Chapter 9:
Information Retrieval

See corresponding chapter in Manning&Schütze

Goal

- In IR there is a much larger variety of possible metrics
- For different tasks, different metrics might be appropriate

Information Retrieval - Google Search - Microsoft Internet Explorer

Datei Bearbeiten Ansicht Favoriten Extras ?

Zurück Suchen Favoriten Medien

Adresse http://www.google.de/search?hl=en&q=Information+Retrieval&meta= Wechseln zu Links

Google Information Retrieval Los geht's! Lesezeichen 56 blockiert Rechtschreibprüfung Senden an Einstellungen

Sign in

Web [Images](#) [Groups](#) [News](#) [Products](#) [Scholar](#) [more »](#)

Information Retrieval [Search] [Advanced Search](#) [Preferences](#)

Web Results **1** - **10** of about **44,800,000** for **Information Retrieval**. (**0.10** seconds)

Information retrieval - Wikipedia, the free encyclopedia
Information retrieval (IR) is the science of searching for **information** in documents, searching for documents themselves, searching for metadata which **...**
en.wikipedia.org/wiki/**Information_retrieval** - 62k - Cached - Similar pages

Information Retrieval
An online book by CJ van Rijsbergen, University of Glasgow.
www.dcs.gla.ac.uk/Keith/Preface.html - 7k - Cached - Similar pages

Information Retrieval
Online text of a book by Dr. CJ van Rijsbergen of the University of Glasgow covering advanced topics in **information retrieval**.
www.dcs.gla.ac.uk/~iain/keith/ - 5k - Cached - Similar pages

information retrieval
www.springerlink.com/link.asp?id=103814 - Similar pages

Modern **Information Retrieval**
A recent IR book, covering algorithms, implementation, query languages, user interfaces, and multimedia and web **retrieval**.
www.ischool.berkeley.edu/~hearst/irbook/ - 9k - Cached - Similar pages

Information Retrieval Research - SearchTools Topics
An up-to-date overview of research in the field of **information retrieval**.
www.searchtools.com/info/info-**retrieval**.html - 22k - Cached - Similar pages

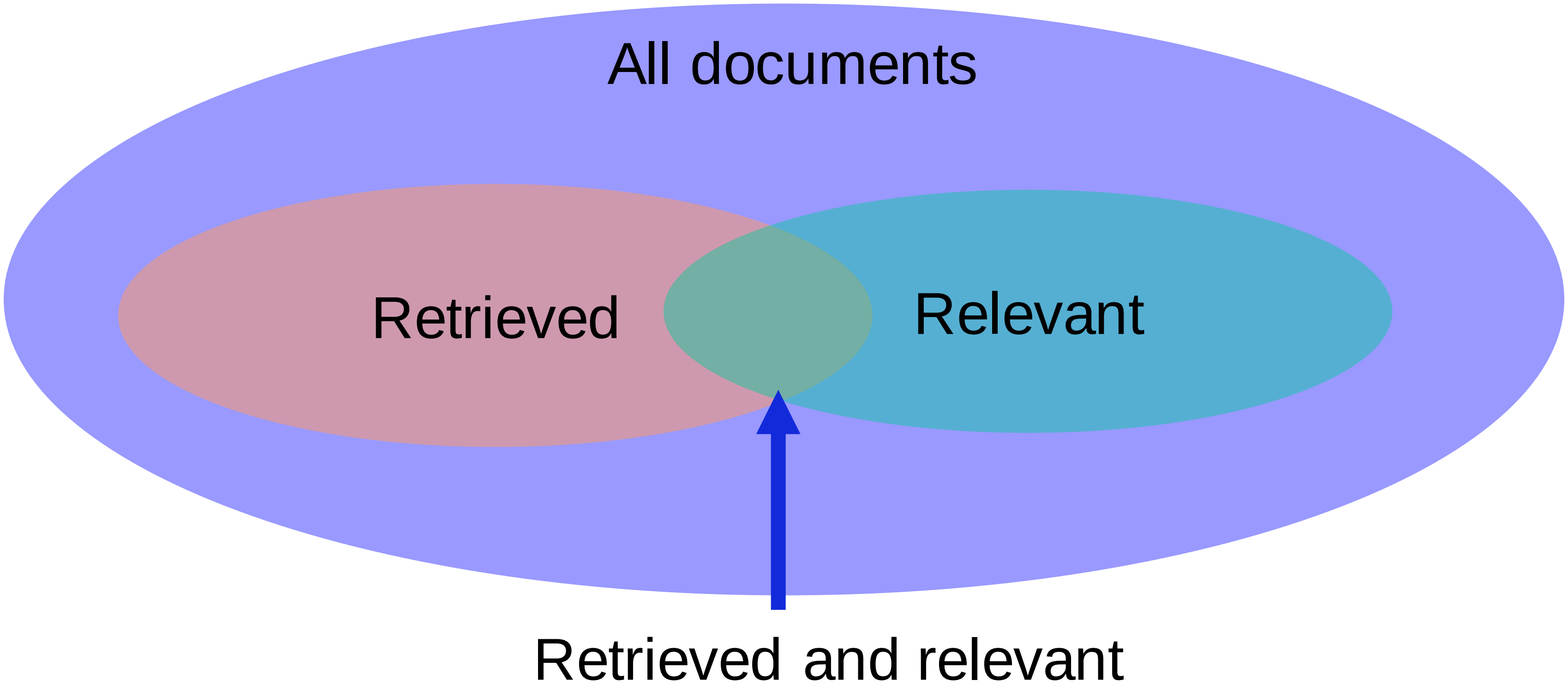
Fertig Internet

Start Dokument1 - Microsof... SNLP_06_Chap8 SNLP_07_Chap1 Information Retrieval... DE 22:25

Evaluation Metrics in IR

Reading: corresponding section in Manning&Schütze

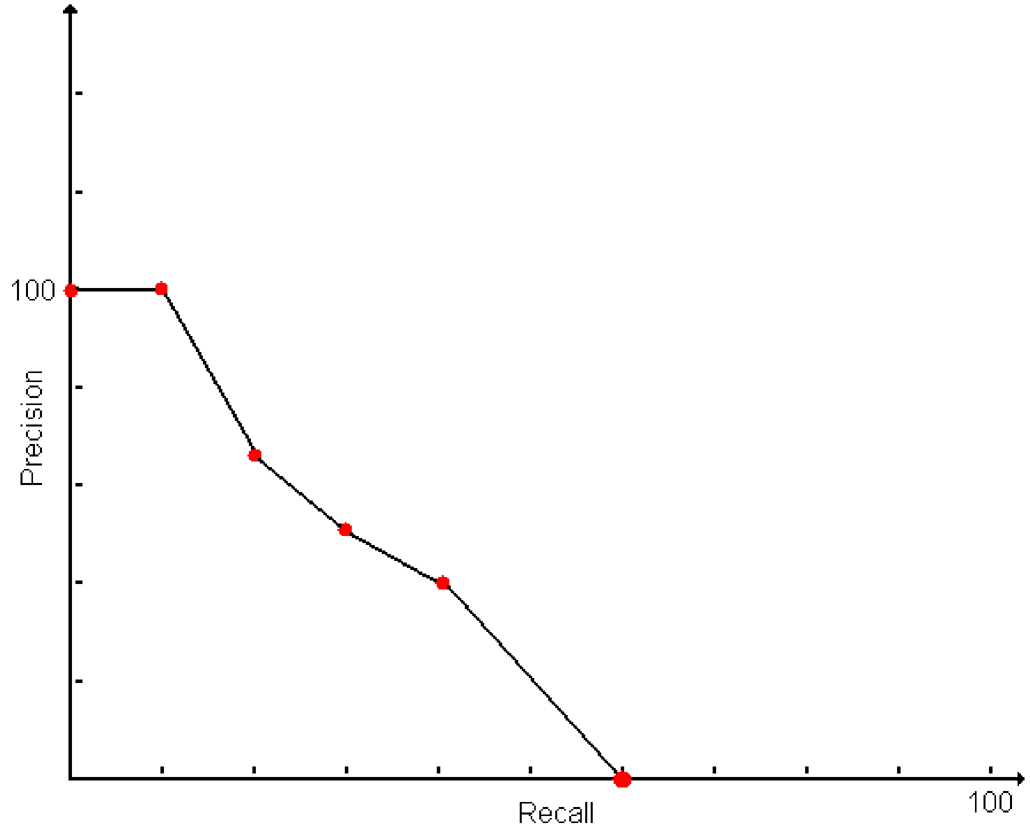
Evaluation of IR Systems



Recall=#(retrieved and relevant)/#(relevant)

Precision=#(retrieved and relevant)/#(retrieved)

Precision vs. Recall



- Standard approach in IR
- Curves can cross!

Average precision

Goal:

- don't focus on a specific recall level
- still get one number
- Each of the numbers below depends on query Q

$$AvgP = \frac{\sum_{r=1}^{N} P(r)\, rel(r)}{\sum_{r=1}^{N} rel(r)}$$

P(r) : precision at rank r

rel(r) : indicator function;

1 if document at rank r is relevant

Mean average precision

- Problem: average precision still specific to query

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q} AvgP(q)$$

Q: number of queries

Other metrics

- Mean reciprocal rank (MRR)

$$MRR = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{rank(q)}$$

$rank(q)$: rank of the first match for query q

- F-Score $\frac{1}{F} = \frac{1}{P} + \frac{1}{R}$

- Precision at r

Vector Space Model and tf-idf

Reading: corresponding section in Manning&Schütze

Goal

- Precursor: boolean search
- Vector space mode: a simple and fast retrieval algorithm that allows for ranking
- Introduced by Karen Spärck Jones in 1972

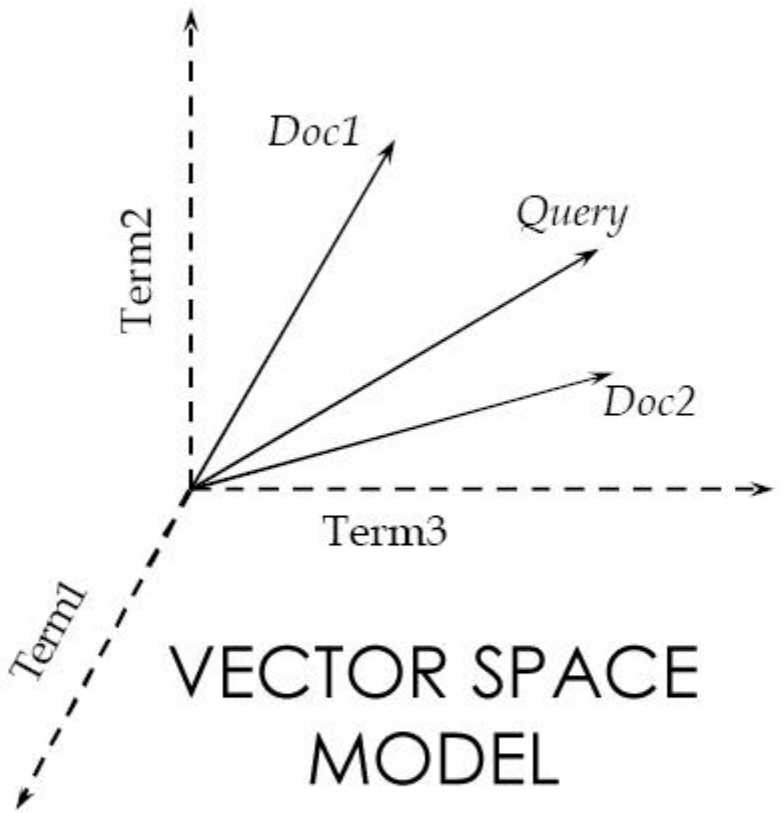
Preprocessing

- Stemming (“going” $\mapsto$ ”go”; “fishes” $\mapsto$ ”fish”, . . .)
- Stop words (remove “and”, “to”, “the”, . . .)
- Longer units (“New York” $\mapsto$ ”New_York”)

- What to do depends to task and retrieval algorithm that is used

Vector-space-model

- Key idea: represent each document and also the query as a vector



Vector-space-model

- Considering every document as vector
  - The vector contains the weights of the index terms as components
  - In case of t index terms the dimension of the vector-space is also t
  - Similarity of querie to a document is the correlation between the their vectors
  - Correlation quantified by cosine of the angle between the vectors

Vector-space-model

- Index term weights

$$tf_{i,j} = \frac{freq_{i,j}}{\max_l freq_{l,j}}$$

with $freq_{i,j}$ the frequency that term i occurs in document j

$$idf_i = \log \frac{N}{n_i}$$

with n_i number of documents (ignoring query) that contain term i

and N total number of documents

Vector-space-model

- Index term weights
  - The weight of a term in a document is then calculated as product of the tf factor and the idf factor

$$w_{i,j} = tf_{i,j} \times idf_i$$

  - Or for the query

$$w_{i,q} = \left(0.5 + \frac{0.5\, freq_{i,q}}{\max_l \; freq_{l,q}} \right) \times idf_i$$

Distance Metrics

- Pick an L-norm
- Angel/cosine between vectors

$$\cos(\vec{q},\vec{d}) = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{i=1}^{n} d_i^2}}$$

Vector-space-model

- Advantages
  - Improves retrieval performance as compared to Boolean retrieval
  - Partial matching allowed
  - Sort according to similarity
- Disadvantages
  - Assumes that index terms are independent

Models of Term Distribution

Reading: corresponding section in Manning&Schütze

Models for Term Distribution

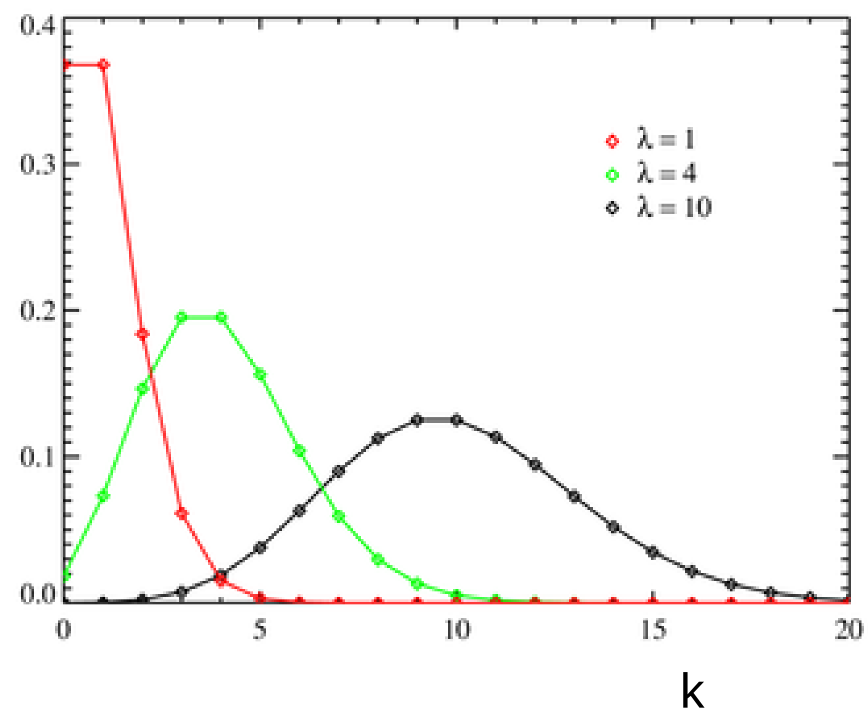
- Goal:
  - Understand the statistical properties of key words in a documents collection
- Assumptions
  - Probability for a term is proportional to the length of the document
  - Short text: each word occurs only once
  - Two neighboring occurrences of the same term are statistically independent

Poission Distribution

Probabiliity that the i-th term occurs k times in the document

$$P_{\lambda_i}(k) = e^{-\lambda_i} \frac{\lambda_i^{\,k}}{k!}$$

λ_i parameter of the distribution



Processes described by Poisson Distribution (Wikipedia)

- The number of cars that pass through a certain point on a road (sufficiently distant from traffic lights) during a given period of time.
- The number of spelling mistakes one makes while typing a single page.
- The number of phone calls at a call center per minute.
- The number of times a web server is accessed per minute.
- The number of roadkill (animals killed) found per unit length of road.
- The number of mutations in a given stretch of DNA after a certain amount of radiation.
- The number of unstable nuclei that decayed within a given period of time in a piece of radioactive substance. The radioactivity of the substance will weaken with time, so the total time interval used in the model should be significantly less than the mean lifetime of the substance.
- The number of pine trees per unit area of mixed forest.
- The number of stars in a given volume of space.
- The number of V2 rocket attacks per area in England, according to the fictionalized account in Thomas Pynchon's Gravity's Rainbow.
- The number of light bulbs that burn out in a certain amount of time.
- The number of viruses that can infect a cell in cell culture.
- The number of hematopoietic stem cells in a sample of unfractionated bone marrow cells.
- The inventivity of an inventor over their career.
- The number of particles that "scatter" off of a target in a nuclear or high energy physics experiment.

Check normalization and expectation value

$\mapsto$ see white board

Relate to tf-idf

Expectation value: how often is the term i expected to occur in a document $\mapsto$

$$N\ E_i(k) = N\,\lambda_i =: cf_i \quad \text{(collection frequency)}$$

Term present at least once $\mapsto$

$$N\,(1 - P_{\lambda_i}(0)) =: df_i \quad \text{(document frequency)}$$

N: the number of documents in the corpus

Experimental Test of Poisson Model

Word	cf_i	λ_i	N(1-P(0))	df_i	Overestimation
follows	23533	0.2968	20363	21744	0.94
transformed	840	0.0106	845	807	1.03
soviet	35337	0.4457	28515	8204	3.48
students	15925	0.2008	14425	4953	2.91
james	11175	0.1409	10421	9191	1.13
freshly	611	0.0077	609	395	1,54

- Model often works
- Some terms like “soviet” are bursty
- independence assumption is not valid

Probabilistic Retrieval

Reading: corresponding section in Manning&Schütze

Goal

- Attempt to justify tf-idf for retrieval

- Probabilistic Retrieval

$\mapsto$ See white board

Language Model based Retrieval

Reading:

- Chapter 12, “Language models for information retrieval” of Manning, Raghavan, and Schutze, Introduction to Information Retrieval, 2009
- Ponte and Croft, “A Language Model, Approach to IR”, SIGIR, 1998

Goal

- Practical way of using the probabilistic ideas for retrieval

Language Modeling

- The probability that a query q was generated by a probabilistic model based on a document.

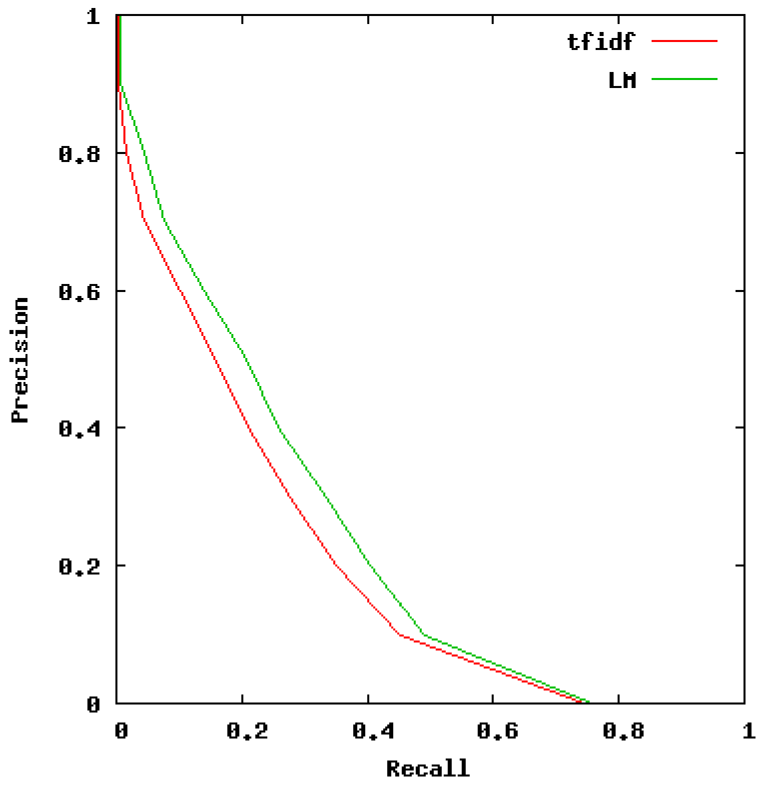
$$q = q_1 q_2 \ldots. q_n \qquad d = d_1 d_2 \ldots d_m$$

$$P(d \mid q) \approx P(q \mid d) P(d)$$

- Unigram model and ignoring prior P(d):

$$P(q \mid d) = \prod_{i=1}^{n} P(q_i \mid d)$$

Performance tf-idf vs LM (original Ponte&Croft-paper)



Results on
TREC 10 collection

$\mapsto$ LM outperforms tf-idf

Smoothing Methods

- Jelinek-mercer method:

a linear interpolation

$$P_{\lambda}(w \mid d) = (1-\lambda)P_{ml}(w \mid d) + \lambda P(w \mid C)$$

of the ML model

$$P_{ml}(w \mid d))$$

with the collection model (trained on all documents)

$$P(w \mid C)$$

Smoothing Methods

- Absolute discounting:

decrease the probability of seen words byc substracting a constant from their counts

$$P_s(w\,|\,d) = \frac{\max(c(w;d) - \delta, 0)}{\sum_{w^* \in V} c(w^*;d)} + \sigma P(w\,|\,C)$$

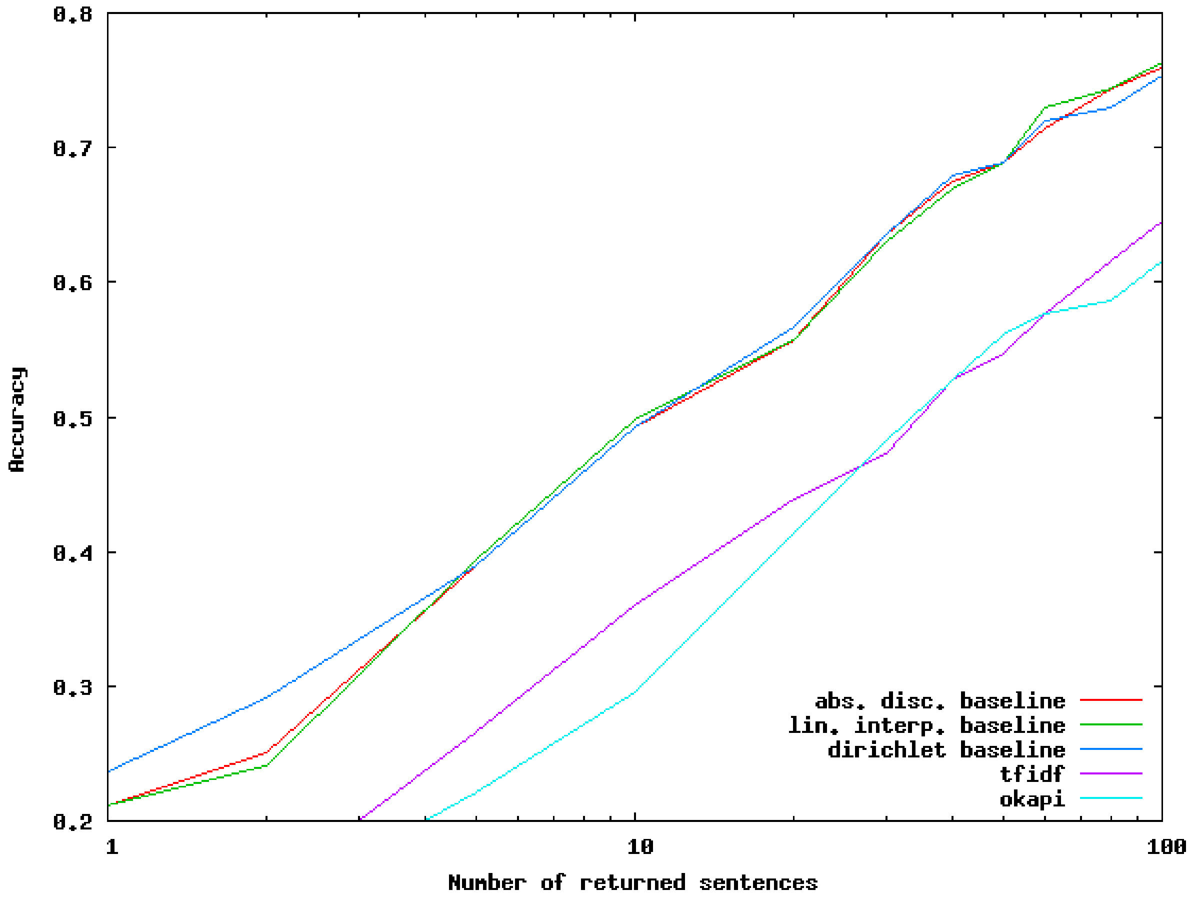
- See chapter 5 (notation differs in IR!)

Smoothing Methods

- **Bayesian smoothing using Dirichlet priors:** A multinomial distribution, for which the conjugate prior for bayesian analysis is the dirichlet distribution:

$$P_{\mu}(w|d) = \frac{c(w;d) + \mu P(w|C)}{\sum_{w^* \in V} c(w^*;d) + \mu}$$

Comparing different smoothing methods: sentence retrieval in question answering



Improved Language Models for IR

- Bigrams
- Class LMs
- Grammar
- Prior knowledge (document length)
- Other resources (e.g. WordNet)

Latent Semantic Analysis (LSA)

Reading: corresponding section in Manning&Schütze

Goal

Overcome semantic mismatch between terms in the query and the documents

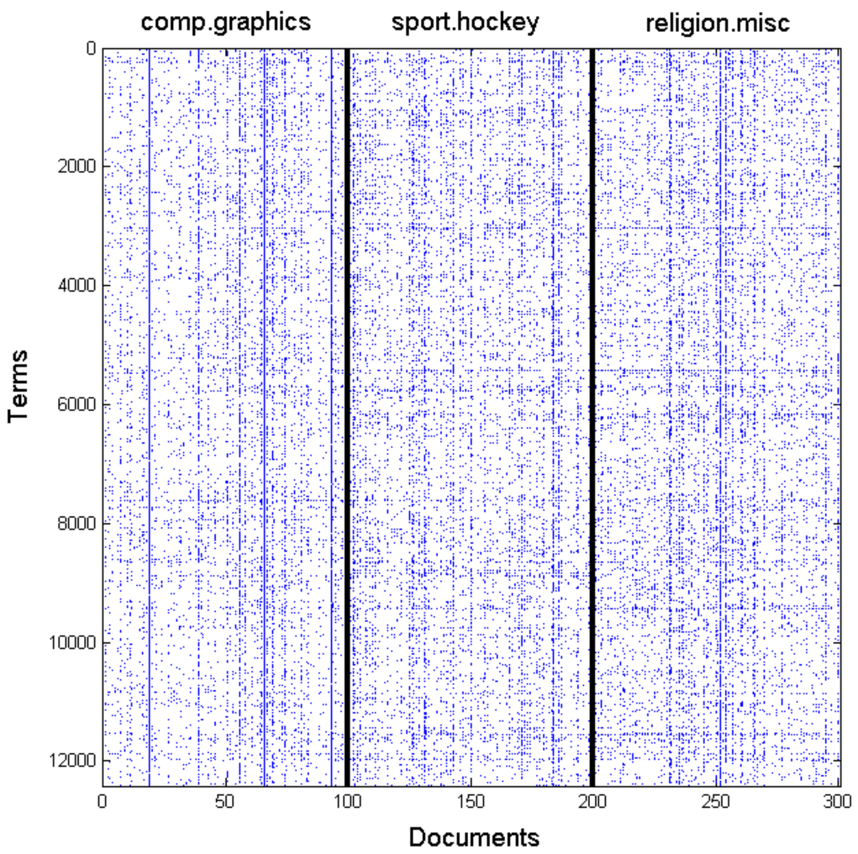
(e.g. cosmonaut vs. astronaut)

Term Document Matrix Structure

- 100 documents from 3 distinct newsgroups
- Indexed using standard stop word list
- 12418 distinct terms
- Term × Document Matrix (12418 × 300)

Term Document Matrix Structure

Idea: derive semantic relatedness from co-occurance in term document matrix



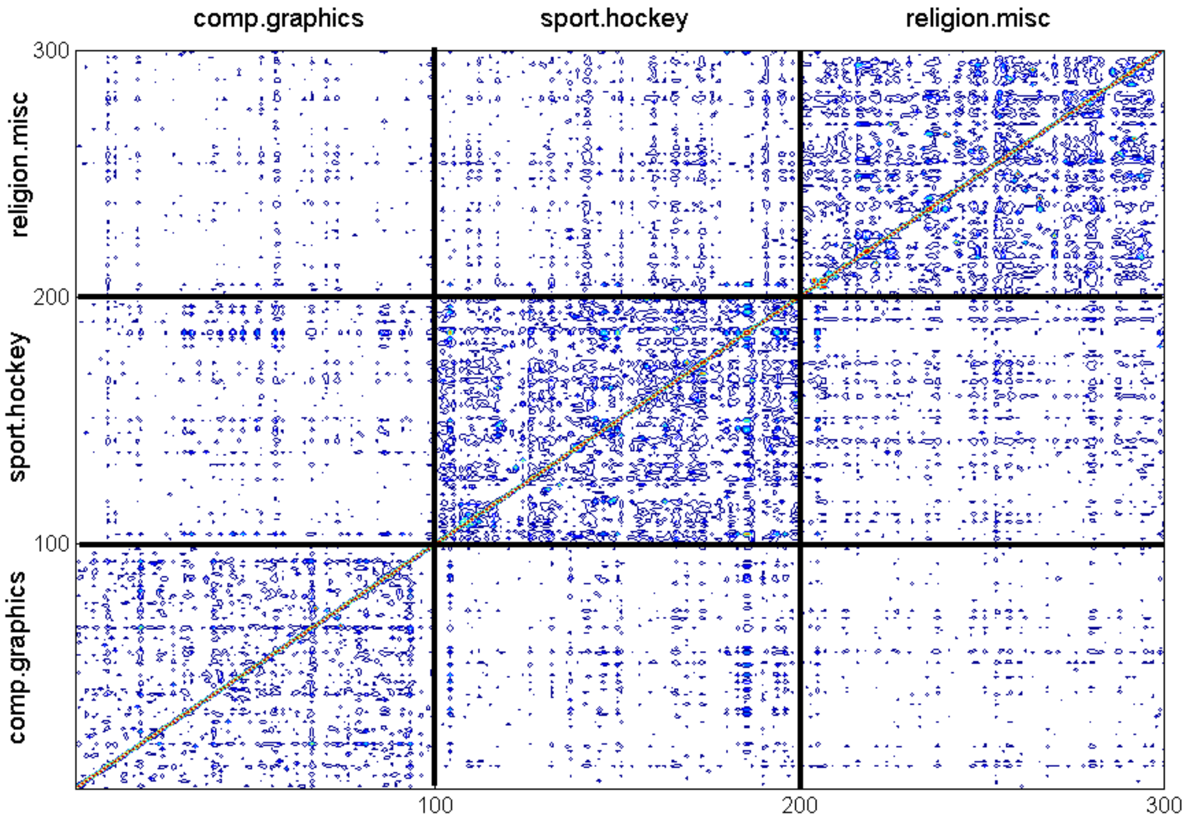
Theory of LSA

- Whiteboard

Latent Semantic Analysis

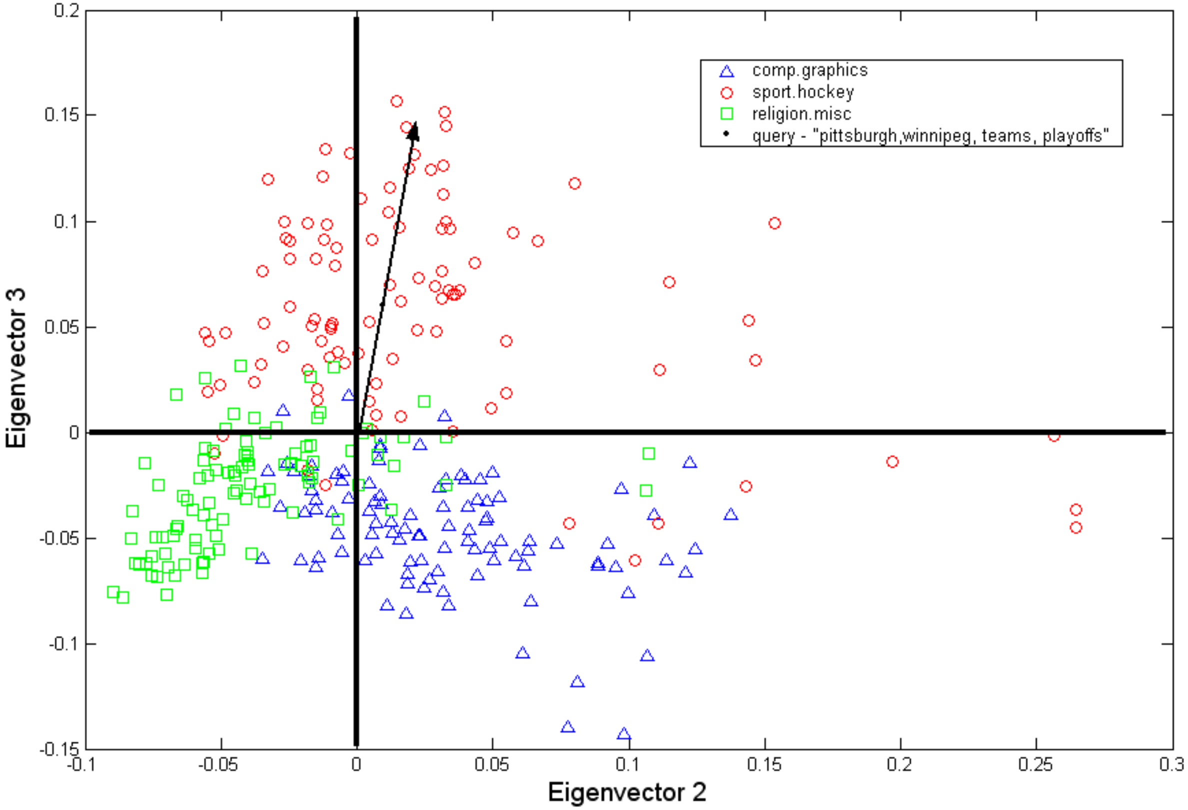
- Word usage defined by term and document co-occurrence – matrix structure
- Latent structure / semantics in word usage
- Clustering documents or words Singular Value Decomposition
- Cubic Computational Scaling

Term Document Matrix Structure



↦ clearly visible similarity within a topic

Retrieval with LSA: map query to semantic space



$\mapsto$ Plausible mapping of query

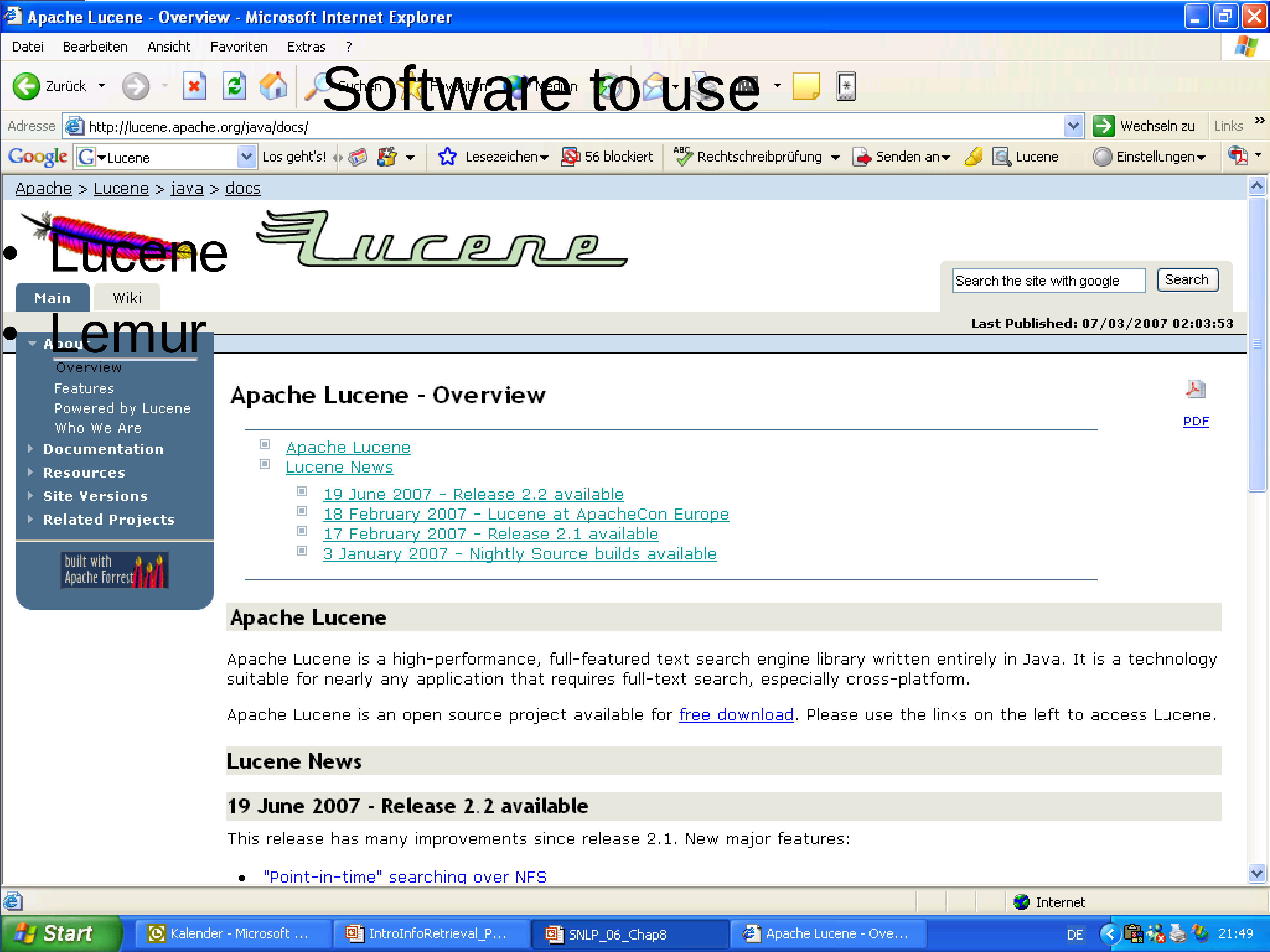
LSA Performance

- LSA consistently improves recall on standard test collections (precision/recall generally improved)
- Variable performance on larger TREC collections
- Dimensionality of Latent Space – a magic number – 300 – 1000 seems to work fine
- Computational cost high (~cubic)

Toolkits

Software to use

- Lucene
- Lemur



The Lemur Toolkit for Language Modeling and Information Retrieval - Microsoft Internet Explorer

Datei Bearbeiten Ansicht Favoriten Extras ?

Zurück Suchen Favoriten Medien

Adresse http://www.lemurproject.org/ Wechseln zu Links

Google Lemur Toolkit Los geht's! Lesezeichen 56 blockiert Rechtschreibprüfung Senden an Lemur Einstellungen

[printable version]

The Lemur Toolkit

for Language Modeling and Information Retrieval

News
Features
The Lemur Toolkit
Indri Search Engine
Download
People
Discussion
Tutorials
Sign Up

The Lemur Toolkit is a open-source toolkit designed to facilitate research in language modeling and information retrieval. Lemur supports a wide range of industrial and research language applications such as ad-hoc retrieval, site-search, and text mining.

The toolkit supports indexing of large-scale text databases, the construction of simple language models for documents, queries, or subcollections, and the implementation of retrieval systems based on language models as well as a variety of other retrieval models. The system is written in the C and C++ languages, and is designed as a research system to run under Unix operating systems, although it can also run under Windows.

The toolkit is in constant development as part of the Lemur Project, a collaboration between the Computer Science Department at the University of Massachusetts and the School of Computer Science at Carnegie Mellon University.

News	News and announcements about the Lemur Toolkit, such as the latest release notes, upcoming releases and known problems with current versions.
Features	An "at-a-glance" listing of features within the Lemur Toolkit.
The Lemur Toolkit	How to install and use the Lemur Toolkit, together with code-level documentation, applications guides, working with offset annotations and beginners guides.
Indri Search Engine	More about Indri, Lemur's latest search engine that is also available on its own when all you need is a search engine. Indri has an index capable of indexing very large collections and a structured query

Carnegie Mellon University

Fertig Internet

Start Kalender - Microsoft ... IntroInfoRetrieval_P... SNLP_06_Chap8 The Lemur Toolkit for ... DE 21:50

Summary

- Evaluation measures
- Vector space model
- Models of term distribution
- Probabilistic retrieval
- Latent semantic analysis
- Language models for IR