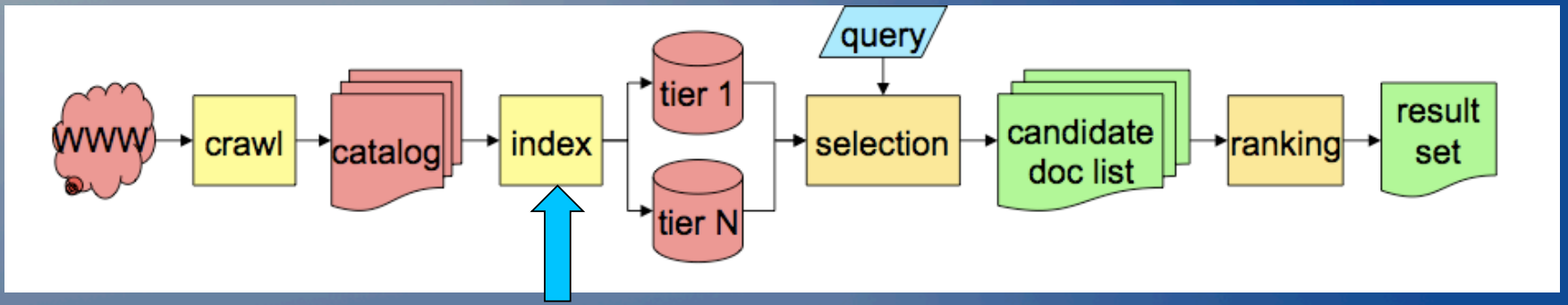
URLs redirect to other resources (often)

Dynamic pages can generate loops, unending lists, and other traps

Not Found pages often return ok HTTP codes
DNS failures
Pages can look different to end-user browsers and crawlers
Pages can require JavaScript processing
Pages can require cookies
Pages can be built in non-HTML environments

To index or not to index…

After crawling need to convert documents into index terms… to create an index.

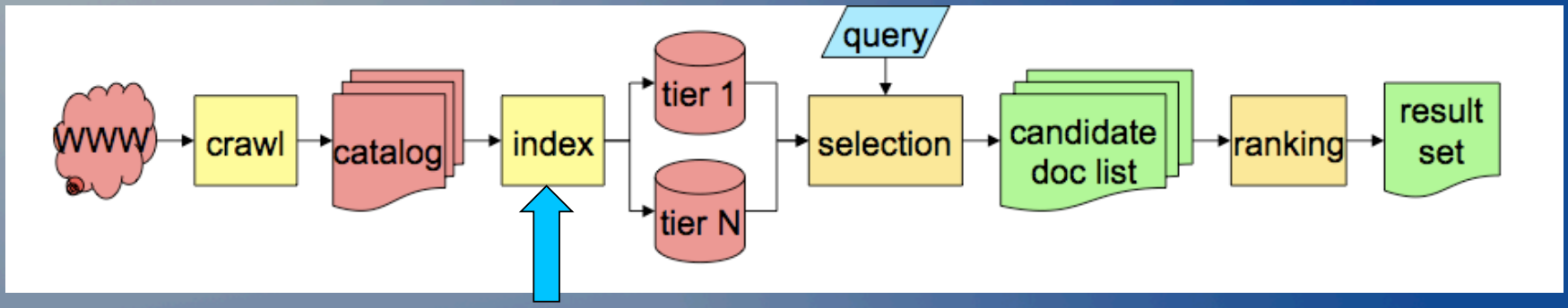


You need to decide how much of the page you index and which `features or signals' you care about.

Also which kinds of file to index? Text, sure.

Pdfs, jpegs, audio, torrents, videos, Flash???

The obvious, the ugly and the clever…



Some obvious:
hundreds of billions of web pages…
Neither practical nor desirable to index all. Must remove:
spam pages, illegal page, malware, repetitive or duplicate pages, crawler traps, pages that no longer exist, pages that have substantially changed, etc…

Most search engines index in the range of 20 to 50 billion documents (Williams)
How many pages each engine indexes, and how many pages are on the web are hard research problems…

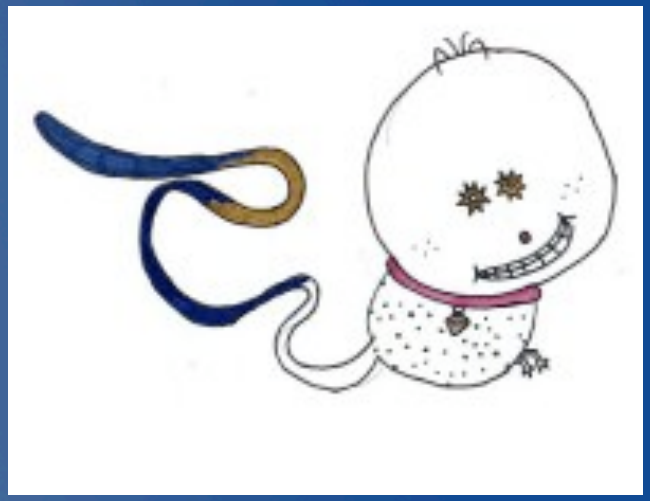
Indexing: the obvious…

which are the right pages?

pages that users want (duh…)

Pages: popular in the web link graph

- match queries
- from popular sites
- clicked on in search results
- shown by competitors
- in the language or market of the users
- distinct from other pages
- change at a moderate rate, etc…



The head is stable, The tail consists of billions of candidate pages with similar scores

Indexing: more obvious…features!

How to index pages so that we can find them?

A feature (signal) is an attribute of a document that a computer can detect, and that we think may indicate relevance.

Some features are obvious, some are secret sauce and how to use them is definitely a trade secret for each search engine.


INDEX LIBRORVM
PROHIBITORVM,
CVM REGVLIS CONFECTIS
per Patres a Tridentina Synodo delectos,
auctoritate Sanctiſs. D.N. Pij IIII,
Pont. Max. comprobatus.
VENETIIS, M. D. LXIIII.

Features: the obvious…

Term matching

The system should prefer documents that contain the query terms.

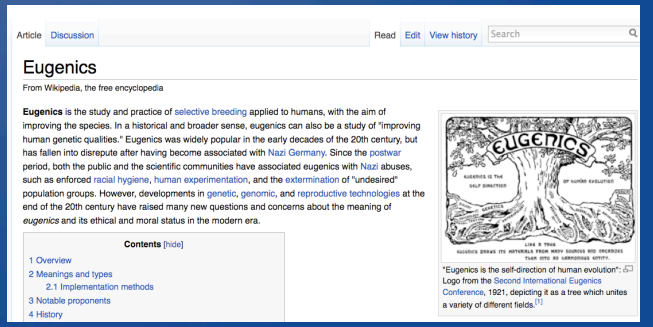
Term frequency

The system should prefer documents that contain the query terms many times.

Inverse document frequency

Rare words are more important than frequent words

TD/IDF Karen Sparck-Jones



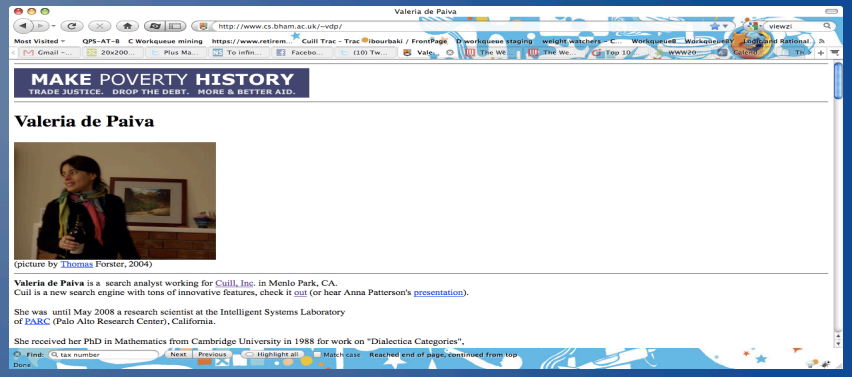
Indexing: some ugly…

Proximity

Words that are close together in the query should be close together in relevant documents.

Term Location

Prefer documents that contain query words in the title or headings.

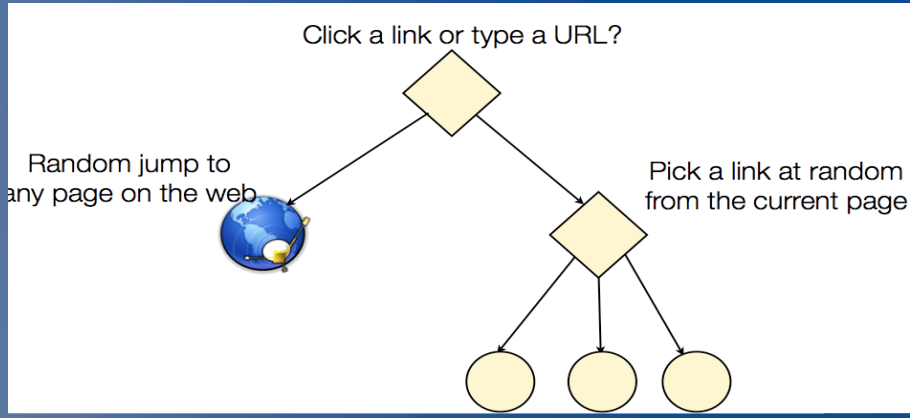


Prefer documents that contain query words in the URL.

www.**whitehouse**.gov

www.linkedin.com/in/valeriadepaiva

Indexing: some cleverness?



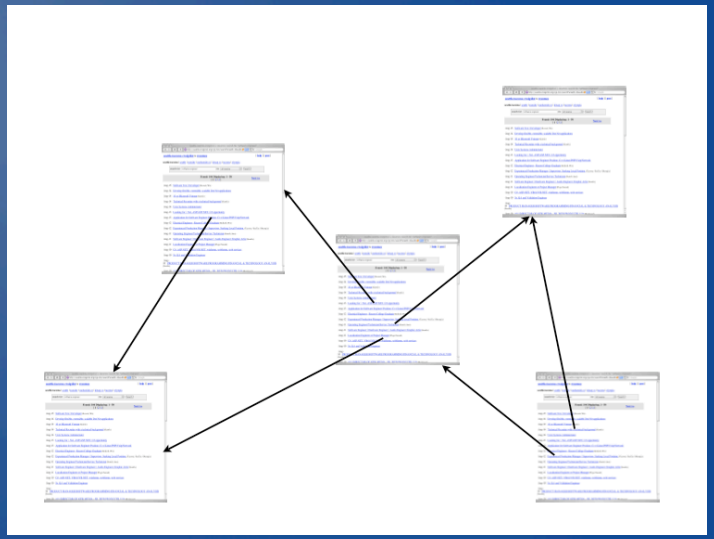
Prefer documents that are authoritative and popular.

HOW?

PageRank

in 3 easy bullets:

1. The **random surfer** is a hypothetical user that clicks on web links at random.
2. Popular pages connect…

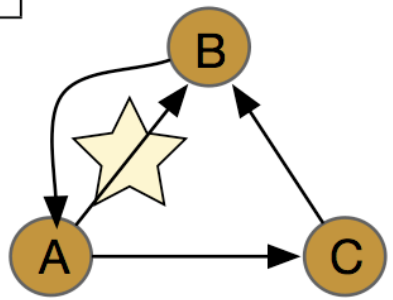


3: leverage connections?

- Think of a gigantic graph (connected pages) and its transition matrix
- Make the matrix probabilistic
- Those correspond to the random surfer choices
- Find its principal eigen-vector= pagerank

Graphs: Matrix Representation

	A	B	C
A	0	1	1
B	1	0	0
C	0	1	0

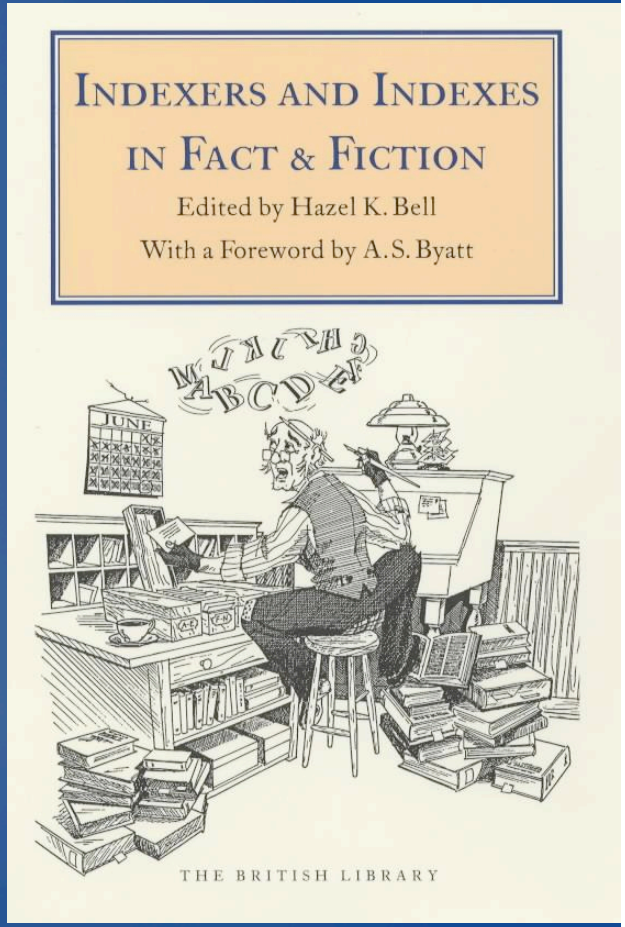


PageRank is the proportion of time that a random surfer would jump to a particular page.

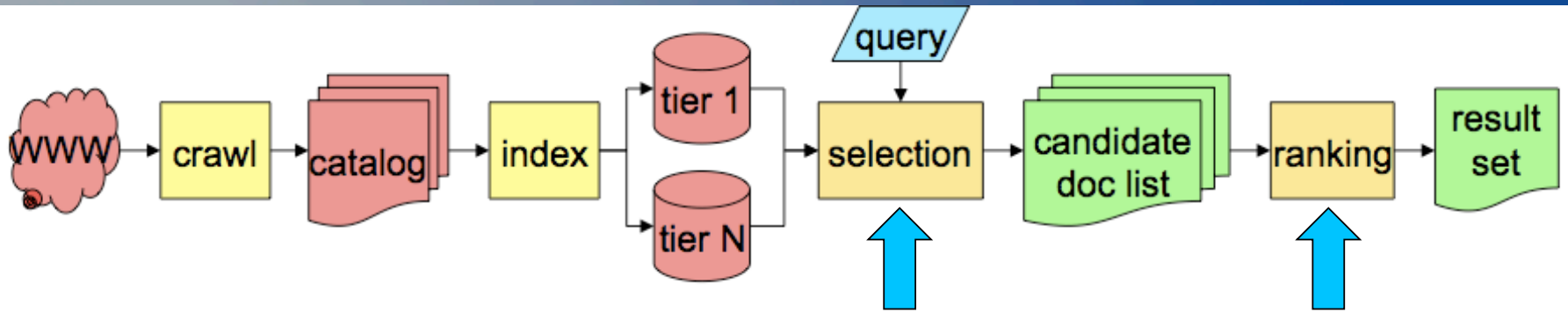
Indexing: summing up

- Store (multiple copies of?) the web in a document store
- Iterate over the document store to choose documents
- Create an index, and ship it to the index serving nodes
- Repeat…
- Sounds easy? It isn't!

Three words: scale-up, parallelism, time



Selection and Ranking



Quality of Search: what do we want to do?

Optimize for user satisfaction in each component of the pipeline

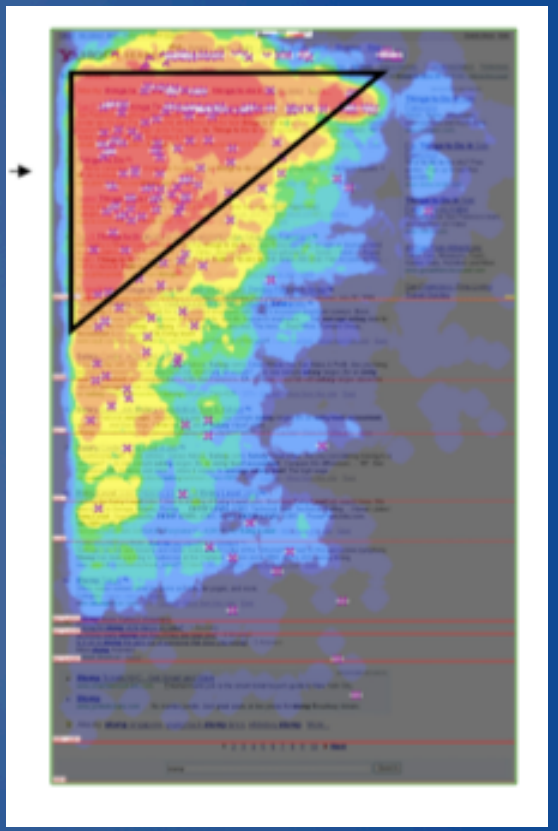
Need to ***measure*** user satisfaction

Evaluating Relevance is HARD!

- Effectiveness, efficiency and cost are related - difficult trade-off
- Two main kinds of approach:
  - IR traditional evaluations
  - click data evaluation & log analysis
- Many books on IR, many patents/trade secrets from search engines…
- Another talk

IR evaluation vs Search Engine

- TREC competitions (NIST and U.S. Department of Defense), since 1992
- Goal: provide the infrastructure necessary for large-scale evaluation of text retrieval methodologies
- several tracks, including a Web track, using ClueWeb09, one billion webpages
- TREC results are baseline, do they work for search engines?



Evaluating Relevance in IR…

if universe small you can check easily precision and recall

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})}$$

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})}$$

- **F-measure: Harmonic mean of precision and recall**
  - related to van Rijsbergen's effectiveness measure
  - reflects user's willingness to trade precision for recall controlled by a parameter selected by the system designer

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \qquad \alpha = \frac{1}{(\beta^2+1)}$$

Information Retrieval

Relevance

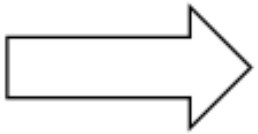
-Effective ranking

Evaluation

-Testing and measuring

Information needs

-User interaction



Search Engines

Performance

-Efficient search and indexing

Incorporating new data

-Coverage and freshness

Scalability

-Growing with data and users

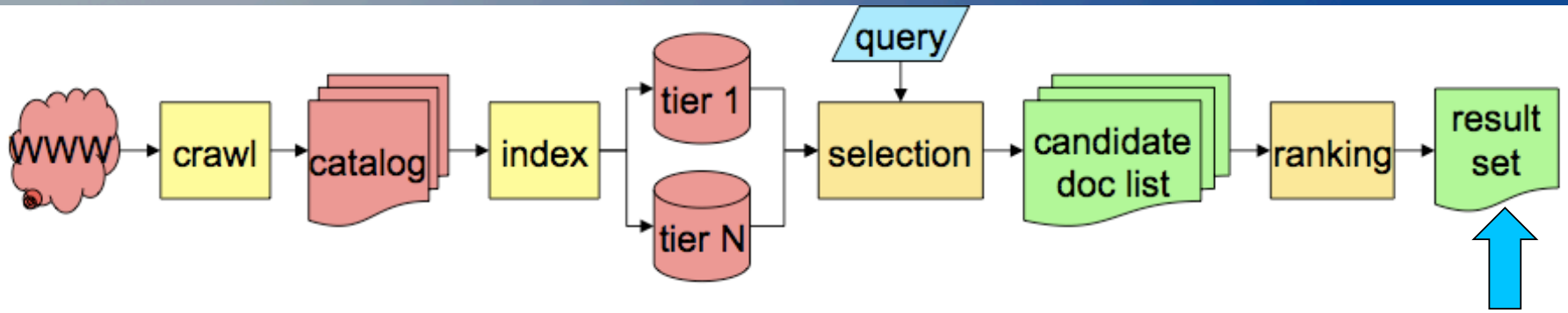
Adaptability

-Tuning for applications

Specific problems

-e.g. Spam

Evaluating the whole system…



Coverage metrics
Latency and Discovery metrics
Diversity metrics
Freshness metrics
Freshness of snippets?
Measuring change of all internet?
Presentation metrics:

suggestions, spelling corrections, snippets, tabs, categories, definitions, images, videos, timelines, maplines, streaming results, social results, …

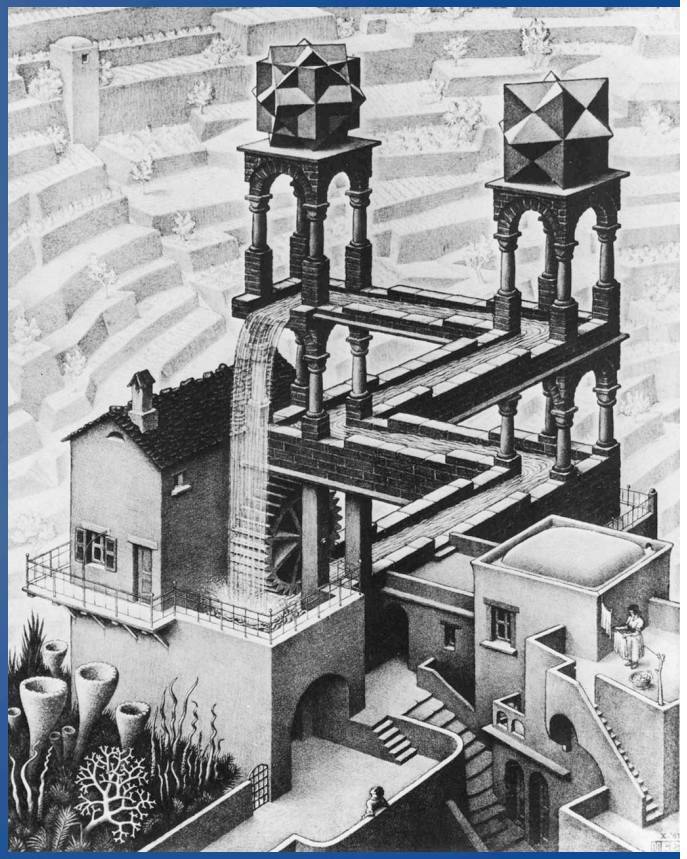
Conclusions

- Relevance is elusive, but essential.
- Improvement requires metrics and analysis, continuously
- Gone over a rough map of the issues and some proposed solutions
- Many thanks to Dasdan, Tsioutsiouliklis and Velipasaoglu, Croft, Metzler and Strohman, (especially Strohman!) and Hugh Williams

  for slides/pictures.

Coda

- There are many search engines. Their results tend to be very similar. (how similar?)
- Are we seeing everything? Reports estimate we can see only 15% of the existing web.
- Probing the web is mostly popularity based. You're likely to see what others have seen before. But your seeing increases the popularity of what you saw, thereby reducing the pool of available stuff. Vicious or virtuous circle? How to measure?



Search

Find 384,165,027 automated articles

About Cpedia | Preferences | Add Cpedia to Firefox

References

Croft, Metzler and Strohman's "Search Engines: Information Retrieval in Practice", 2009, Addison-Wesley

the tutorial `**Web Search Engine Metrics'** by Dasdan, Tsioutsiouliklis and Velipasaoglu for WWW09/10, available from http://www2010.org/www/program/tutorials/

Hugh Williams slides for ACM Data Mining May 2010

Anna Patterson:
Why Writing Your Own Search Engine Is Hard
ACM Q, 2004