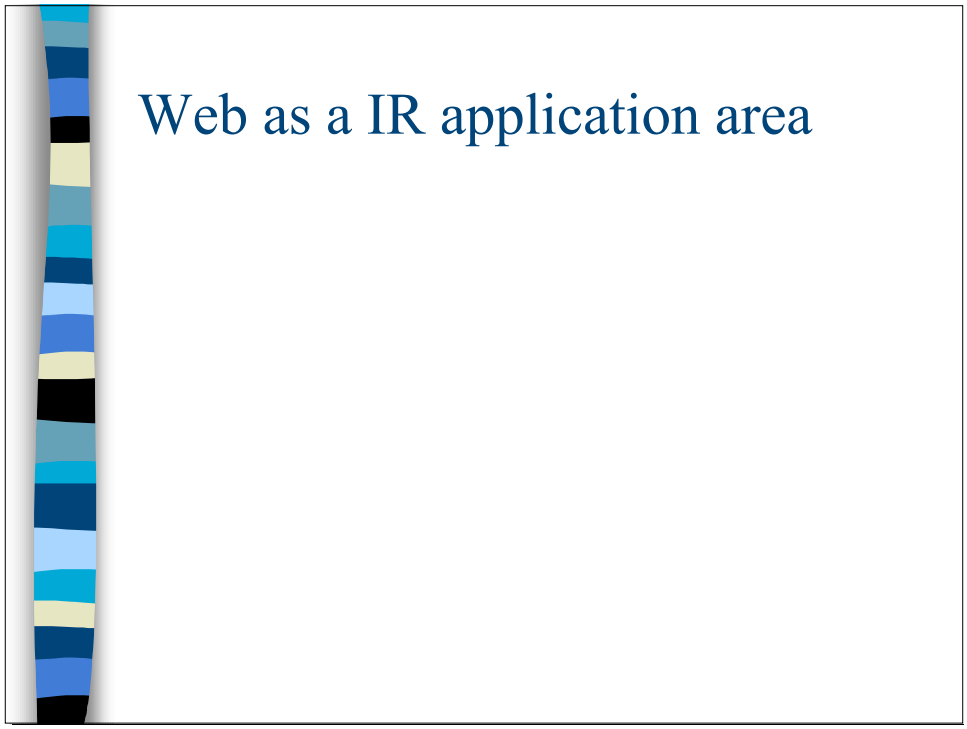
INFSCI 2140
Information Storage and Retrieval
Lecture 9: Web Information Retrieval

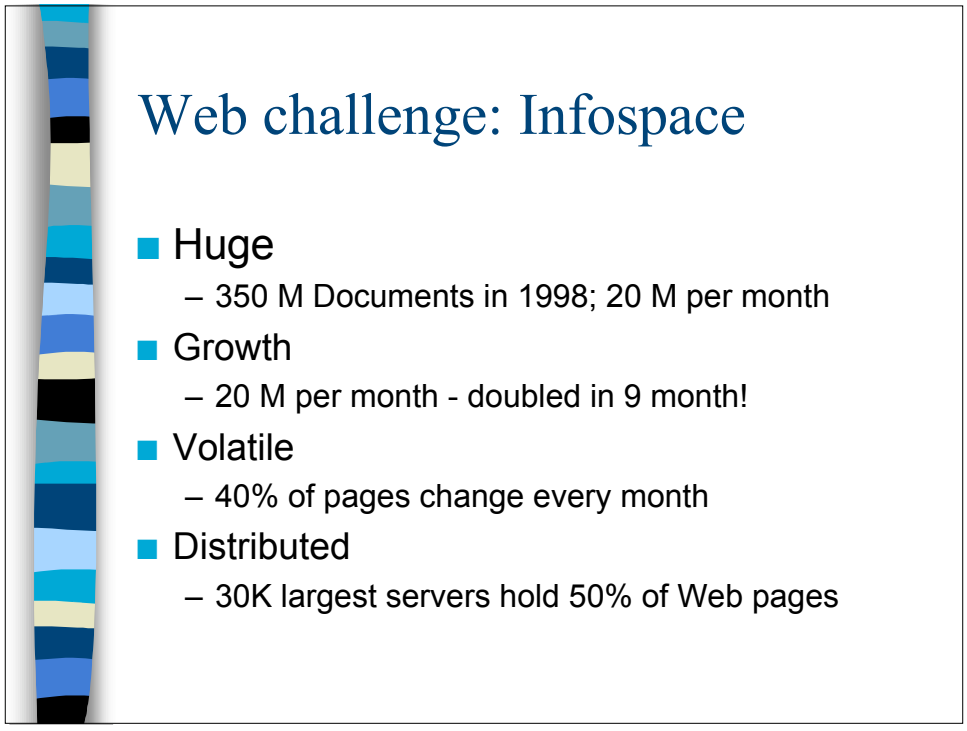
Peter Brusilovsky
http://www2.sis.pitt.edu/~peterb/2140-051/

Overview

- Characterizing the Web
  - Size
  - Topology
  - Users
- Searching the Web
- Browsing the Web
- Information services of the future



Web as a IR application area



Web challenge: Infospace

- Huge
  - 350 M Documents in 1998; 20 M per month
- Growth
  - 20 M per month - doubled in 9 month!
- Volatile
  - 40% of pages change every month
- Distributed
  - 30K largest servers hold 50% of Web pages

Web challenge: Content

- Redundant
  - 30% of Web pages are duplicates or variants
- Heterogeneous
  - Text, HTML, images, videos, music...
- Multi-language
  - 10 major languages hold >1% of the Web
- Varying quality
  - Publishing without editors
- Linked
  - An average page has 5-15 links (average is 8)

Web challenge: Users

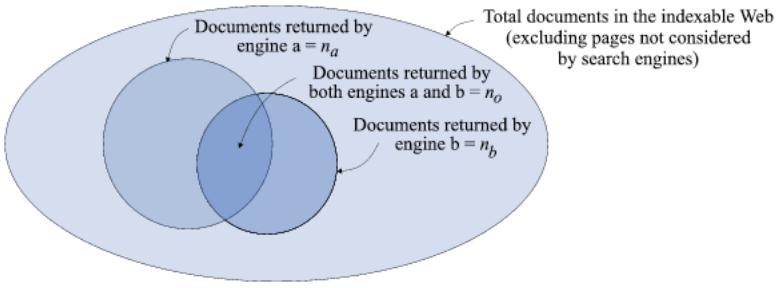
- Largest variety of users
  - Differ in needs, education, skills…
- Not skilled in query formulation
  - Average query is about 2 words, no operators
- Not patient/skilled in browsing results
  - 85% of users only look at the first screen returned by search engine
  - 78% of users use only one query
- But there are incredible power users

How large is the Web?

- Steve Lawrence and Lee Giles, NECI
- “Searching the World Wide Web", Science, 280 (April, 1998)
- “Accessibility of information on the Web", Nature, 400 (July, 1999)
- Multi-search engine technology

How to measure the Web?

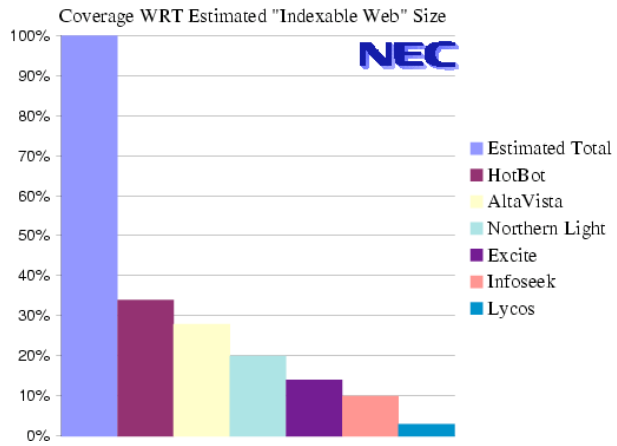
- Sampling and checking technology
  - Sample: pick a large subset of pages
  - Check whether each is indexed by an engine



Web Size: Dec 1997 Snapshot

- Estimated size of the indexable Web (IW) is 320 million pages
- Search engine coverage varies by an order of magnitude
- Any major engine index only a fraction of IW
- Combining the results of multiple engines can increase coverage

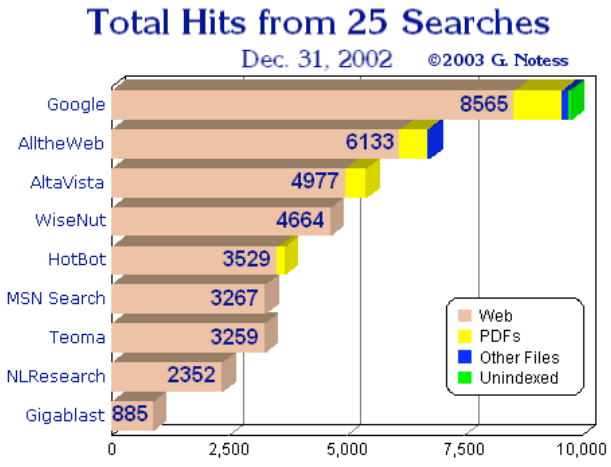
Web Size and SE coverage: 1997



Web Size: 1999 Snapshot

- The publicly indexable WWW contained about 800 million pages - 15TB of info.
- The search engine with the largest index, Northern Light, indexed roughly 16% of the publicly indexable WWW (coverage decreased!)
- The combined index of 11 large search engines covered (very) roughly 42% of the publicly indexable World Wide Web.

Search Engine Coverage 2002

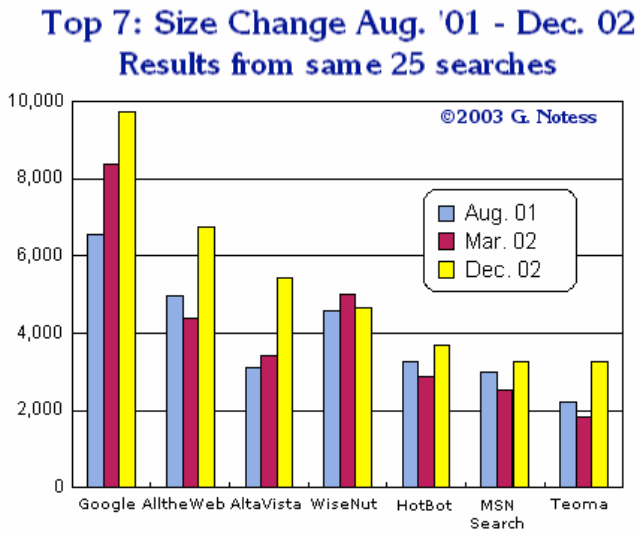


How to measure?

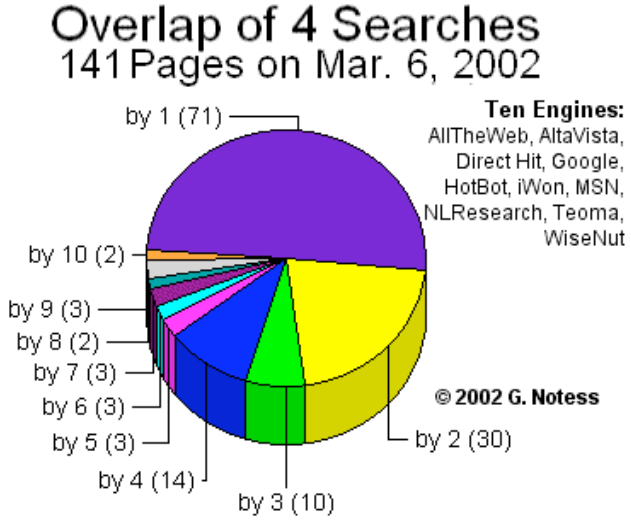
Analysis with 25 small single word queries

By http://www.searchengineshowdown.com/

Dynamics of the Coverage



Search Engines Overlap



Evaluation of Web Search

- How we can use traditional evaluation metrics?
- How we can measure precision if the user does not scroll?
- How we can measure recall if the coverage in inherently incomplete?
- "Web Directories" method (using category names/content of Web drectories)

Web Growth (OCLC Data)

- Number of Web Sites
  - A Web site is defined as a distinct location on the Internet, identified by an IP address, that returns a response code of 200 and a Web page in response to an HTTP request for the root page. The Web site consists of all interlinked Web pages residing at the IP address.
- **1998:** 2,851,000
- **1999:** 4,882,000
- **2000:** 7,399,000
- **2001:** 8,745,000
- **2002:** 9,040,000

Web Content: Language

1999		2002	
Language	**% Public Sites**	**Language**	**% Public Sites**
English	72%	**English**	72%
German	7%	**German**	7%
French	3%	**Japanese**	6%
Japanese	3%	**Spanish**	3%
Spanish	3%	**French**	3%
Chinese	2%	**Italian**	2%
Italian	2%	**Dutch**	2%
Portuguese	2%	**Chinese**	2%
Dutch	1%	**Korean**	1%
Finnish	1%	**Portuguese**	1%
Russian	1%	**Russian**	1%
Swedish	1%	**Polish**	1%

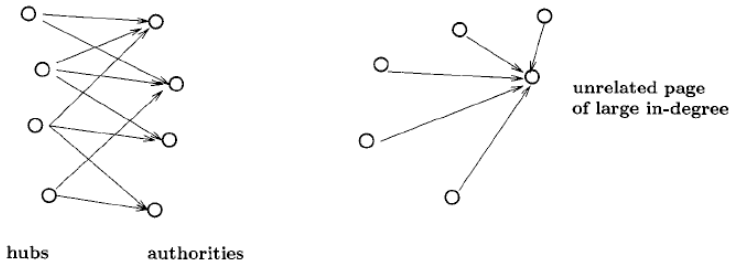
Web Content

- Type (1999)
  - 83% commercial
  - 6% scientific and education
  - 1.5% adult
- The growth of Adult Web Sites
  - Public Web sites whose primary content is sexually explicit images or text.

Year	**N**	**% Public Sites**
2000:	68,000	2.3%
2001:	74,000	2.4%
2002:	102,000	3.3%

Web topology

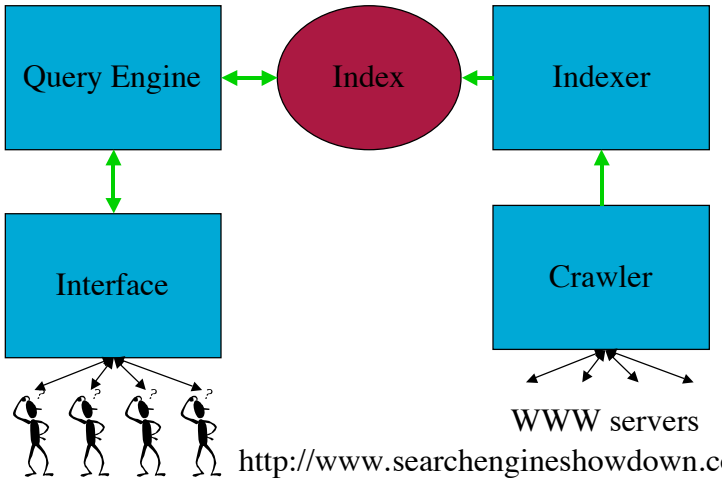
- Domain suffixes and names
- Hierarchical structure
- Hubs and Authorities (J. Kleinberg)



Search tools

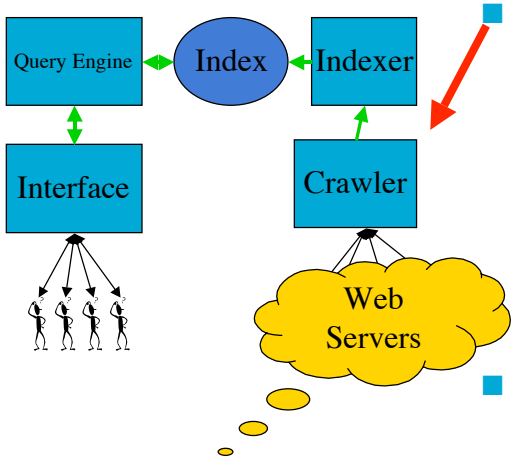
- Search Engines
  - Good coverage, low quality
- Directories
  - Good quality, low coverage
  - Automatic classification?
- Meta-search engines
- Dynamic search

Search Engines: Architecture



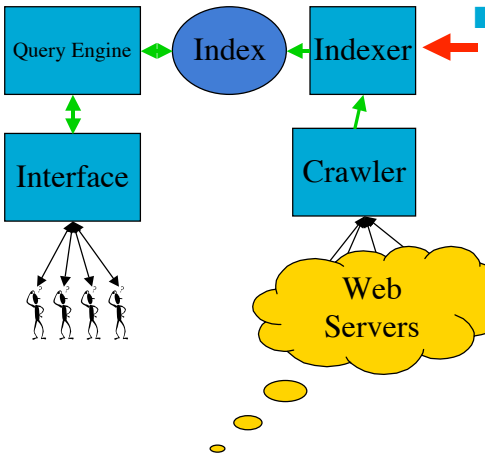
http://www.searchengineshowdown.com/

Search Engines: Architecture



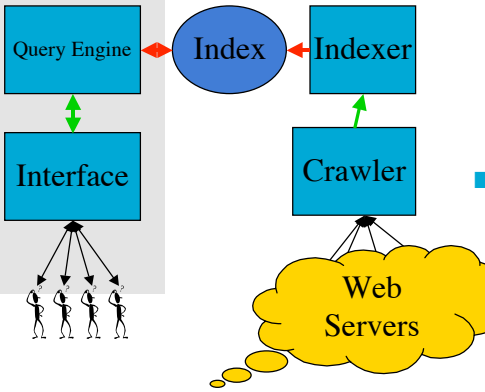
- Crawlers (robots, spiders, walkers) run on the local server(they do not move on the Web) and send requests to remote Web servers
- They gather Web pages

Search Engines: Architecture



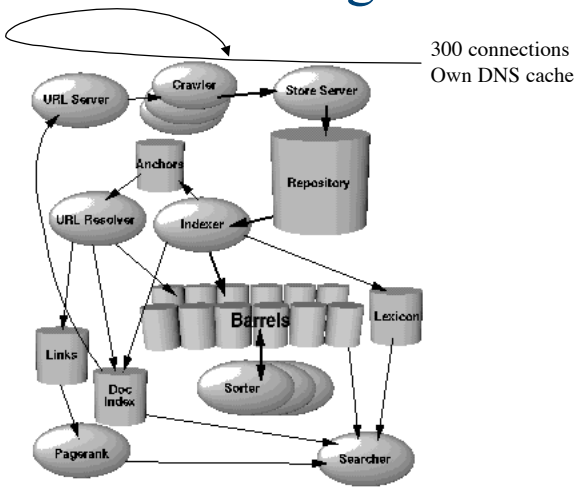
- Indexers creates the surrogate or document representation that is memorized in the index

Search Engines: Architecture



- On the other side we have the part that is related to the user
- Some search engines cache the most common queries and the corresponding results

Architecture of Google

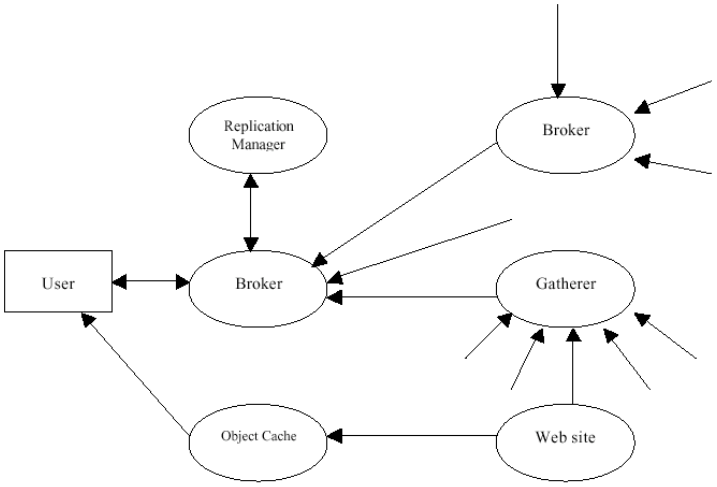


Source: Sergey Brin and Lawrence Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine
http://www7.scu.edu.au/programme/fullpapers/1921/com1921.htm

Crawling and crawling problems

- What is a crawler?
- Problems of crawling
  - Lots of work once and again
  - Where to go?
  - Be fair to Web sites
- Solutions:
  - Distributed crawling (Google vs. Harvest)
  - Ranking for crawling (Winograd)
  - Techniques to deal with hitting Web sites
  - Re-visiting techniques

Harvest Distributed Crawling



Harvest Distributed Crawling

- A gather collects the information from the Web and extracts indexing information from the material
- A broker provide the indexing mechanism and the query interface to the user
  - A broker can run on a Web server generating no extra traffic for that server
  - One of the goal of the project is to build a topic-specific broker, focusing the index content and avoiding many of the vocabulary problems
- A replicator is used to replicate brokers
- An object cache is used to reduce network and servers load

Indexer and indexing problems

- Size! What an indexer can store?
  - Terms (and positions of terms in a document)
  - Date of visiting
  - Start of a page
  - The whole page (cached)
- The example of Google
- The problem of changing Web
  - Internet Archive: Wayback Machine: http://web.archive.org/

Search problems

- How to find results fast?
  - Smaller indexes
    - stop words, stemming, one-case
  - Distributed architecture
- How to rank results
  - Users have no patience
  - High-quality first
  - Fight spam

Ranking search results

- Classic Vector Model
  - TF*IDF and relative term frequencies
- Spread Activation
  - Takes links into account
  - Boolean and vector spread activation
- Google's PageRank Algorithm
- Kleinberg's HITS Algorithm

Meta-search Engines

- Why should you develop your own search engine? There are so many...
- Discovery from Web measurement
  - Intersection between search engines is small
- The problem of ranking
- Adaptive Meta-Search

Dynamic Search

- The idea is not to search the information stored in a search engine, but actually *the Web itself*
  - The search is slower (agent metaphor)
  - It might be used in small and dynamic subsets of the web
- Also known as focused crawling
- Same considerations as in building crawlers

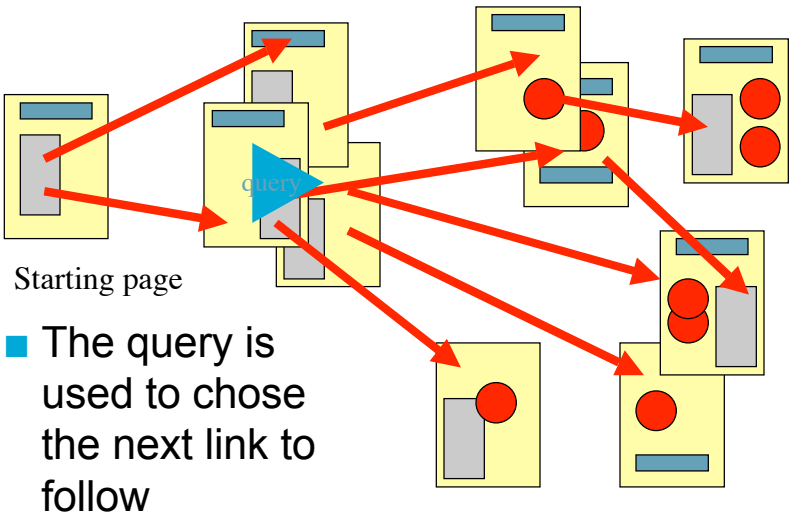
Dynamic Search

- One of the proposed algorithm is *fish search* which exploits the intuition that relevant documents often have neighbors that are relevant
- The main idea of these algorithms is to follow links in some priority given:
  - a starting page
  - a query that defines the kind of page we are looking for

Adaptive Focused Crawling

- An ability to launch an agent that search the Web for you taking into account your query and your profile and discover useful resources
- More close to Information Filtering
- A number of research projects
  - Bazaar, Arachnid
  - MySpiders: http://myspiders.informatics.indiana.edu/

Dynamic Search



- The query is used to chose the next link to follow

Browsing + Crawling

- The idea
  - While you are browsing the Web, your "agent" runs ahead of you checking pages one-two-three steps in front of you
  - Knowing your interests (and whatever else your User Model stores) it can recommend best pages and best direction for browsing
- Letizia project (Henry Lieberman, MIT)

Current Challenges

- New IR models are needed in order to
  - face the constant change of the document set
  - (better) exploit HTML and link information: link position in the page, anchors, etc…
- Querying modes
  - So far we searched only for content, but we can also search for structure (of a page, or links to of from a page, we should want to search for hubs or references).

Current Challenges

- Crawling
  - more sophisticated architectures in order to cope with the grow rate of the Web.
- Ranking
  - ranking pages not only on the basis of the relevance to the query but also the "authority" of the page (better than Google)
  - ranking meta-search results

Current Challenges

- Searching "hidden Web"
  - A large part of the Web is dynamically created for the user. These pages are invisible for a search engine
- Multimedia search
  - We need a way to search images, video, audio, Flash animations, Animated GIFs
- Personalization
  - We need to provide adaptive IA

Current Challenges

- Query space does not match document space
- Main hypothesis of IR is broken!
- How to build a mapping from a query to documents?
- Rely on human relevance judgement!
  - Using Web directories
  - Mining Web anchors (Kraft, WWW 2004)
  - Mining results of successful search

Evaluation of Web Search

- How we can use traditional evaluation metrics?
- How we can measure precision if the user does not scroll?
- How we can measure recall if the coverage in inherently incomplete?
- “Web Directories” method (using category names/content of Web drectories)

Adaptive Web IA systems

- Adaptive Search Engine Filters
- Adaptive Meta-search engines
- Adaptive Focused Crawlers (agents)
- Browsing guides
  - Adaptive guidance (WebWatcher)
  - Adaptive annotation (Syskill&Webert)
- Recommenders
  - Closed Corpus (SiteIf)
  - Open Corpus (Letizia, SurfLen)
- Adaptive bookmark managers

Information services

- Integrates several functions
  - Support users in ad-hoc retrieval
  - Support SID filtering
  - Support bookmarking
  - Support users in browsing
  - Provide collaborative recommendation
  - Launch adaptive agents for collecting information
- Examples: FAB, ELFI