VK Multimedia Information Systems

Mathias Lux, mlux@itec.uni-klu.ac.at

Dienstags, 16.oo Uhr c.t., E.2.69

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 2.0 License. See http://creativecommons.org/licenses/by-nc-sa/2.0/at/

Information Retrieval Basics: Agenda

- **Information Retrieval History**
- Information Retrieval & Data Retrieval
- Searching & Browsing
- Information Retrieval Models
- Exercise 01

Information Retrieval History

Currently there are no museums for IR

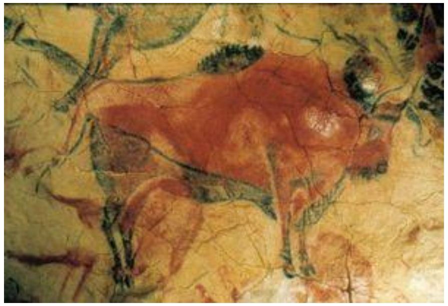
IR is the process of **searching** through a **document collection** based on a particular **information need**.

IR Key Concepts

- Searching
  - Indexing, Ranking
- Document Collection
  - Textual, Visual, Auditive
- Particular Needs
  - Query, User based

A History of Libraries

Libraries are perfect examples for document collections.



- Wall paintings in caves
  - e.g. Altamira, ~ 18,500 years old
- Writing in clay, stone, bones
  - e.g. Mesopotamian cuneiforms, ~ 4.000 BC
  - e.g. Chinese tortoise-shell carvings, ~ 6.000 BC
  - e.g. Hieroglyphic inscriptions, Narmer Palette ~ 3.200 BC

A History of Libraries (ctd.)

- Papyrus
  - Specific plant (subtropical)
  - Organized in rolls, e.g. in Alexandria
- Parchment
  - Independence from papyrus
  - Sewed together in books
- Paper
  - Invented in China (bones and bamboo too heavy, silk too expensive)
  - Invention spread -> in 1120 first paper mill in Europe

A History of Libraries (ctd.)

- Gutenberg’s printing press
  - Inexpensive reproduction
  - e.g. “Gutenberg Bible”
- Organization & Storage
  - Dewey Decimal System (DDC, 1872)
  - Card Catalog (early 1900s)
  - Microfilm (1930s)
  - MARC (Machine Readable Cataloging, 1960s)
  - Digital computers (1940s+)

Library & Archives today

- Partially converted to electronic catalogues
  - From a certain timepoint on (1992 - ...)
  - Often based on proprietary systems
  - Digitization happens slow
  - No full text search available
  - Problems with preservation
    - Storage devices & formats

History of Searching

- Browsing
  - Like “Finding information yourself”
- Catalogs
  - Organized in taxonomies, keywords, etc.
- Content Based Searching
  - `SELECT * FROM books WHERE title='%Search%'`
- Information Retrieval
  - Ranking, models, weighting
  - Link analysis, LSA, ...

History of IR

- Starts with development of computers.
- Term “Information Retrieval” coined by Mooers in 1952

- Two main periods (Spark Jones u. Willett)
  - 1955 – 1975: Academic research
    - Models and Basics
    - Main Topics: Search & Indexing
  - 1975 – ... : Commercial applications
    - Improvement of basic methods

A Challenge: The World Wide Web

- First actual implementation of **Hypertext**
  - Interconnected documents
  - Linked and referenced
- World Wide Web (1989, T. Berners-Lee)
  - Unidirectional links (target is not aware)
  - Links are not typed
  - Simple document format & communication protocol (HTML & HTTP)
  - Distributed and not controlled

Some IR History Milestones

- Book “Automatic Information Organization and Retrieval”, *Gerard Salton* (1968)
  - Vector Space Model
- Paper “A statistical interpretation of term specificity and its application in retrieval”, Karen Spärck Jones (1972)
  - IDF weighting
  - http://www.soi.city.ac.uk/~ser/idf.html
- Book “Information Retrieval” of *C.J. Rijsbergen* (1975)
  - Probabilistic Model
  - http://www.dcs.gla.ac.uk/Keith/Preface.html

Some IR History Milestones

- Paper “Indexing by Latent Semantic Analysis”, S. Deerwester, Susan Dumais, G. W. Furnas, T. K. Landauer, R. Harshman (1990).
  - Latent Semantic Indexing
- Paper “Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval” Robertsen & Walker (1994)
  - BM25 weighting scheme
- Paper “The Anatomy of a Large-Scale Hypertextual Web Search Engine”, *Sergey Brin & Larry Page* (1998)
  - World Wide Web Retrieval

Information Retrieval Basics: Agenda

- Information Retrieval History
- **Information Retrieval & Data Retrieval**
- Searching & Browsing
- Information Retrieval Models
- Exercise 01

Organizational: References

- in the Library
  - *Modern Information Retrieval*, Ricardo Baeza-Yates & Berthier Ribeiro-Neto, Addison Wesley
  - *Google's Pagerank and Beyond: The Science of Search Engine Rankings*, Amy N. Langville & Carl D. Meyer, University Presses of CA
  - *Distributed Multimedia Database Technologies supported by MPEG-7 and MPEG-21*, Harald Kosch, CRC Press
  - *Readings in Information Retrieval*, Karen Sparck Jones, Peter Willett, Morgan Kaufmann

Organizational: References

- WWW
  - *Skriptum Information Retrieval*, Norbert Fuhr, Lecture Notes on Information Retrieval - Univ. Dortmund, 1996. Updated in 2002
  - *Information Retrieval 2nd Edt.*, C.J. Rijsbergen, Butterworth, London 1979
- Through me:
  - *Lectures on Information Retrieval: Third European Summer-School, Essir 2000 Varenna, Italy, Revised Lectures*, Maristella Agosti, Fabio Crestani & Gabriela Pasi (eds.), Lecture Notes in Computer Science, Springer 2000

Information Retrieval & Data Retrieval

Information Retrieval

- Information Level
- Search Engine
- Teoma / Google

Data Retrieval

- Data Level
- Data Base
- Oracle / MySQL

Information Retrieval & Data Retrieval

Information Retrieval	**Data Retrieval**
Content Based Search	Search for Patterns and String
Query ambigous	Query formal & unambigous
Results ranked by relevance	Results not ranked
Error tolerant	Not error tolerant
Multiple iterations	Clearly defined result set
Examples	*Examples*
Search for synonyms	Search for patterns
Bag of Words	SQL Statement

- Retrieval is nearly always a combination of both.

IR Dimensionen

Matching	Exact Match	Partial (best) Match
Inference	Deduction	Induction
Model	Deterministic	Probabilistic
Classification	Monothetic	Polythetic
Query language	Artificial	Natural
Query definition	Complete	Incomplete
Query dependence	Yes	No
Items wanted	Matching	Relevant
Error response	Sensitive	Insensitive
Logic	Classical	Non-classical
Representation	A priori	A posteriori
Language Models	Logical	Statistical

Information Retrieval Basics: Agenda

- Information Retrieval History
- Information Retrieval & Data Retrieval
- **Searching & Browsing**
- Information Retrieval Models
- Exercise 01

Information Retrieval Basics: Searching

A **user** has an **information need**, which needs to be **satisfied**.

- Two different approaches:
  - Browsing
  - Searching

Searching & Browsing

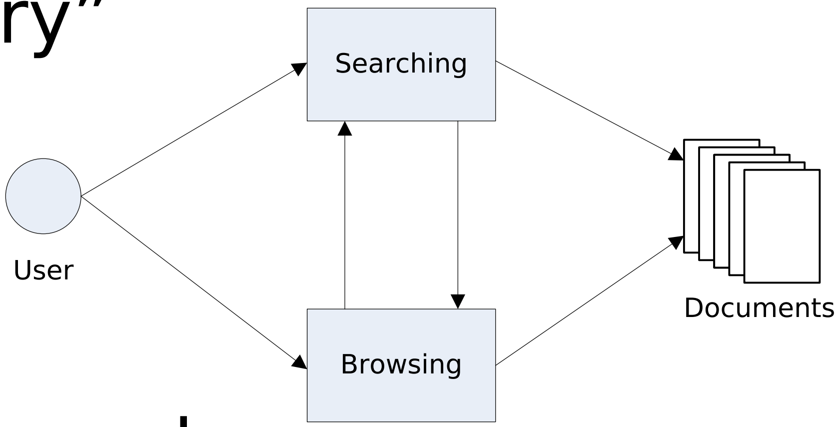
http://www.uni-klu.ac.at

Searching

- Explicit information need
- Definition through “query”
- Result lists
- e.g. Google

Browsing

- Not necessarily explicit need
- Navigation through repositories



Browsing

- Flat Browsing
  - User navigates through set of documents
  - No implied ordering, explicit ordering possible
  - Examples: One single directory, one single file
- Structure Guided Browsing
  - An explicit structure is available for navigation
  - Mostly hierarchical (file directories)
  - Can be generic digraph (WWW)
  - Examples: File systems, World Wide Web

Searching

- Query defines “Information Need”
- Ad Hoc Searching
  - Search when you need it
  - Query is created to fit the need
- Information Filtering
  - Make sets of documents smaller
  - Query is filter criterion
- Information Push
  - Same as filtering, delivery is different

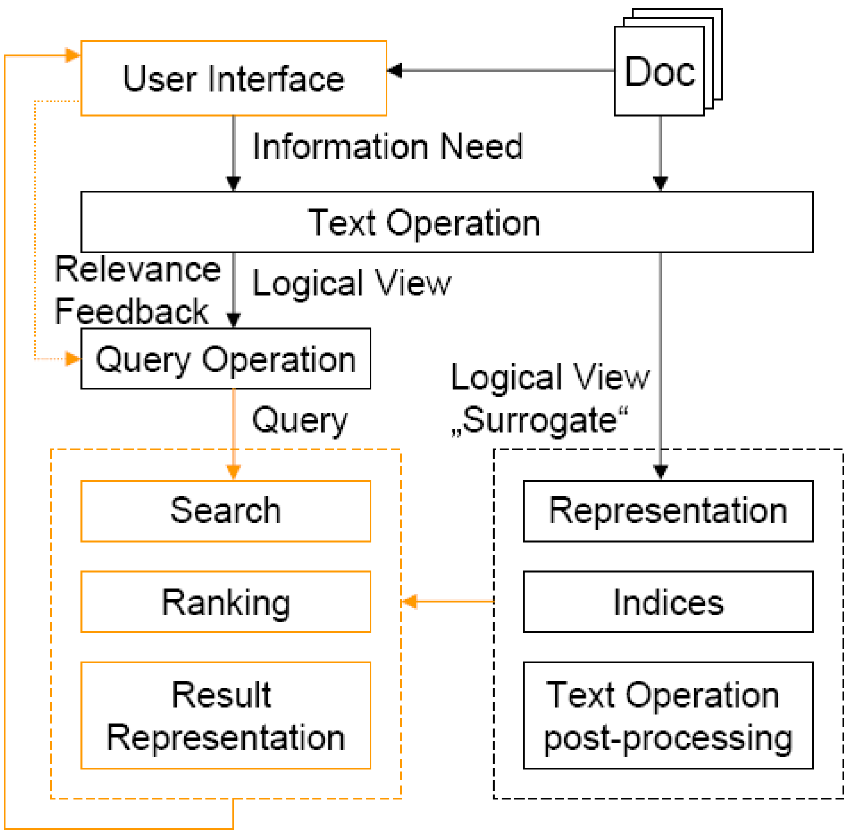
Information Retrieval Basics: Agenda

- Information Retrieval History
- Information Retrieval & Data Retrieval
- Searching & Browsing
- **Information Retrieval Models**
- Exercise 01

Information Retrieval System Architecture

Aspects

- Query & Languages
- IR Models
- Documents
- Internal Representation
- Pre- and Post-processing
- Relevance Feedback
- HCI



Information Retrieval Models

- Boolean Model
  - Set theory & Boolean algebra
- Vector Model
  - Non binary weights on dimensions
  - Partial match
- Probabilistic Model
  - Modeling IR in a probabilistic framework

Formal Definition of Models

An information retrieval model is a quadruple [D, Q, F, $R(q_i, d_j)$]

- D is a set of logical views (or representations) for the **documents** in the collection.
- Q is a set of logical views (or representations) for the user needs or **queries**.
- F is a **framework** for modeling document representations, queries and their relationship.
- $R(q_i, d_j)$ is a **ranking function** which associates a real number with a query q_i of Q and a document d_j of D.

Definitions

in Context of Text Retrieval

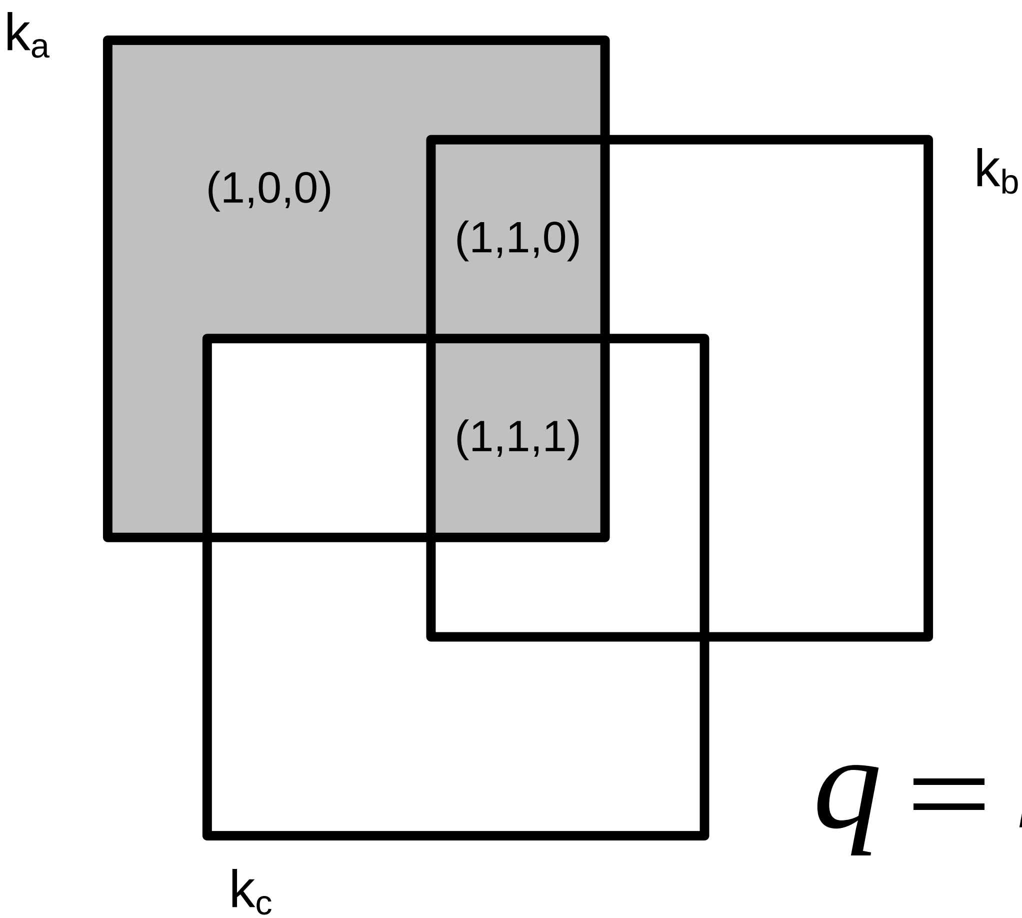
- **index term** – word of a document expressing (part of) document semantics
- **weight** $\boldsymbol{w_{i,j}}$ – quantifies the importance of index term t_i for document d_j
- index term vector for document d_j (having t different terms in all documents):

$$\overrightarrow{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$

Boolean Model

- Based on set theory and Boolean algebra
  - Set of index terms
  - Query is Boolean expression
- Intuitive concept:
  - Wide usage in bibliographic system
  - Easy implementation and simple formalisms
- Drawbacks:
  - Binary decision components (true/false)
  - No relevance scale (relevant or not)

Boolean Model: Example



$$q = k_a \wedge (k_b \vee \neg k_c)$$

Boolean Model: DNF

$$q = k_a \wedge (k_b \vee \neg k_c) \quad \ldots \quad \vec{q}_{dnf} = (1,1,1) \vee (1,1,0) \vee (1,0,0)$$

- Express queries in *disjunctive normal form* (disjunction of conjunctive components)
- Each of the components is a binary weighted vector associated with (k_a, k_b, k_c)
- Weights $w_{i,j} \in \{0,1\}$

Boolean Model: Ranking function

$$sim(d_j,q)=\begin{cases}1 & \text{if } \exists \vec{q}_{cc} \,\middle|\, (\vec{q}_{cc} \in \vec{q}_{dnf}) \wedge (\forall_{k_i}, g_i(\vec{d}_j) = g_i(\vec{q}_{cc})) \\ 0 & \text{otherwise}\end{cases}$$

- similarity is one if one of the conjunctive components in the query is exactly the same as the document term vector.

Boolean Model

- Advantages
  - Clean formalisms
  - Simplicity
- Disadvantages
  - Might lead to too few / many results
  - No notion of **partial match**
  - Sequential ordering of terms not taken into account.

Vector Model

- Integrates the notion of partial match
- Non-binary weights (terms & queries)
- Degree of similarity computed

$$\vec{d}_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$

$$\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$$

Vector model: Similarity

$$sim(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{\left|\vec{d}_j\right| \times \left|\vec{q}\right|} =$$

$$= \frac{\sum_{i=1}^{t} w_{i,j} \cdot w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \cdot \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$$

Vector model: Similarity

$$sim(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{\left|\vec{d}_j\right| \times \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} w_{i,j} \cdot w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \cdot \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$$

Vector Model: Example

http://www.uni-klu.ac.at

$$\vec{d} = (0.3,\ 0.1,\ 0,\ 0.1,\ 1)$$

$$\vec{q} = (1,\ 0,\ 0,\ 0.5,\ 0)$$

$$sim(\vec{d},\vec{q}) = \frac{1 \cdot 0.3 + 0.1 \cdot 0.5}{\sqrt{0.3^2+0.1^2+0.1^2+1^2} \cdot \sqrt{1+0.5^2}} \approx \frac{0.35}{2.24} \approx 0.17$$

Another Example:

- Document & Query:
  - D = "The quick brown fox jumps over the lazy dog"
  - Q = "brown lazy fox"

$$sim(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{\left|\vec{d}_j\right| \times \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} w_{i,j} \cdot w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \cdot \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$$

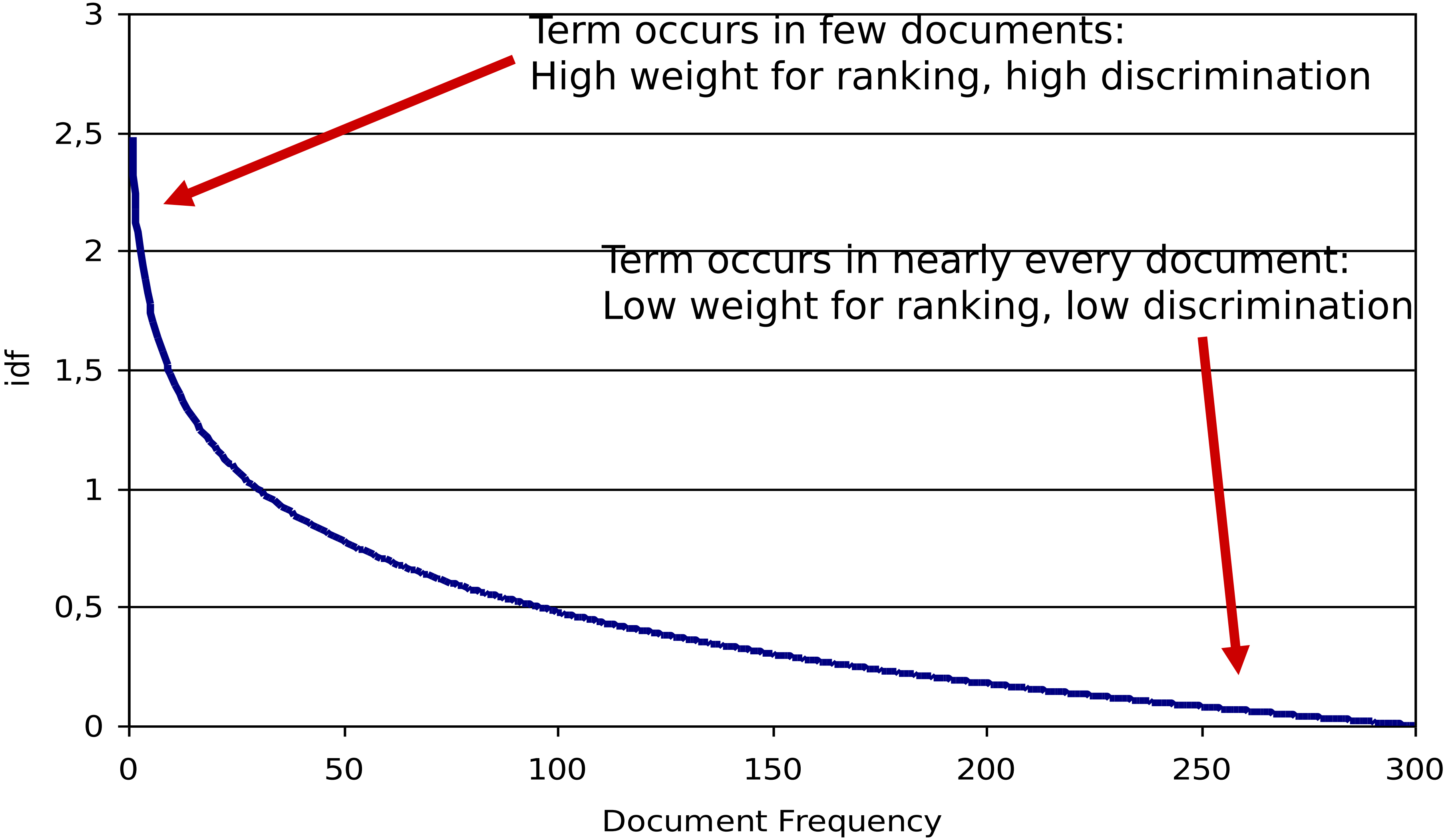
- Results:
  - $(1,1,1,1,1,1,1,1,1)^t * (1,1,1,0,0,0,0,0,0)^t = 3$
  - sqrt(9) + sqrt(3) = 4,732
  - Similarity = 3 / 4,732 = 0,634

Term weighting: TF*IDF

Term weighting increases retrieval performance ...

- Term frequency
  - Most intuitive approach
  - How often does a term occur in a document?
- Inverse Document Frequency
  - How much information content has a term for a document collection?
  - Compare to Information Theory of Shannon

Example: IDF
300 documents corpus



Definitions: Normalized Term Frequency

$$f_{i,j} = \frac{freq_{i,j}}{\max_l(freq_{l,j})}$$ ... normalized term frequency

$freq_{i,j}$ ... raw term frequency of term i in document j

- Maximum is computed over all terms in a document
- Terms which are not present in a document have a raw frequency of *0*

Definitions: Inverse Document Frequency

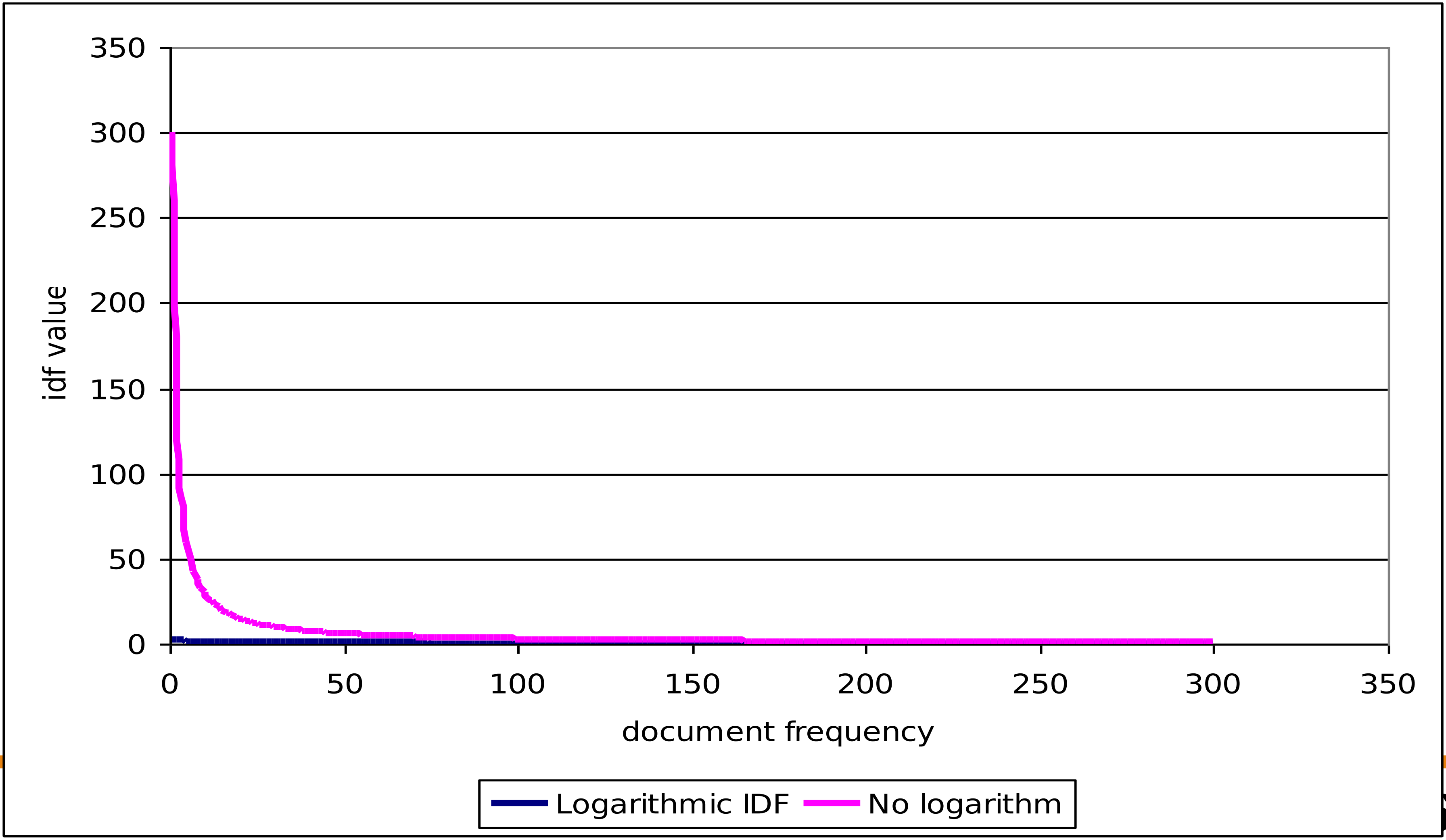
$idf_i = \log \frac{N}{n_i}$ ... inverse document frequency for term i

N ... number of documents in the corpus

n_i ... number of document in the corpus which contain term i

- Note that idf_i is independent from the document.
- Note that the whole corpus has to be taken into account.

Why log(...) in IDF?



TF*IDF

- TF*IDF is a very prominent weighting scheme
  - Works fine, much better than TF or Boolean
  - Quite easy to implement

$$w_{i,j} = f_{i,j} \cdot \log \frac{N}{n_i}$$

Weighting of query terms

$$w_{i,q} = (0.5 + \frac{0.5 \cdot f_{i,q}}{\max_l (f_{l,q})}) \cdot \log \frac{N}{n_i}$$

- Also using IDF of the corpus
- But TF is normalized differently
  - TF > 0.5
- Note: the query is not part of the corpus!

Vector Model

- Advantages
  - Weighting schemes improve **retrieval performance**
  - Partial matching allows retrieving documents that **approximate query** conditions
  - Cosine coefficient allows **ranked list** output
- Disadvantages
  - Term are assumed to be mutually independent

Simple example (i)

- Scenario
  - Given a **document corpus on birds**: nearly each document (say 99%) contains the word bird
  - someone is searching for a document about sparrow nest construction with a query **“sparrow bird nest construction”**
  - Exactly the document which would satisfy the user needs **does not have the word “bird”** in it.

Simple example (ii)

- TF*IDF weighting
  - knows upon the low discrimative power of the term bird
  - The weight of this term is near to zero
  - This term has virtually no influence on the result list.

Probabilistic Model

- Introduced 1976
  - Robertson & Sparck Jones
  - Binary independence retrieval (BIR) model
  - Based on a probabilistic framework

- Basic idea:
  - Given a user query there is a set of documents, that contains only the relevant ones
  - This set is called the **ideal answer set**

Probabilistic Model: Basic Idea

- Querying = specification of the ideal answer set.
  - We do not know the specification
  - We just have some terms to reflect it
- Initial guess for the specification:
  - Allows to generate a preliminary probabilistic description of the ideal answer set.
- User interaction then enhances the probabilistic description.

Probabilistic Model

- For Query *q* und Document d_j:
  - Probabilistic Model tries to determine the **probability of relevance**
- Assumptions
  - The probability of relevance depends on *q* and *D* only
  - The ideal answer set is labeled *R*
  - *R* maximizes the probability of relevance
  - Rank: P(d_j relevant for *q*)/P(d_j not relevant for *q*)
- Note:
  - No way to compute the probability is given
  - No sample space for the computation is given.

Probabilistic Model: Definition

Definition Probabilistic Model:

- All weights are binary:
  - $w_{i,j} \in \{0,1\}$, $w_{i,q} \in \{0,1\}$
- *q* part of the set of index terms k_i
- Ideal Answer Set is *R*, not relevant documents: $\bar{R}$
- Probability that d_j is relevant for q:

$$P(R \mid \vec{d}_j)$$

- Probability that d_j is not relevant for q:

$$P(\bar{R} \mid \vec{d}_j)$$

Probabilistic Model: Definition

- Similarity q and d_j:

$$sim(d_j, q) = \frac{P(R \mid \vec{d}_j)}{P(\bar{R} \mid \vec{d}_j)}$$

- Using Bayes' Rule:

$$sim(d_j, q) = \frac{P(R \mid \vec{d}_j)}{P(\bar{R} \mid \vec{d}_j)} = \frac{P(\vec{d}_j \mid R) \times P(R)}{P(\vec{d}_j \mid \bar{R}) \times P(\bar{R})}$$

- Probability for randomly selecting d_j out of R

$$P(\vec{d}_j \mid R)$$

- Probability for a randomly selected document to be in R

$$P(R)$$

Probabilistic Model: Definition

- As $P(R) = P(\overline{R})$ $\quad sim(d_j, q) \approx \frac{P(\vec{d}_j \mid R)}{P(\vec{d}_j \mid \overline{R})}$
- Assuming independent index terms:

$$sim(d_j, q) \approx \frac{(\prod_{g_i(\vec{d}_j)=1} P(k_i \mid R)) \times (\prod_{g_i(\vec{d}_j)=0} P(\overline{k}_i \mid R))}{(\prod_{g_i(\vec{d}_j)=1} P(k_i \mid \overline{R})) \times (\prod_{g_i(\vec{d}_j)=0} P(\overline{k}_i \mid \overline{R}))}$$

- $P(k_i \mid R)$ .... Probability that k_i is in a randomly selected document from R
- $P(\overline{k}_i \mid R)$ .... Probability that k_i is not in a randomly selected document from R
- the same for $P(k_i \mid \overline{R})$, $P(\overline{k}_i \mid \overline{R})$

Probabilistic Model: Definition

Simplification based on

- $P(k_i \mid R) + P(\bar{k}_i \mid R) = 1$
- Using logarithms
- And ignoring factors constant for all documents:

$$sim(dj, q) \approx \sum_{i=1}^{t} w_{i,q} \times w_{i,j} \times \left(\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{1 - P(k_i \mid \bar{R})}{P(k_i \mid \bar{R})} \right)$$

- Problems
  - *R* is not know at query time
  - Therefore we cannot calculate $P(k_i \mid R)$ and $P(k_i \mid \bar{R})$

Probabilistic Model: Starting Probabilities (i)

- Assumptions:
  - $P(k_i|R)$ is constant for all k_i (e.g. *0.5*)
  - Distribution of index terms k_i in ^R is ~ distribution of index terms k_i in D

$$P(k_i \mid R) = 0{,}5 \qquad P(k_i \mid \overline{R}) = \frac{n_i}{N}$$

  - n_i ... number of document containing k_i
  - $N = |D|$

Probabilistic Model: Starting Probabilities (ii)

- Based on these assumptions a ranked list is generated
- Iterative enhancement
  - Automatically, without user interaction
  - *V* is set of top ranked documents (up to *r* docs)
  - V_i is subset of *V* containing k_i
  - These variables also denote the set cardinality.

$$P(k_i \mid R) = \frac{V_i}{V} \qquad P(k_i \mid \overline{R}) = \frac{n_i - V_i}{N - V}$$

Probabilistic Model: Starting Probabilities (iii)

- Problems with small numbers, e.g.
  - *V* is *1*, V_i is *0*
  - e.g. with constant adjustment factor

$$P(k_i \mid R) = \frac{V_i + 0{,}5}{V + 1} \qquad P(k_i \mid \overline{R}) = \frac{n_i - V_i + 0{,}5}{N - V + 1}$$

  - or not constant:

$$P(k_i \mid R) = \frac{V_i + \frac{n_i}{N}}{V + 1} \qquad P(k_i \mid \overline{R}) = \frac{n_i - V_i + \frac{n_i}{N}}{N - V + 1}$$

Probabilistic Model

- Advantages:
  - Relevance is decreasing order of probability
  - Therefore partial match is supported
- Disadvantages
  - Initial guessing of R
  - Binary weights
  - Independence assumption of index terms

Information Retrieval Basics: Agenda

- Information Retrieval History
- Information Retrieval & Data Retrieval
- Searching & Browsing
- Information Retrieval Models
- **Exercise 01**

Exercise

- Given a document collection ...
- Find the results to a query ...
  - Employing the Boolean model
  - Employing the vector model (with TF*IDF)
- Some hints:
  - Excel:
    - Sheet on homepage
    - Use functions “Summenprodukt” & “Quadratesumme”

Exercise

- Document collection (6 documents)
  - spatz, amsel, vogel, drossel, fink, falke, flug
  - spatz, vogel, flug, nest, amsel, amsel, amsel
  - kuckuck, nest, nest, ei, ei, ei, flug, amsel, amsel, vogel
  - amsel, elster, elster, drossel, vogel, ei
  - falke, katze, nest, nest, flug, vogel
  - spatz, spatz, konstruktion, nest, ei
- Queries:
  - spatz, vogel, nest, konstruktion
  - amsel, ei, nest

Exercise

	d1	d2	d3	d4	d6	d6	idf
amsel	1	3	2	1			
drossel	1			1			
ei			3	1		1	
elster				2			
falke	1				1		
fink	1						
flug	1	1	1		1		
katze					1		
konstruktion						1	
kuckuck			1				
nest		1	2		2	1	
spatz	1	1				2	
vogel	1	1	1	1	1		

Thanks ...

http://www.uni-klu.ac.at

for your attention!