Introduction to Information Retrieval

Carsten Eickhoff

TU Delft
Delft University of Technology

CWI

What is it all about?

1 Introduction

- Google and other Web search providers are on the rise for more than 10 years
- They advance to take the role of "media all-rounders"
- On Coursera, you are learning about the range of products offered by Google
- In this lecture, we will try and understand **how** these services are facilitated

IR Applications

- Many on-line activities involve IR technology
  - Media consumption (music, videos, text, …)
  - News tracking
  - Social networking
  - Online shopping
  - Advertisement
  - Mobile communication

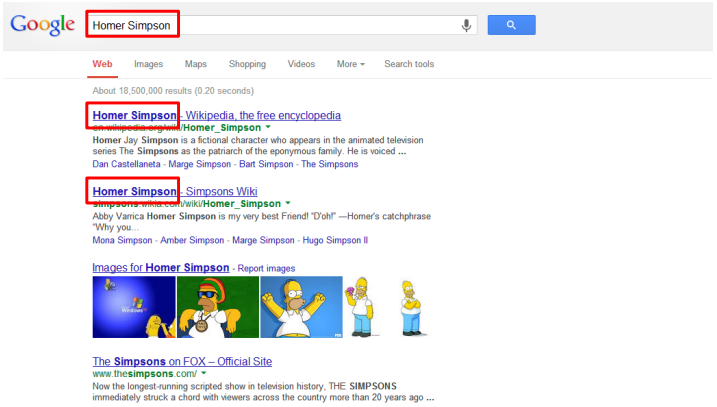
Agenda

1. Introduction
2. IR Technology
   a) Crawling
   b) Indexing & Storage
   c) Retrieval
   d) Data-driven Technologies
3. IR for Children
4. Recommended Reading

Section 2

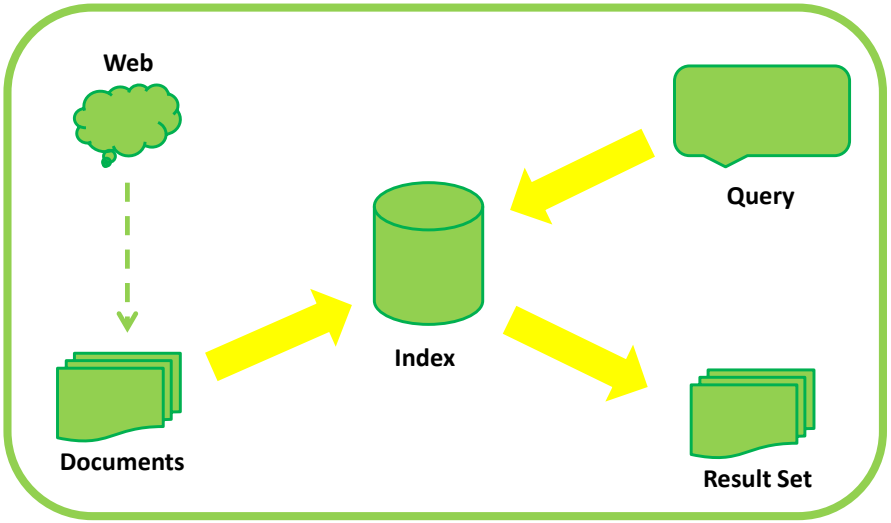
IR TECHNOLOGY

Searching and Term Matching



- Search engines do not understand queries!
- They only find things that are similar

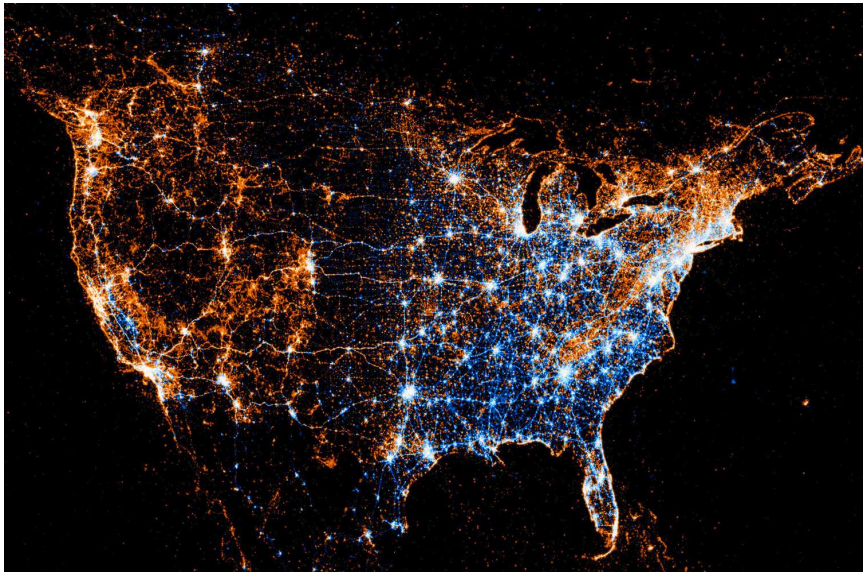
Overview Web Search



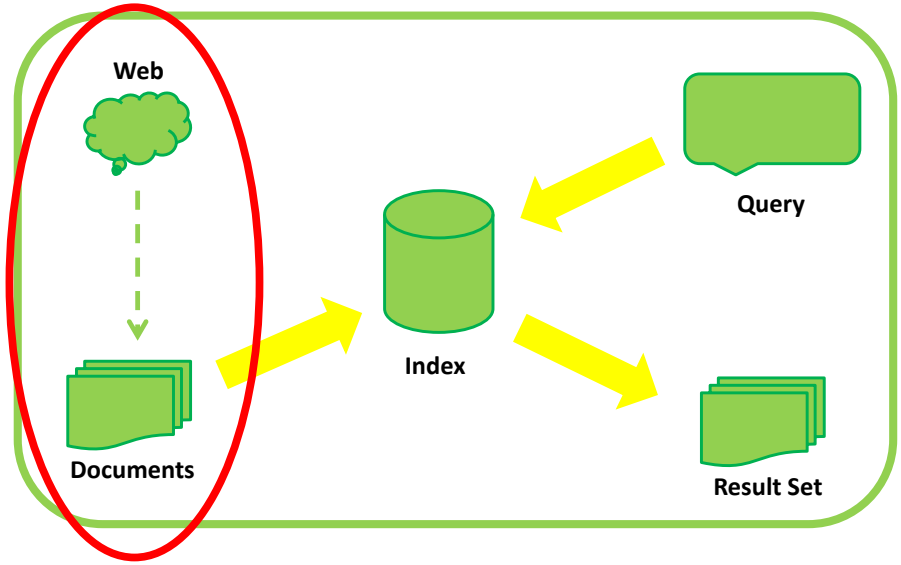
Section 2.1

CRAWLING

How does the Engine know what is out there?



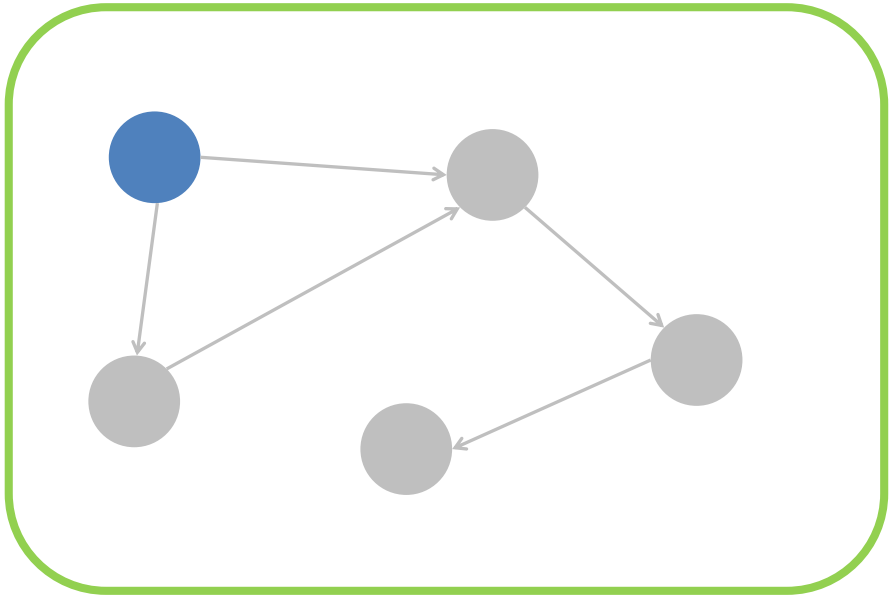
Overview Web Search



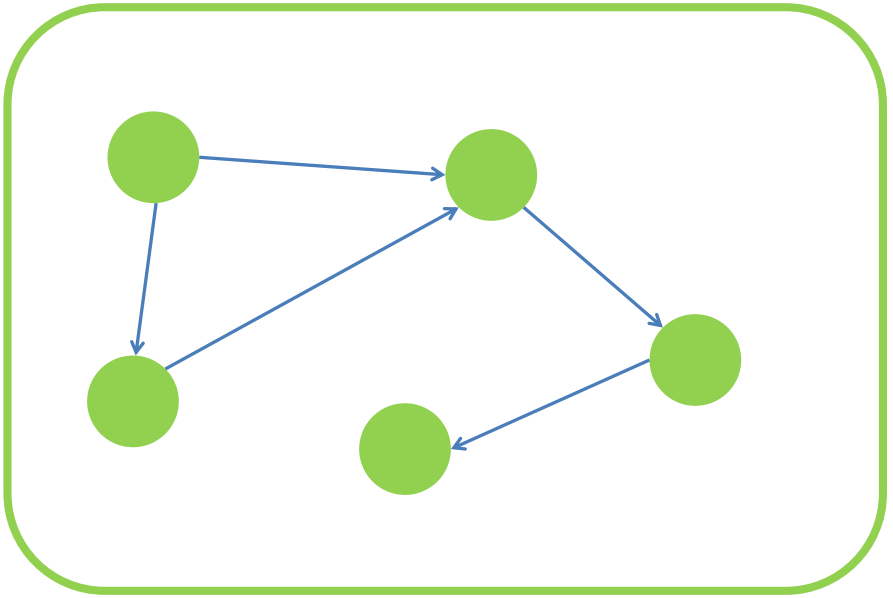
Spiders, Crawlers, Robots…

- Search engine companies send hordes of light-weight programs through the web
- Whenever they encounter a new or changed Web page, they save it.
- From there, they follow the page's outgoing links

The Crawling Process



The Crawling Process



Crawling the Web

- Web crawling is an ongoing process
- To keep up with the pace at which information on the Web changes, pages are re-crawled all the time
- Search engine providers have to make sure that their crawlers are polite

The Ethics of Saving Everything…

- Large shares of the Web are of private or proprietary nature
- Crawlers should not visit those
- “robots.txt” regulates simple access rules for each Web server

Access Policy Examples

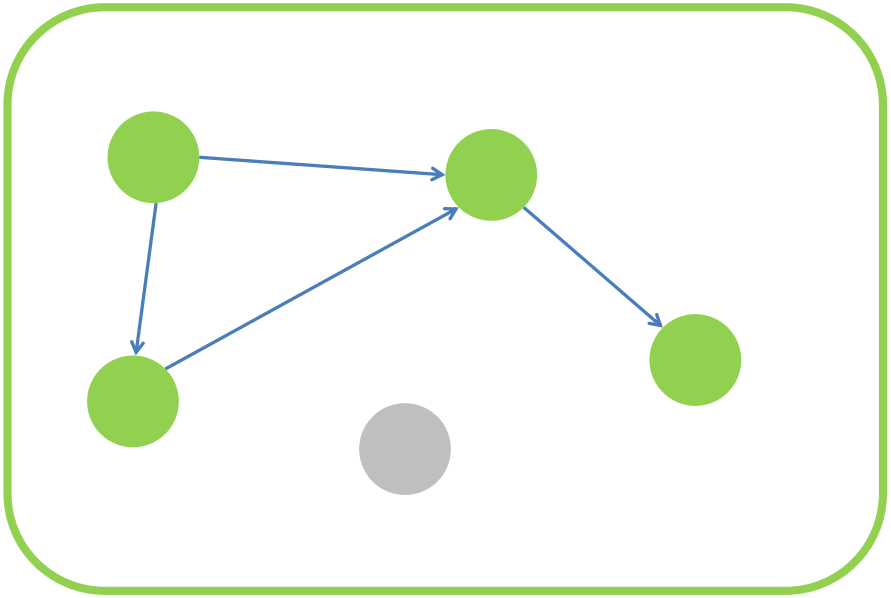
- Examples of robot policies in robots.txt

Example 1:
User-agent: *
Disallow: /cyberworld/map/

User-agent: cybermapper
Disallow:

Example 2:
User-agent: *
Disallow: /

The Hidden Web



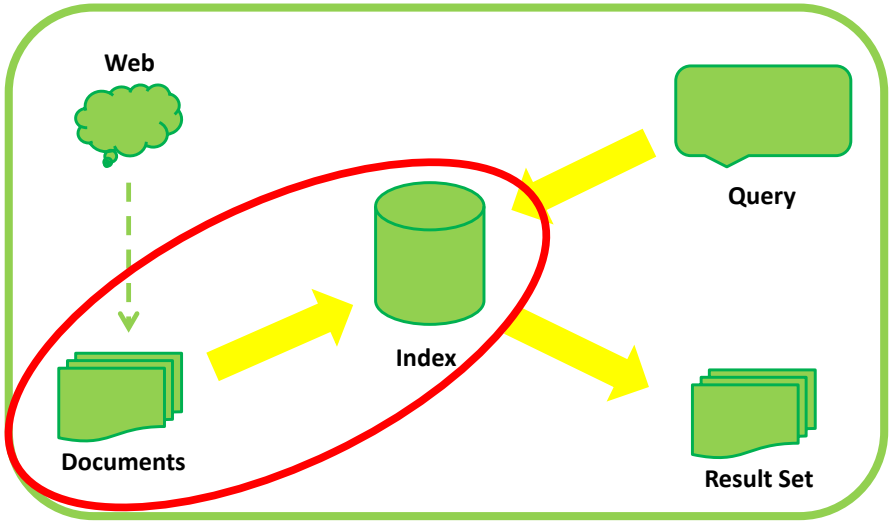
The Deep Web

- Similar to the hidden Web, pages that only become available after user interaction
- E.g., product pages in a Web shop
- Some crawlers can fill in information in forms and explore the underlying pages
- But be careful, or your crawler buys a cruise ship tour…

Section 2.2

INDEXING & STORAGE

Overview Web Search



Index Construction

- Naïve Approach:
  - Make a copy of each page
  - When a query comes in, look for the terms in all saved pages
- But wait, is that clever?
  - We need to answer queries in half a second…

Inverted Index

- Turn the problem around
- Build an inverted index that tells us which document contains the query terms
- At query time, we only consider those documents that contain at least one query term

Example: Inverted Index

he	drink	ink	likes	pink	thing	wink	
2	1	0	2	0	0	1	He likes to wink, he likes to drink.
1	3	0	1	0	0	0	He likes to drink, and drink, and drink.
1	1	1	1	0	1	0	The thing he likes to drink is ink.
1	1	1	1	1	0	0	The ink he likes to drink is pink.
1	1	1	1	1	0	1	He likes to wink and drink pink ink.

Example: Sparse Index

- Most fields of the matrix would be empty
- So we store it as a sparse matrix

he	D1:2	D2:1	D3:1	D4:1	D5:1
ink	D3:1	D4:1	D5:1		
pink	D4:1	D5:1			
thing	D3:1				
wink	D1:1	D5:1			

Positional Indices

- The inverted index allows us to know which terms are in which documents
- But we lose position information
- Term proximity may be important

Query: "Square dance"

Doc 1: "…Times Square in new York is often host to dance performances..."

Doc 2: "… Square dance is a dance for four couples…"

Example: Positional Index

- To preserve position information, we store each occurrence as a tuple <doc,pos>

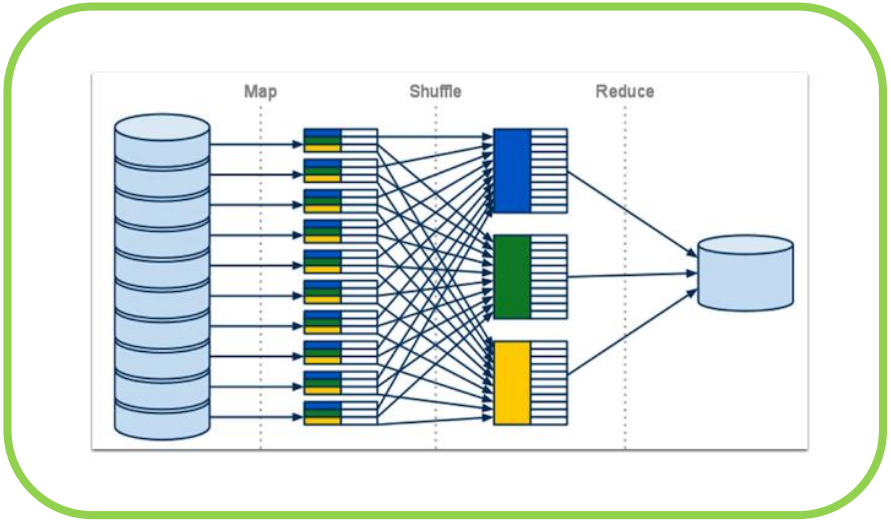
he	D1:1	D1:5	D2:1	D3:3	D4:3	D5:1
ink	D3:8	D4:2	D5:8			
pink	D4:8	D5:7				
thing	D3:2					
wink	D1:4	D5:4				

What if the index becomes too large?

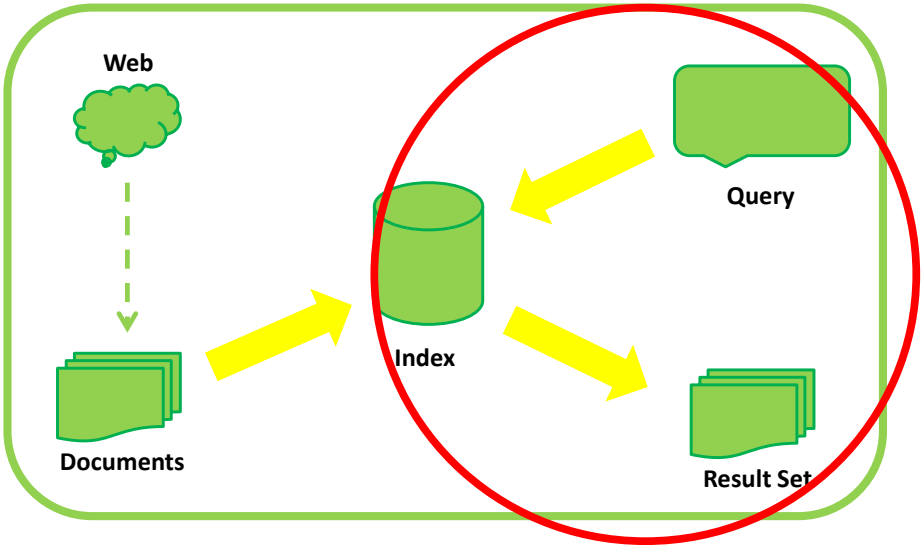
Distributed Indices

- If we cannot fit the index on a single machine any more, we split it up
- Such distributed architectures introduce new problems
  - Crawling
  - Indexing
  - Retrieval

Map Reduce



Overview Web Search



Section 2.3

RETRIEVAL

Boolean Retrieval

- The early days (1960's-80's)
- Searching in highly specialized collections
  - Legal or medical documents
  - Newspaper archives
- Searcher is a trained professional

Boolean Queries

- Complex queries in expert syntax describe information need (E.g., Westlaw)

Information need:
Information on the legal theories involved in preventing the disclosure of trade secrets by employees formerly employed by a competing company.

Query:
"trade secret" /s disclos! /s prevent /s employe!

Information need:
Requirements for disabled people to be able to access a workplace.

Query:
disab! /p access! /s work-site work-place (employment /3 place)

Boolean Retrieval Summary

\+

- Intuitive
- Gives searcher direct control over the retrieval process

\-

- High cognitive load during query formulation and result set exploration
- No ordering to result sets
- Requires expert syntax

Ranked Retrieval

- Introduced in the early 1990's
- Returns documents in their order of relevance
- New problem: How to determine relevance?
- Retrieval model computes one relevance score per document and sorts accordingly

Vector Space Models

- Geometric model
- Idea:
  - Relevant documents are similar to the query
  - The more similar they are, the higher they are ranked

Vector Space Mappings

- Translate text documents into vectors

Document Text:
"IT WAS the best of times, it was the worst of times…"

Binary Vector:

it	was	the	best	of	times	worst	cat
1	1	1	1	1	1	1	0

Frequency-based Vector:

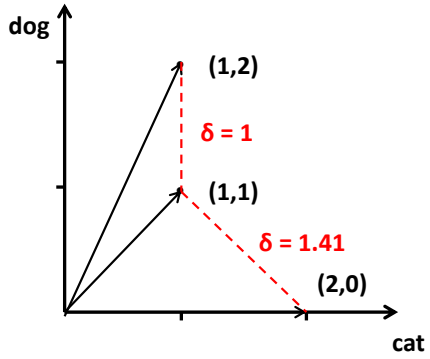
it	was	the	best	of	times	worst	cat
2	2	2	1	2	2	1	0

Documents in Space

- Each vector is a point in a high-dimensional space
- We tend to stop drawing at 3 dimensions, but 10,000 are also no problem mathematically ;-)
- Recall: Our document index already looks like a vector!

Distances in Vector Spaces

- Similar documents lie close together



Probabilistic Models

- Alternatively, use probabilistic methods
- For each document, compute the probability of relevance towards the query
- Rank by probability

Building Document Models

- Build a language model of each document
- Count term frequencies and divide by document length

he	drink	ink	likes	pink	thing	wink	the	to	and	is	
$\frac{2}{8}$	$\frac{1}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{1}{8}$	$\frac{0}{8}$	$\frac{2}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	He likes to wink, he likes to drink.
$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{0}{8}$	$\frac{1}{8}$	$\frac{0}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{0}{8}$	$\frac{1}{8}$	The thing he likes to drink is ink.
$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{0}{8}$	$\frac{1}{8}$	$\frac{0}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{0}{8}$	He likes to wink and drink pink ink.

Ranking Documents

- Compute the likelihood of the query being generated by the document models
- Order documents by decreasing probability

Query: “he wink”

M1: $\boldsymbol{P}(\text{he}) * \boldsymbol{P}(\text{wink}) = \frac{2}{8} * \frac{1}{8} = \frac{1}{32}$

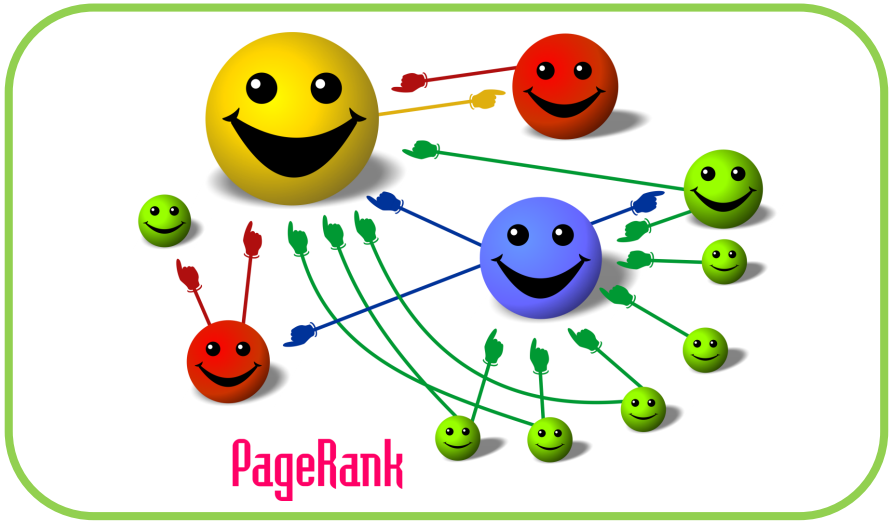
M2: $\boldsymbol{P}(\text{he}) * \boldsymbol{P}(\text{wink}) = \frac{1}{8} * \frac{0}{8} = 0$

M3: $\boldsymbol{P}(\text{he}) * \boldsymbol{P}(\text{wink}) = \frac{1}{8} * \frac{1}{8} = \frac{1}{64}$

PageRank

- Google’s fabled original ranking criterion
- Based on the idea that page authoritativeness should be rewarded during ranking
- Authoritativeness scores are iteratively propagated along hyperlinks

Example: PageRank



Section 2.4

DATA-DRIVEN TECHNOLOGIES

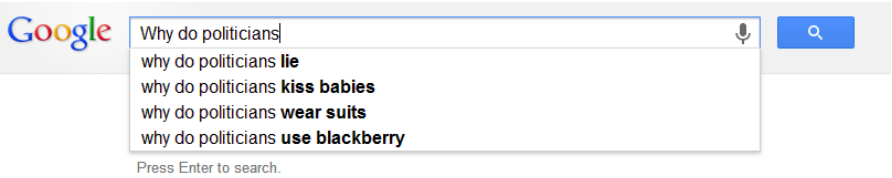
Everyone wants your Data.
But what do they do with it?

Search Personalization

Search Personalization

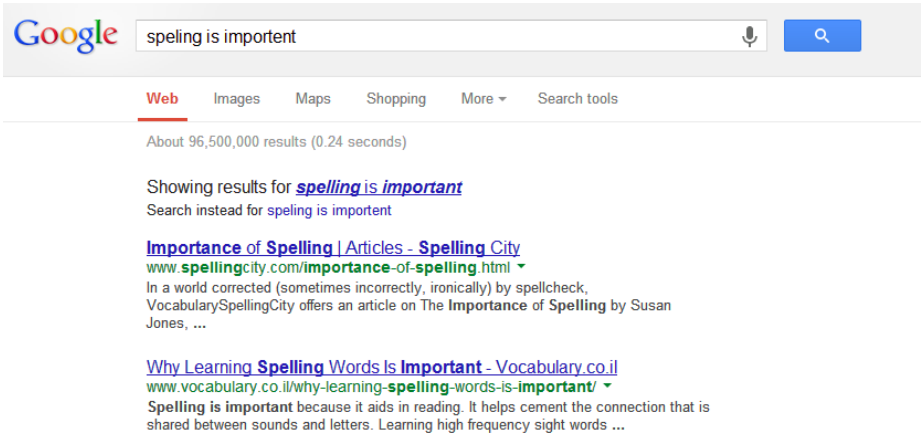
- People have very specific preferences and interests (Topics, language, textual complexity, location, etc.)
- Based on your previous search history, we can find out about these preferences
- For future searches, the engine tries to optimize for the newly found preference

Query Suggestion



- Query suggestions try to help people formulate their information needs
- They are selected on the basis of frequent queries that extend the current query terms

Spelling Correction



Data-driven Spelling Correction

- Idea: Consensus is strong / errors are random

Spelling	Frequency
albert einstein	4834
albert einstien	525
albert einstine	149
albert einsten	27
albert einsteins	25
albert einstain	11
albert einstin	10
albert eintein	9

Dangers: Welcome to the Filter Bubble

- Data-driven methods are very powerful
- But what happens to the niche information need?
- What if I want to see that video that nobody else likes?
- Diversification techniques can help but there is a danger of drowning in the mainstream

Section 3

IR FOR CHILDREN

Meet the Users

The PuppyIR Project

- European Union research project on child-friendly information access
- Investigate the specific needs of children
- Create an open-source framework to cater for these needs
- 5 Universities, 3 Business partners

The Internet – A Place full of Kids

- Children interact with the Internet at a younger age (~ 4 years old)
- They spend more time online
- ~ 40% of British 10-year-olds have regular unsupervised Internet access

The Internet – A Place for Kids?

- Many media platforms are mirrors of popular culture
- Large parts of their content might not be suitable for children
- Most IR systems are designed with adult users in mind, not children

How to learn what Children Need?

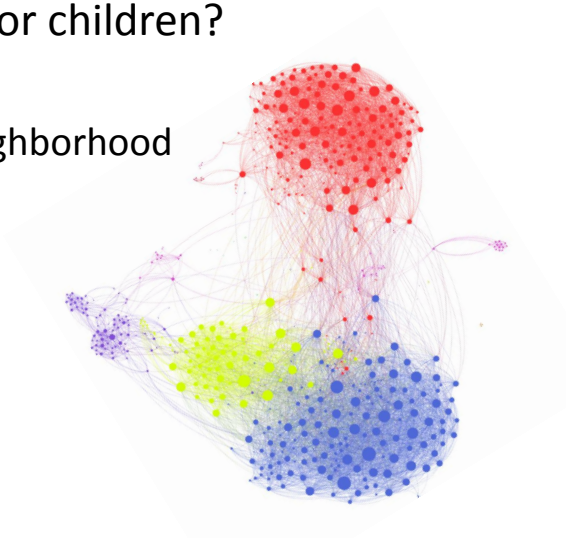
- Literature study
- Surveys
  - 300 US parents and teachers
- User studies
  - 49 Dutch elementary school children
- Query log analyses
  - Thousands of young Yahoo users

Children's Deficits

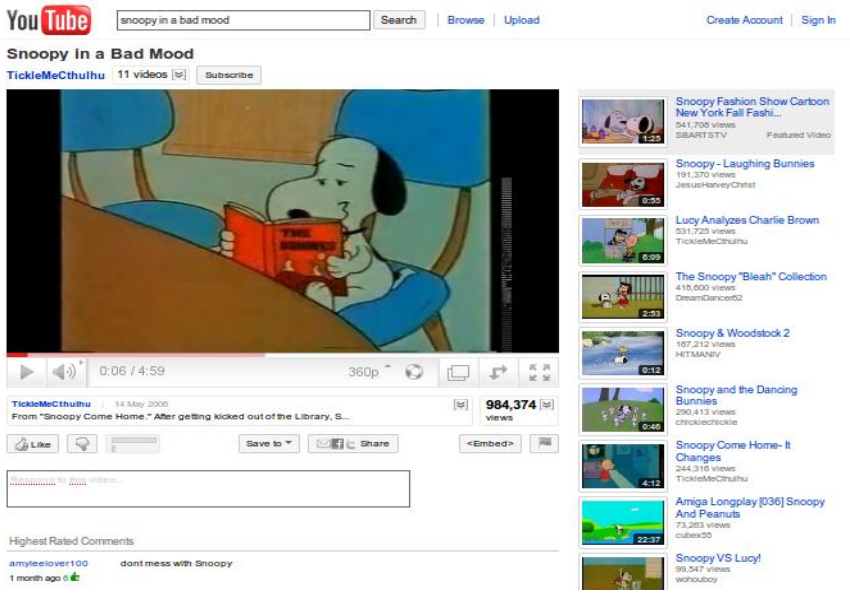
- They type / spell badly
- They browse rather than search
- They are bad at keywording
- They struggle with search interfaces
- Everything is relevant
- Complex content is a challenge

Web Page Classification

- Can we automatically determine whether a web site is suitable for children?
  - Based on its content
  - Based on its link neighborhood



Video Classification



Content Simplification

- Sometimes we do not want to completely exclude certain documents
- But they still contain complex language that is hard to grasp for kids
- In these cases, we can automatically offer simplifications for difficult/technical terms

Demo: Museon/Gemeentemuseum

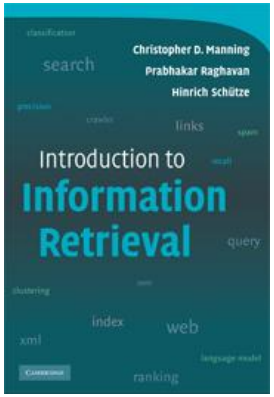
Demo: Emma Kinderziekenhuis

Section 4

RECOMMENDED READING

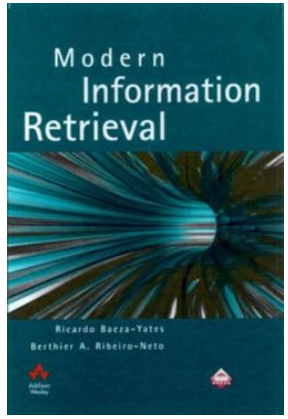
Introduction to IR

- Manning, Raghavan, Schutze
- Available online at:
- http://nlp.stanford.edu/IR-book/
- All-in-one you need to know



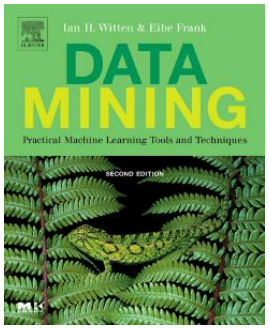
Modern Information Retrieval

- Baeza-Yates, Ribeiro-Neito
- Good overview over all basic topics



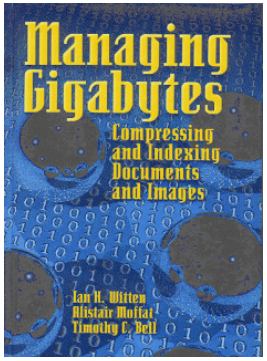
Data Mining

- Witten, Frank
- Data mining and pattern recognition essentials



Managing Gygabytes

- Witten, Moffat, Bell
- What to do when your data gets large?



Speech and Language Processing

- Jurafsky, Martin
- Everything you ever want to know about language processing