

707.000
Web Science and Web Technology
„Link Analysis and Search“

Markus Strohmaier

Univ. Ass. / Assistant Professor
Knowledge Management Institute
Graz University of Technology, Austria

e-mail: markus.strohmaier@tugraz.at
web: <http://www.kmi.tugraz.at/staff/markus>

Overview

Agenda

- Architecture of search on the web including an overview of
 - Crawling, indexing
 - Link analysis
 - Search Evaluation

Slides based on

- M. Lux, Information Retrieval I&II, Web-based Retrieval,
<http://www.itec.uni-klu.ac.at/~mlux/>
- C. Gütl, Information Search and Retrieval,
<http://www.iicm.tugraz.at/isr/>

Web based Retrieval: Challenges

Working with an enormous amount of data

10 billion pages a 500kB estimated in 01-2004

- 2 pages / person on the globe

20 times larger than the LoC print collection

- estimated in 2003

Furthermore there is a **Deep Web**

- 550 billion pages estimated in 2004

Web based Retrieval: Challenges

Example for the amount of web pages:

- Searching for 'Star Trek' yielded about 11 million of results on Google [Nov 2007]
- Ordinary users investigate 20-30 result list entries.

What web page is the most interesting?

How to store an index (inverted file) with this size?

Web based Retrieval: Challenges

The Web is highly dynamic

Study by Cho & Garcia-Molina (2002):

- 40% of the web pages changed their dataset within a week
- 23% of the .com pages changed on daily basis

Study by Fetterly et al. (2003):

- 35 % of the pages changed while the investigations
- Larger web pages change more often

Web based Retrieval: Challenges

The Web is self-organized

No central authority (for the WWW) or main index

Everyone can add (even edit) pages

Pages disappear on regular basis

- A US study claimed that in 2 investigated tech. journals 50% of the cited links were inaccessible after four years.

Lots of errors and falsehood, no quality control

Web based Retrieval: Challenges

The Web is hyperlinked

Based on HTML Markup tags and URIs

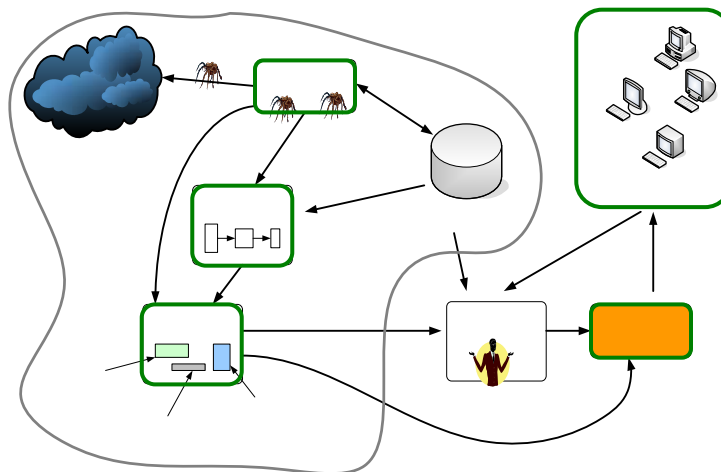
Pages are interconnected

- Unidirectional links (in-link, out-link, self-link)

Network structures emerge from the links

- Link analysis is possible

Common Architecture



History of Crawlers [Witten 2007]

- World Wide Web Wanderer (1993)
 - Purpose not to index, but to measure its growth
- WebCrawler (1994)
 - First full-text index for entire web pages
- Lycos, Infoseek, Hotbot (1996)
- AskJeeves, Northern Light (1997)
- Others: OpenText, AltaVista
- Yahoo (What's that acronym?)
 - Two Stanford PhD students

Yet Another
Hierarchical Official
Oracle“

And then came Google (1998)

- Another two Stanford PhD students (T. Winograd)
 - Who are now allowed to land their private air planes on a NASA airfield close to Mountain View
- <http://www.sfgate.com/cgi-bin/article.cgi?f=/c/a/2007/09/13/BUPRS4MHA.DTL>

Crawler

Crawlers, robots & spiders harvest sites

Starting with a **root set** of URLs

Following links, that are found on the pages

Applying **filters** to the links

- e.g. only .at domains -> Austrian web pages
- e.g. based on link title & position (focused crawling)

Crawlers: Index Update

- Which sites should be updated and when?
- A page content might have changed since last visit
 - last modified dates are eventually inaccurate
- Different strategies are possible:
 - Refresh only portions ...
 - Prefer most popular sites ...

Ethical Questions:

- How much bandwidth is used?
 - Hit counts ...
- What does that mean for the server load?
- Let loose several spiders at once
 - Decrease of crawling time
 - Increase of load

Crawling: Robots.txt

Robots.txt is option for webmasters to

- restrict crawler access
- point crawlers to interesting URLs
- identify crawlers (with hit on the robots.txt)
- see <http://www.robotstxt.org/wc/robots.html>

Example

```
User-agent: *  
Disallow: /wp-admin/  
Disallow: /netadmin/
```

Crawler: Google sitemaps

XML schema to identify interesting portions & updates
of a web page

Integration into CMS optimal

Example:

```
<url>
  <loc>http://www.semanticmetadata.net/</loc>
  <lastmod>2007-02-06T11:26:06+00:00</lastmod>
  <changefreq>daily</changefreq>
  <priority>1</priority>
</url>
```



What's a
good
crawler?

Crawler: Coverage, Freshness and Coherence [Witten 2007]

Coverage:

- The percentage of pages that a crawler indexes

Freshness:

- The reciprocal of the time that elapses between successive visits to websites

Coherence:

- The overall extent to which the index corresponds to the web itself

Indexing Module

Takes each new uncompressed page

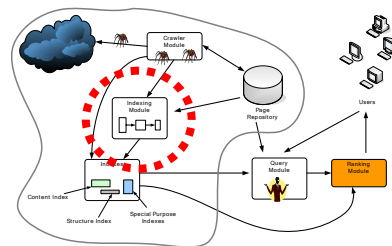
Extracts vital descriptors

- terms, positions, links

Creates compressed version of page

Stores

- Page in cache
- Descriptors in index



Constructing a Full-text Index [Witten 2007]

word	position in text
be	2 6 ...
is	8 ...
not	4 ...
or	3 ...
question	10 ...
that	7 ...
the	9 ...
to	1 5 ...

(a) The beginning of the index.

1	2	3	4	5	6	7	8	9	10
to be or not to be that is the question . . .									

(b) The text.

Figure 4.3 Making a full-text index.

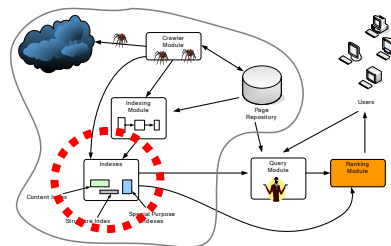
Indexes

Content Index

Structure Index

Special Purpose Index

- Document Formats (PDF, Doc, ...)
- Media (Images, Video, ...)



Indexes

Content Index

- Inverted Document Index
 - term x -> <d11>, <d28>, <d31>, ...
 - term y -> <d10>, <d35>, <d36>, ...
- Index is a
 - quick lookup table
 - smaller than documents

Structure Index

- Hyperlink Information
- In-links, out-links & self-links
- Stored for ...
 - Later analysis
 - Later queries (who links to whom)

Ranking Module

- Orders set of relevant pages
 - Input from query module
- Employs **ranking algorithm**
 - Based on several aspects (terms, links, etc.)
 - Overall score is combination of
 - Content score (TF*IDF)
 - Popularity score (PageRank, HITS, etc.)

Popularity Ranking

- 2 Algorithms developed independently
 - PageRank, Brin & Page
 - Hypertext Induced Topic Search (HITS), Kleinberg
- Basic idea of popularity
 - Someone likes a page
 - Gives a recommendation (on another page)
 - Using a hyperlink

Popularity Ranking: Basic Idea

There are different types of people:

- Regarding their idea of recommendation
 - People giving a lot of recommendations (links)
 - People giving few recommendations (links)
- Regarding their state of recommendation
 - Recommended by a lot of people
 - Recommended by few people

Combinations are possible:

- Having no recommendation, but recommending a lot, ...

Popularity Ranking: Basic Idea

Think of

people as pages

recommendations as links

PageRank (Google)

Therefore:

“Pages are popular, if popular pages link them”

“PageRank is a global ranking of all web pages, regardless of their content, based solely on their location in the Web’s graph structure.” [Page et al 1998]

A Tangled Web [Witten 2007]

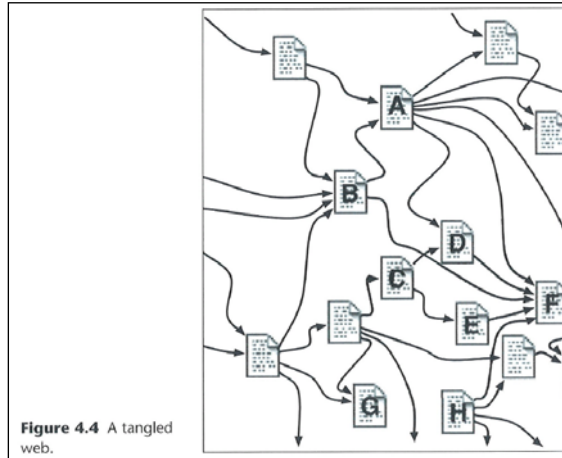
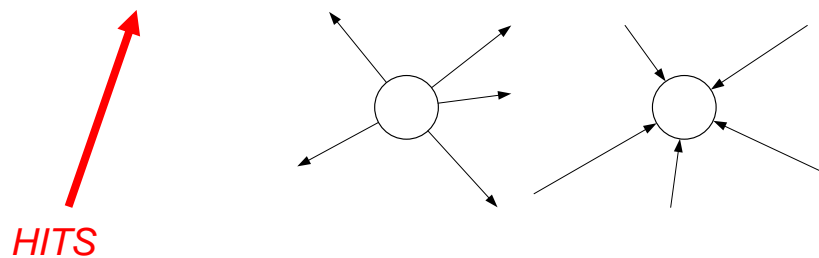


Figure 4.4 A tangled web.

Popularity Ranking: Basic Idea

Additional assumptions:

- **Hubs** are pages that point to highly ranked vertices
- **Authorities** are pages, which are pointed to by highly ranked vertices



PageRank: Original Summation Formula

Original summation formula

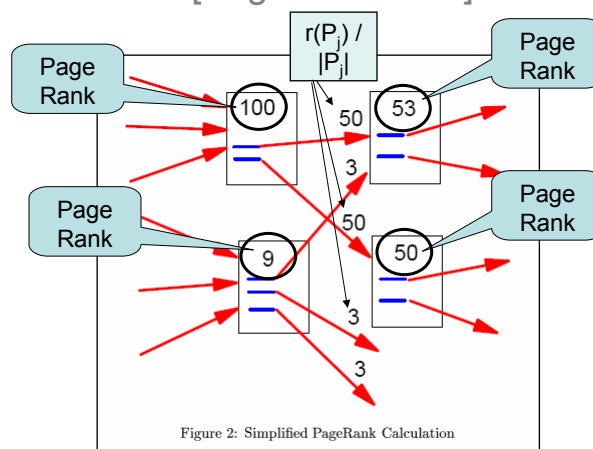
- PageRank of page P_i is given by the summation of: all pages P_j that link to P_i given by the set B_{P_i} *divided by the set of outbound links of P_j : $|P_j|$*

$$r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|}$$

Iterative formula, starting with rank $1/n$ for all n pages:

$$r_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \frac{r_k(P_j)}{|P_j|}$$

PageRank: Original Summation Formula [Page et al 1998]



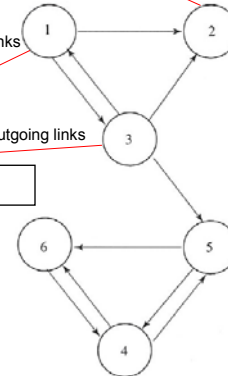
PageRank: Original Summation Formula [Amy N. Langville and Carl D. Meyer 2004]

$$r_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \frac{r_k(P_j)}{|P_j|}$$

$$r_1(P_2) = (1/6)/2 + (1/6)/3 = ?$$

two outgoing links

three outgoing links

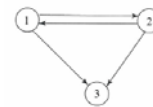


Iteration 0	Iteration 1	Iteration 2	Rank at Iter. 2
$r_0(P_1) = 1/6$	$r_1(P_1) = 1/18$	$r_2(P_1) = 1/36$	5
$r_0(P_2) = 1/6$	$r_1(P_2) = ?$	$r_2(P_2) = 1/18$	4
$r_0(P_3) = 1/6$	$r_1(P_3) = 1/12$	$r_2(P_3) = 1/36$	5
$r_0(P_4) = 1/6$	$r_1(P_4) = 1/4$	$r_2(P_4) = 17/72$	1
$r_0(P_5) = 1/6$	$r_1(P_5) = 5/36$	$r_2(P_5) = 11/72$	3
$r_0(P_6) = 1/6$	$r_1(P_6) = 1/6$	$r_2(P_6) = 14/72$	2

Initial Problems

Rank sinks & cycles:

- Some pages get all of the score, other pages none
- Cycles just flip the rank
- Some nodes do not have outlinks:
Dangling nodes



How many iterations?

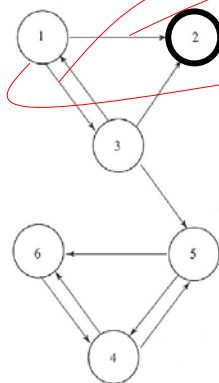
- Will the process converge?
- Will it converge to one single vector?

Approach of Brin & Page

Notion of the random surfer

- Someone navigates through the web using hyperlinks
- If there are 6 links, there is a probability of 1/6 that s/he takes a specific link
- On dangling nodes (without out links) s/he can jump everywhere with equal chance
- Furthermore s/he can leave the link path with a given probability every time

Approach of Brin & Page: Example taken from [Amy N. Langville and Carl D. Meyer 2004]



$$\mathbf{H} = \begin{matrix} & \begin{matrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \end{matrix} \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} & \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

replace all zero rows, 0^T , with $1/ne^T$, where e^T is the row vector of all ones and n is the order of the matrix.



Dealing with dangling nodes

$$\mathbf{S} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Leaving the link structure: [Amy N. Langville and Carl D. Meyer 2004]

Introduction of the Google Matrix: $G = \alpha S + (1 - \alpha)1/n ee^T$

$$H = \begin{matrix} & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} & \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

$$G = .9H + (.9 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + .1 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}) 1/6(1 \ 1 \ 1 \ 1 \ 1 \ 1)$$

$$= \begin{pmatrix} 1/60 & 7/15 & 7/15 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 19/60 & 19/60 & 1/60 & 1/60 & 19/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 7/15 & 7/15 \\ 1/60 & 1/60 & 1/60 & 7/15 & 1/60 & 7/15 \\ 1/60 & 1/60 & 1/60 & 11/12 & 1/60 & 1/60 \end{pmatrix}$$

Brin and Page suggested a damping factor $\alpha = 0.85$
 „That means, roughly five-sixths of the time a web surfer randomly clicks on hyperlinks (i.e. following the structure of the web) while one-sixth of the time this web surfer will go to the URL line and type the address of a new page to „teleport“ to.“

Every node is now directly connected to every other node, making the chain irreducible by definition.

The Google Matrix Step by Step

$$H = \begin{matrix} & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} & \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

$$G = .9H + (.9 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + .1 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}) 1/6(1 \ 1 \ 1 \ 1 \ 1 \ 1)$$

$$0.9 \cdot \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 9/20 & 9/20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 3/10 & 3/10 & 0 & 0 & 3/10 & 0 \\ 0 & 0 & 0 & 0 & 9/20 & 9/20 \\ 0 & 0 & 0 & 9/20 & 0 & 9/20 \\ 0 & 0 & 0 & 9/10 & 0 & 0 \end{pmatrix}$$

The Google Matrix Step by Step

$$G = .9H + (.9 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + .1 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}) \cdot \frac{1}{6} (1 \ 1 \ 1 \ 1 \ 1 \ 1)$$

$$\begin{pmatrix} 0 \\ 9/10 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1/10 \\ 1/10 \\ 1/10 \\ 1/10 \\ 1/10 \\ 1/10 \end{pmatrix} = \begin{pmatrix} 1/10 \\ 10/10 \\ 1/10 \\ 1/10 \\ 1/10 \\ 1/10 \end{pmatrix}$$

The Google Matrix Step by Step

$$G = .9H + (.9 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + .1 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}) \cdot \frac{1}{6} (1 \ 1 \ 1 \ 1 \ 1 \ 1)$$

$$\begin{pmatrix} 1/10 \\ 10/10 \\ 1/10 \\ 1/10 \\ 1/10 \\ 1/10 \end{pmatrix} \cdot \frac{1}{6} (1 \ 1 \ 1 \ 1 \ 1 \ 1) = \begin{pmatrix} 1/60 & 1/60 & 1/60 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/60 & 1/60 \end{pmatrix}$$

The Google Matrix Step by Step

$$G = .9H + (.9 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + .1 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}) \frac{1}{6} (1 \ 1 \ 1 \ 1 \ 1 \ 1)$$

$$G = \begin{bmatrix} 0 & 9/20 & 9/20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 3/10 & 3/10 & 0 & 0 & 3/10 & 0 \\ 0 & 0 & 0 & 0 & 9/20 & 9/20 \\ 0 & 0 & 0 & 9/20 & 0 & 9/20 \\ 0 & 0 & 0 & 9/10 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1/60 & 1/60 & 1/60 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/60 & 1/60 \end{bmatrix} = \begin{bmatrix} 1/60 & 7/15 & 7/15 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 19/60 & 19/60 & 1/60 & 1/60 & 19/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 7/15 & 7/15 \\ 1/60 & 1/60 & 1/60 & 7/15 & 1/60 & 7/15 \\ 1/60 & 1/60 & 1/60 & 11/12 & 1/60 & 1/60 \end{bmatrix}$$

Result of the adaptations [Amy N. Langville and Carl D. Meyer 2004]

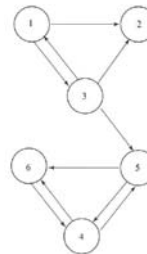
Iterative Formula $\pi^{(k+1)T} = \pi^{(k)T} G,$

- Converges to a single PageRank vector

In our example:

$$\pi^T = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ .03721 & .05396 & .04151 & .3751 & .206 & .2862 \end{pmatrix}$$

taken from "Google's PageRank & Beyond", Langville & Meyer



Retrieval Evaluation: Motivation

Objectively compare different

- Search engines
- Models & Weighting Schemes
- Methods & Techniques

Scope

- Academic
- Commercial & Industrial

Axis

- Runtime, Retrieval performance

Retrieval Evaluation

Approaches since first prototypes differ in:

- Test collections
- Experts assessing retrieval performance
- Metrics
 - What's good? / What's bad?

Overall problem:

- What is relevant?

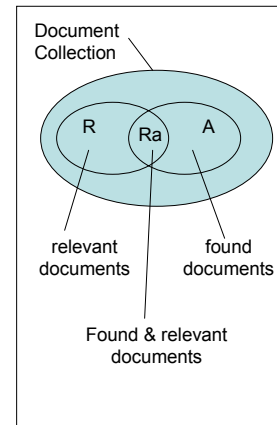
Metrics: Precision & Recall

Within a document collection D with a given query q

$|R|$.. num. of relevant docs

$|A|$.. num. of found docs

$|Ra|$.. num. found & relevant



Metrics: Precision

$$\text{Precision} = \frac{|Ra|}{|A|} = \frac{\text{found relevant docs}}{\text{found docs}}$$

Gives % how many of the actual found documents have been relevant

Between 0 and 1

– Optimum: 1 ... all found docs are relevant

Metrics: Recall

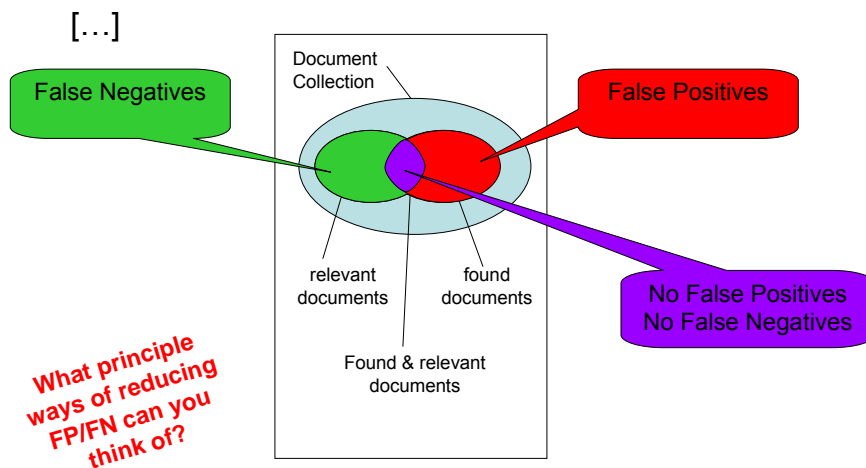
$$\text{Recall} = \frac{|Ra|}{|R|} = \frac{\text{found relevant docs}}{\text{relevant docs}}$$

Gives % how many of the actual relevant documents have been found

Between 0 and 1

- Optimum: 1 ... all relevant docs are found

False Positives and False Negatives



Examples: Precision & Recall

With a query only 1 document has been found, but this one is relevant (100 would be relevant):

- Precision & Recall?
- **Precision = 1**
- **Recall = 0,01**

With a query all documents of D have been found (5% of D would be relevant)

- Precision & Recall?
- **Precision = 0,05**
- **Recall = 1**

Recall vs. Precision Plot

Assumption:

- Result list is sorted by descending relevance
- User investigates result list linearly
 - Precision and Recall change

Approach:

- Map different states to graph

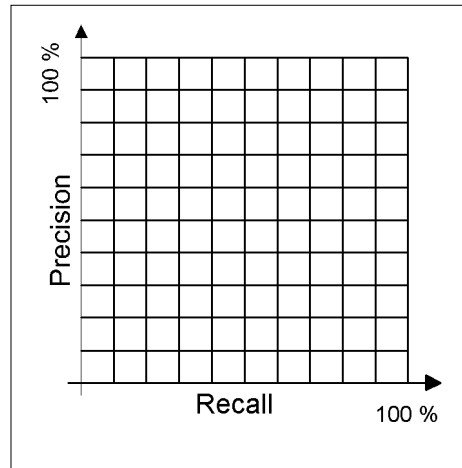
Recall vs. Precision Plot

Result Set:

- 01. d123 * 06. d9 * 11. d38
- 02. d84 07. d511 12. d48
- 03. d56 * 08. d129 13. d250
- 04. d6 09. d187 14. d113
- 05. d8 10. d25 * 15. d3 *

Relevant Results:

Rq={d3, d5, d9, d25, d39, d44, d56, d71, d89, d123} → Σ 10



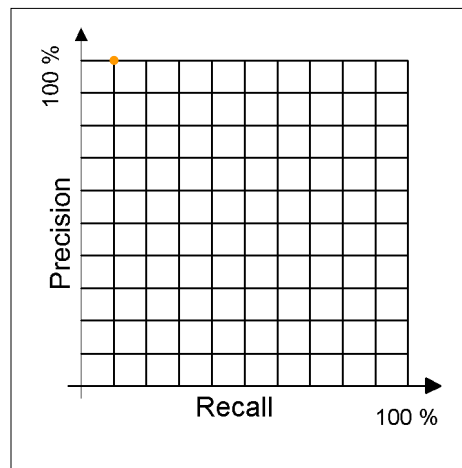
Recall vs. Precision Plot

- 01. d123 * 06. d9 * 11. d38
- 02. d84 07. d511 12. d48
- 03. d56 * 08. d129 13. d250
- 04. d6 09. d187 14. d113
- 05. d8 10. d25 * 15. d3 *

11 Standard Recall Levels
 {0%, 10%, 20%, ..., 90%, 100%}

$$\text{Recall} = \frac{|Ra|}{R} = \frac{1}{10}$$

$$\text{Precision} = \frac{|Ra|}{A} = \frac{1}{1}$$

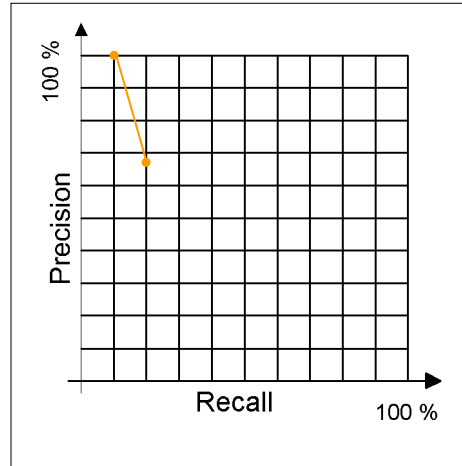


Recall and Precision

- | | | |
|------------|-----------|----------|
| 01. d123 * | 06. d9 * | 11. d38 |
| 02. d84 | 07. d511 | 12. d48 |
| 03. d56 * | 08. d129 | 13. d250 |
| 04. d6 | 09. d187 | 14. d113 |
| 05. d8 | 10. d25 * | 15. d3 * |

$$\text{Recall} = \frac{|Ra|}{R} = \frac{2}{10}$$

$$\text{Precision} = \frac{|Ra|}{A} = \frac{2}{3}$$

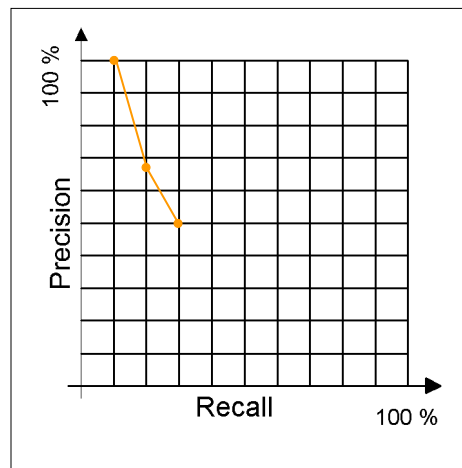


Recall and Precision

- | | | |
|------------|-----------|----------|
| 01. d123 * | 06. d9 * | 11. d38 |
| 02. d84 | 07. d511 | 12. d48 |
| 03. d56 * | 08. d129 | 13. d250 |
| 04. d6 | 09. d187 | 14. d113 |
| 05. d8 | 10. d25 * | 15. d3 * |

$$\text{Precision} = ?$$

$$\text{Recall} = ?$$

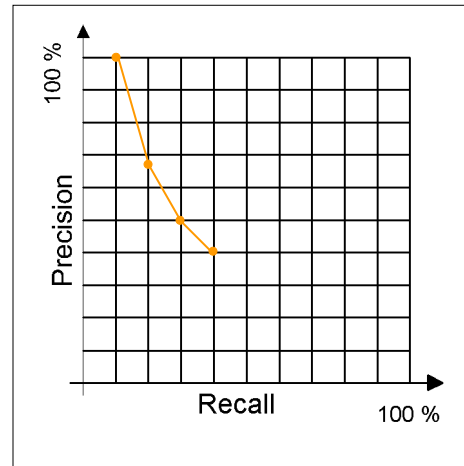


Recall and Precision

- | | | |
|------------|-----------|----------|
| 01. d123 * | 06. d9 * | 11. d38 |
| 02. d84 | 07. d511 | 12. d48 |
| 03. d56 * | 08. d129 | 13. d250 |
| 04. d6 | 09. d187 | 14. d113 |
| 05. d8 | 10. d25 * | 15. d3 * |

Precision =

Recall =



Problems

The Deep Web

What is the deep web?

⇒ Pages crawlers do not currently find.

Example: <http://www.aekstmk.or.at/>

Communications of the ACM

Volume 50, Number 5 (2007), Pages 94-101

“Accessing the deep web”, Bin He, Mitesh Patel, Zhen Zhang, Kevin Chen-Chuan Chang

Problems

Spam

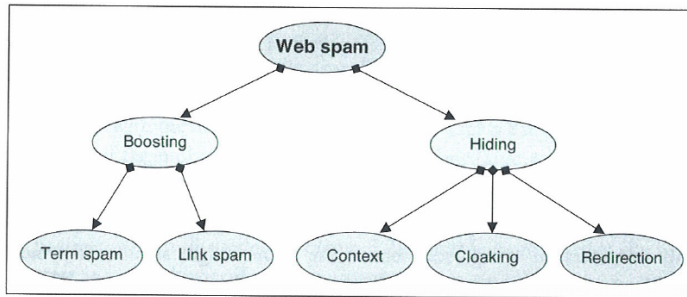


Figure 5.1 The taxonomy of web spam.

Any questions?

See you next week!