Summary: EITN01 Web Intelligence and Information Retrieval

Anders Ardö

EIT – Electrical and Information Technology, Lund University

March 13, 2013

Summary agenda

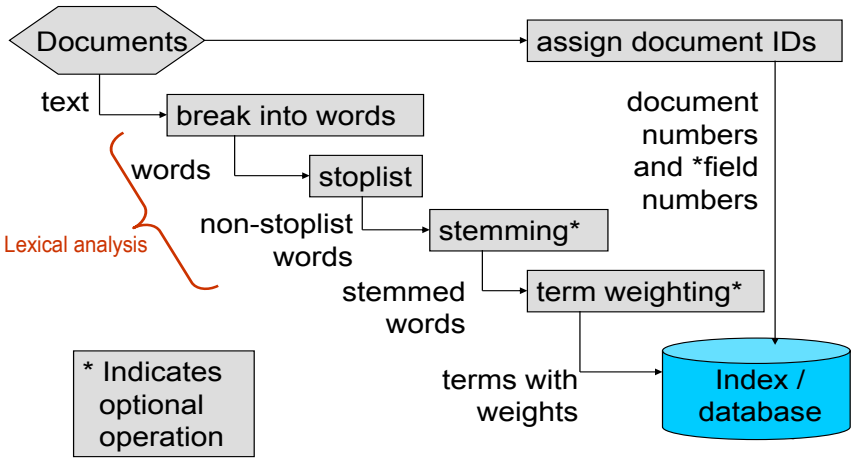
Outline

Lecture 1

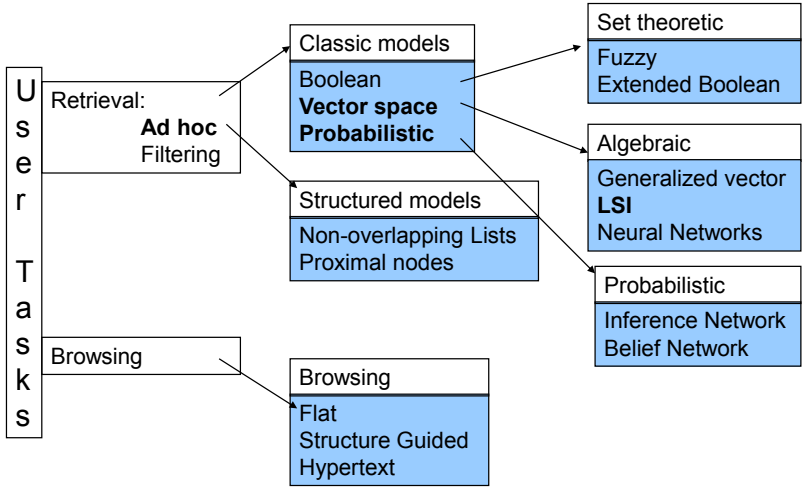
Information Retrieval - IR

- Indexing
- IR models
- Weighting
- Evaluation

Representation/Indexing (fig 1.2)



IR models - overview (fig 2.1)

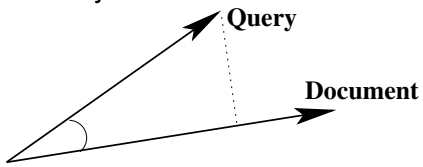


IR models - vector space

Bag-Of-Words:

- syntax irrelevant
- document structure irrelevant
- meta-information irrelevant
- document/query = n-dimensional vector

similarity = cosine between vectors



Weighting TF*IDF

Terms should be important in the document
Terms present in many documents are not important

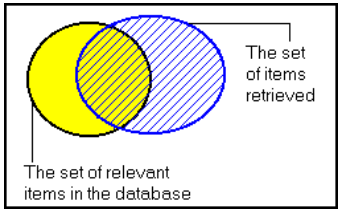
$\Downarrow$

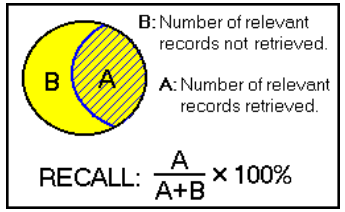
(importance in document) * (how rare is the term)
tf * *idf*

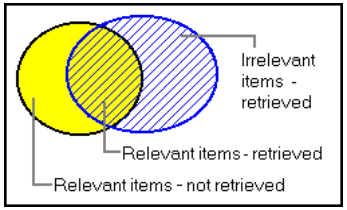
tf term frequency
idf inverse document frequency

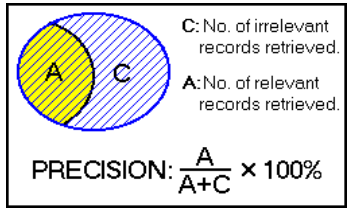
Various normalizations

Recall/Precision









Outline

Lecture 2

- Query languages/protocols
- Link based ranking

Query languages - aspects

- keyword
- context
- Boolean
- natural language
- pattern matching
  - truncation
  - wild cards
  - regular expressions
  - fuzzy search
- structured
  - fields

Query languages

- Expressiveness
- Standardization
- Examples
  - Structured Query Language - SQL
  - Common Query Language - CQL
  - XQuery
  - Proprietary/home grown
    - Google

Query operations

- relevance feedback
  - Give me more like this
  - New Relevance Feedback vector =
    Original query vector
    + mean of relevant documents (vectors) in hit set
    - mean of non-relevant documents (vectors) in hit set
- query expansion
  - Add or remove search terms
  - Change Boolean operators
  - Change wild cards

Network protocols/queries

- Dedicated protocol
  - Z39.50
  - MySQL network protocol
- CGI-script - Web Common Gateway Interface
- Web services
- Search/Retrieval via URL - SRU
- OpenSearch
- Open Archives Initiative - OAI

PageRank

$$PR(u)_t = d \sum_{v \in B_u} \frac{PR(v)_{t-1}}{N_v} + (1-d)E(u)$$

- Random surfer model
  - Click on a random link in the page
  - Eventually gets bored and jumps to a random page
- Converges to a stable solution
- Problems
  - size of the Web
  - pages without links - 'dangling pages' (rank sinks)
  - converging
  - link-spamming

PageRank + TF*IDF

Relevance ranking

Combine PageRank with vector space model

$$PR(D) * sim(Q, D)$$

or

$$f(PR(D)) * sim(Q, D)$$

In practice

- proximity
- structure: title, link-anchor text
- metadata: keywords, description
- and $\cdots$

Outline

Lecture 3

- Text
  - Unicode character set (UTF-8) $>$ 100000 characters
  - Zipf's law $\Rightarrow$ Skewed distribution - stopwords
  - Heaps' law: $Vocabulary \sim n^{\beta}$; $\beta < 1$
  - Metadata
    - Author, source, length
    - Dublin Core Metadata Element Set
  - Similarity models: Hamming Distance, Edit (Levenshtein) Distance
- Markup languages
- Text operations

Markup Languages

- SGML - Standard Generalized Markup Language
  - HTML - HyperText Markup Language
  - XML - eXtensible Markup Language
- TEX/ LATEX

Text operations - preprocessing

- 1: Lexical analysis
- 2: Stopwords
- 3: Stemming
- 4: Selection of indexing terms
- 5: Thesaurus

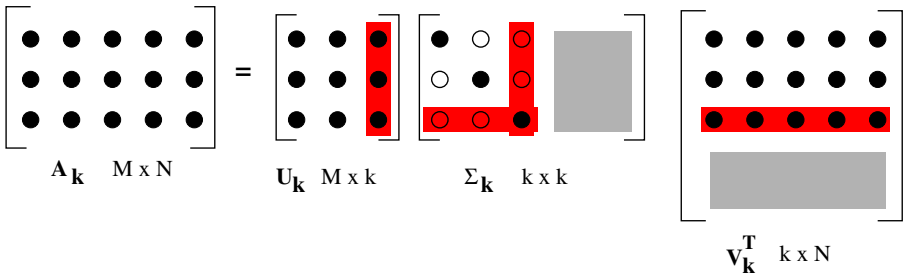
Outline

Lecture 4

- LSI (Latent Semantic Indexing)- concepts
  - The term-document matrix is decomposed into three other matrices of a special form by use of Singular Value Decomposition (SVD)
  - The matrices show a breakdown of the original relationships into linearly independent components
  - Many of these components are very small and can be ignored - leading to an approximate model that contains fewer dimensions.
- SVM (Support Vector Machines) - classification

LSI - reduced SVD

- Reduce dimensionality => retain only k largest singular values
- Saved space



- M x N matrix A (term/document) $\rightsquigarrow$ reduced SVD:

$$A \approx A_k = U_k \Sigma_k V_k^T$$

LSI - Concept extraction

use rows of $\Sigma_k^{-1} U_k^T$ as concepts

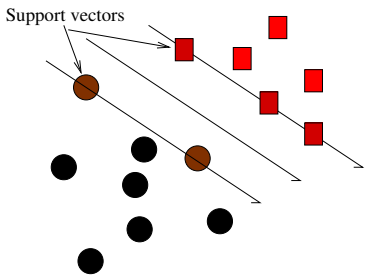
Concept 1		Concept 2	
carlstrom	0.0354	regia	0.0523
rick	0.0354	oct	0.0521
amelnx	0.0354	chrisp	0.0314
advmar	0.0354	problems	0.0273
cuttings	0.0329	pm	0.0265
september	0.0322	ip-forum	0.0264
miller	0.0265	stratification	0.0261
re	-0.0287	uk	-0.0267
wants	-0.0303	bladderwort	-0.0361
aquatic	-0.0363	cuttings	-0.0317
rotundifolia	-0.0371		
bladderwort	-0.0605		

HARD to interpret

Text classification

- Goal: classify documents into predefined categories
- Examples
  - Subject classification: 'business', 'sports', 'engineering', ...
  - Review classification: 'positive' or 'negative'
  - Web page classification: 'Personal homepage' or others
- Approach: supervised machine learning (⇒ SVM)
  - Each predefined category needs a set of training documents
  - From training sets train a classifier
  - Use classifier to classify new documents

SVM



- SVM maximize the margin around the separating hyper-plane
- Decision function specified by support vectors (from training examples)
- Quadratic programming problem

Hot text classification method

Outline

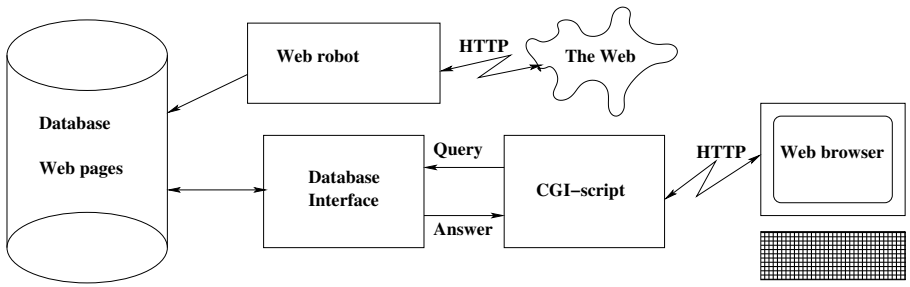
Lecture 5

- Web Search
- Metasearch engines
- Web crawling
- Browsing vs search

Web Search

- Challenges
  - Distributed, dynamic data
  - Large volume
  - Unstructured, heterogeneous data
- Size, coverage
- General vs focused
- Special functions, User interface
- Ranking
- Limited overlap between search engines

Search Engine - Basic structure



Size efficiency response time

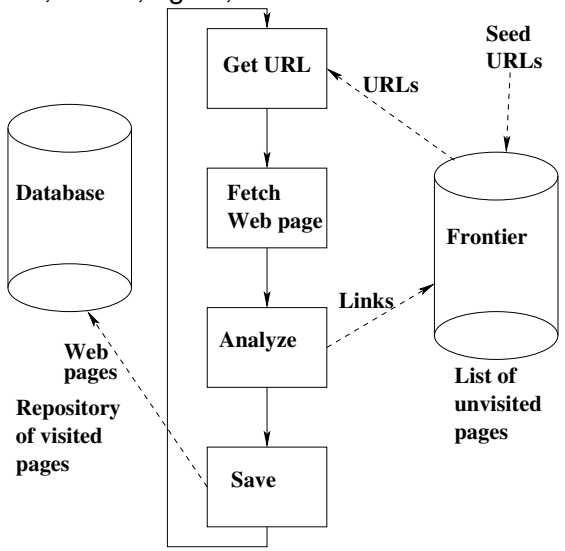
- software crawling the web (much like a human clicking on links)
- collect all found web-pages into a database (IR system)
- offer a web-interface to that database

Metasearch engines

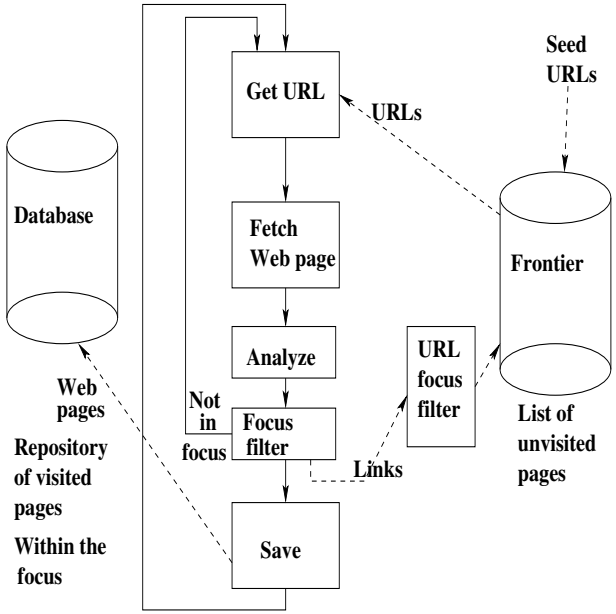
- Simultaneously search several individual search engines
- Query translation
- Result merging
  - Simple merge
  - Duplicate detection
  - tf-idf/similarity ranking
  - Position based

Web Robot - Basic architecture

Spider, Crawler, Robot, agent, ...



Focused Crawling



Focus:

- Domain
- Project
- Country Region
- Topic
- Subject

Browsing vs search

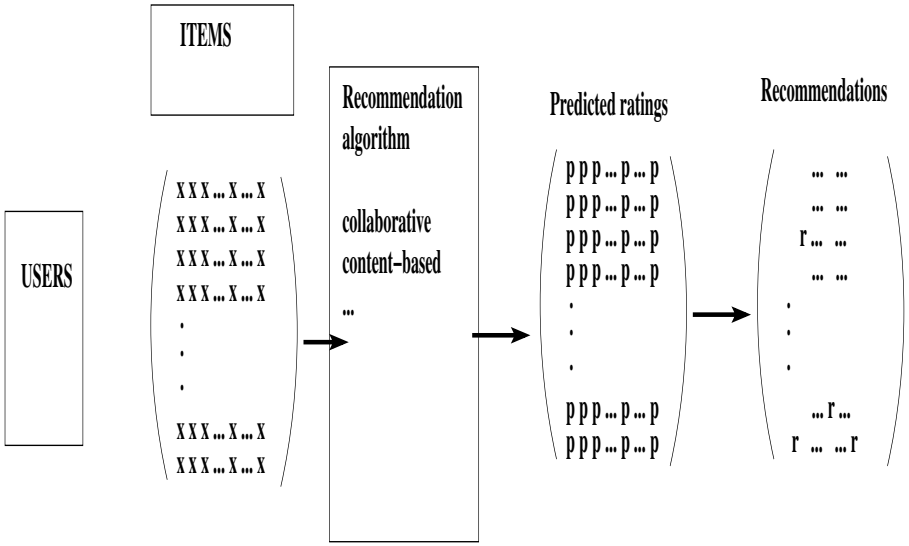
- Search
  - LOTS of data
  - Unstructured
  - Unrelated items clutter results
- Browsing
  - Small amounts of data
  - Hierarchically structured
  - Quality assessed

Outline

Lecture 6

- Recommender systems
- Indexing, searching
- Example IR systems

Recommender systems



Content based filtering

Try to predict a rating based on my own ratings

- Represent items as a set of features
  - $item_j = (w_{1j}, ...w_{kj})$
- Users rank items $\rightarrow$ user profile in feature space
  - $user_c = (w_{c1}, ...w_{ck})$
- Vector space! (feature/item matrix, tf idf, cosine similarity, ...)
- User profile used as query

Collaborative filtering

Try to predict rating based on other users ratings

- Memory based
  - Make rating based on entire collection
  - Ex: $rating_{c,s} = k * \sum_{c' \in C} sim(c, c') * rating_{c',s}$

    User c, Item s
    C Set of users most similar to c
    k Normalizing factor ($usually \frac{1}{\sum_{c' \in C} |sim(c, c')|}$)
- Model based
  - Try to learn a model to be used for predicting ratings
  - Ex: Probabilistic model, Machine learning, ...

Hybrid systems

- Content based filtering + Collaborative filtering
  - Combining separate recommenders
  - Adding content based characteristics to collaborative filtering
  - Adding collaborative characteristics to content based filtering

Introduction: Indexing, searching

- Sequential search
  - Small databases
  - Volatile data
- Indexes
  - Large databases
  - Semi-static data
    - Inverted files

Inverted files

- Principal data structure
  - Effective
  - Allows fast searching
  - Substantial storage overhead
    - Speed more important than storage
- For each term
  - List of document ID's
  - (Term frequency in each document)
  - (Position in document)
- Used for
  - Boolean searches
  - Vector space ranking
  - Proximity, phrases

Example IR-systems

- IndexData: Zebra
- free, GPL license
- index records in XML, SGML, MARC, e-mail archives, ...
- supports SRU/CQL, Z39.50, ZOOM

- Apache: Lucene and Solr
- free, open source
- index records via XML over HTTP
- query via HTTP GET, XML results

Outline

Lecture 7

- Digital Libraries
- Example systems

Digital Library

- body of knowledge
- digital information
- large collection
- builds on current libraries
- can be accessed from anywhere
- extended IR system
  - support large collections
  - searching
  - cataloguing(indexing)

Problems

- multilingual
- multimedia
- structured docs
- distribution
- federated search
- access

interoperability!

Standards

- protocols
- metadata
- classification systems
- query languages