VK Multimedia Information Systems

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

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

Addendum: BM25

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

- BM-25 weighting based on Roberston et al.
  - df_j is the document frequency of term j
  - dl is the document length
  - $avdl$ is the average document length across the collection
  - k_1 and b are free parameters

$$w_j(\bar{d}, C) := \frac{(k_1 + 1)d_j}{k_1((1-b) + b\frac{dl}{avdl}) + d_j} log \frac{N - df_j + 0.5}{df_j + 0.5}$$

$$W(\bar{d}, q, C) = \sum_j w_j(\bar{d}, C) \cdot q_j$$

Information Retrieval Basics: Agenda

- **Probabilistic Model**
- Other Retrieval Models
- Common Retrieval Methods
  - Query Modification
  - Co-Occurrence
  - Relevance Feedback
- Retrieval Evaluation
- The Lucene Search Engine
- Exercise 02

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: $\overline{R}$
- Probability that d_j is relevant for q:

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

- Probability that d_j is not relevant for q:

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

Probabilistic Model: Definition

- Similarity *q* and d_j:

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

- Using Bayes' Rule:

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

- 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

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

  - or not constant:

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

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

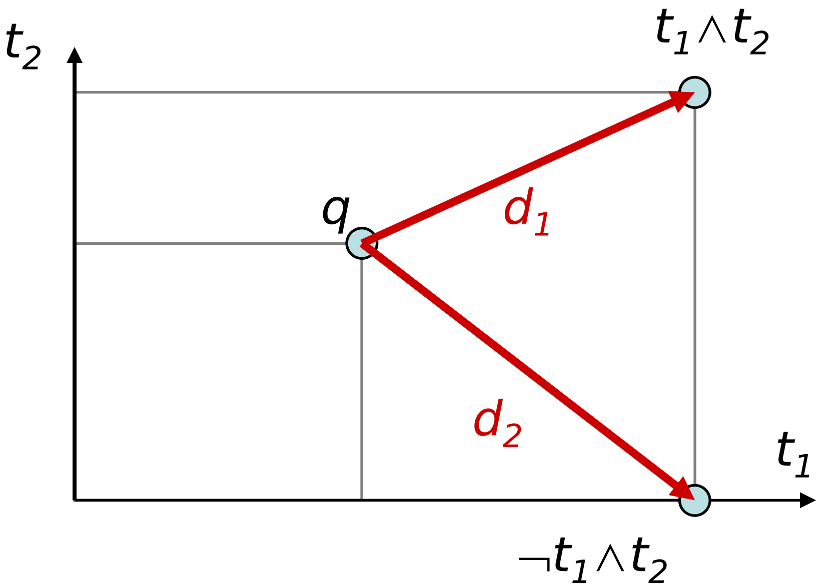
- Probabilistic Model
- **Other Retrieval Models**
- Common Retrieval Methods
  - Query Modification
  - Co-Occurrence
  - Relevance Feedback
- Retrieval Evaluation
- The Lucene Search Engine
- Exercise 02

Set Theoretic Models: Fuzzy Set Model

- Each query term defines a fuzzy set
- Each document has a **degree of membership**
  - e.g. d_1 is part of set of term t_1 at 70%
- Done e.g. with query expansion (co-occurrence or thesaurus)

Set Theoretic Models: Extended Boolean Model

- Incorporates **non binary** weights
- Geometric interpretation:
  - Distance between document vector and
  - desired Boolean state (query)



Algebraic Models: Generalized Vector Space M.

- Term independence not necessary
- Terms (as dimensions) are not orthogonal and may be linear dependent.
- Smaller linear independent units exist.
  - m ... minterm
  - Constructed from co-occurrence: 2^t minterms
- Dimensionality a problem
  - Number of active minterms (which actually occur in a document)
  - Depends on the number of documents

Algebraic Models: Latent Semantic Indexing M.

- Introduced 1988, LSI / LSA
- Concept matching vs. term matching
- Mapping documents & terms to concept space:
  - Fewer dimensions
  - Like clustering

Algebraic Models: Latent Semantic Indexing M.

- Let M_{ij} be the document term matrix
  - with t rows (terms) and N cols (docs)
- Decompose M_{ij} into $K*S*D^t$
  - K .. matrix of eigenvectors from term-to-term (co-occurence) matrix
  - D^t .. matrix of eigenvectors from doc-to-doc matrix
  - S .. $r \times r$ diagonal matrix of singular values with $r=min(t,N)$, the rank of M_{ij}

Algebraic Models: Latent Semantic Indexing M.

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

- With $M_{ij} = K*S*D^t$ ...
- Only the s largest singular values from S:
  - Others are deleted
  - Respective columns in K and D^t remain
- $M_s = K_s*S_s*D^t{}_s$ ...
  - $s < r$ is new rank of M
  - s large enough to fit in all data
  - s small enough to cut out unnecessary details

Algebraic Models: Latent Semantic Indexing M.

- Reduced doc-to-doc matrix:
  - $M^t_s * M_s$ is NxN Matrix quantifying the relationship between documents
- Retrieval is based on pseudo-document
  - Let column *0* in M_{ij} be the query
  - Calculate $M^t_s * M_s$
  - First row (or column) gives the relevance

Algebraic Models: Latent Semantic Indexing M.

- Advantages
  - M even more sparse
  - Retrieval on a “conceptual” level
- Disadvantages
  - Doc-to-doc matrix might be quite big
  - Therefore: Processing time

Example LSA …

Example of text data: Titles of Some Technical Memos

c1:	*Human* machine *interface* for ABC *computer* applications
c2:	A *survey* of *user* opinion of *computer system response time*
c3:	The *EPS user interface* management *system*
c4:	*System* and *human system* engineering testing of *EPS*
c5:	Relation of *user* perceived *response time* to error measurement
m1:	The generation of random, binary, ordered *trees*
m2:	The intersection *graph* of paths in *trees*
m3:	*Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4:	*Graph minors*: A *survey*

from Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. Discourse Processes, 25, 259-284.

Example LSA ...

$$\{X\} =$$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

from Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. Discourse Processes, 25, 259-284.

Example LSA ...

$\{W\} =$

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

$\{S\} =$

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

$\{P\} =$

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

Example LSA ...

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

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Example LSA ...

Correlations between titles in raw data:

	c1	c2	c3	c4	c5	m1	m2	m3
c2	-0.19							
c3	0.00	0.00						
c4	0.00	0.00	0.47					
c5	-0.33	0.58	0.00	-0.31				
m1	-0.17	-0.30	-0.21	-0.16	-0.17			
m2	-0.26	-0.45	-0.32	-0.24	-0.26	0.67		
m3	-0.33	-0.58	-0.41	-0.31	-0.33	0.52	0.77	
m4	-0.33	-0.19	-0.41	-0.31	-0.33	-0.17	0.26	0.56

0.02	
-0.30	0.44

Correlations in two dimensional space:

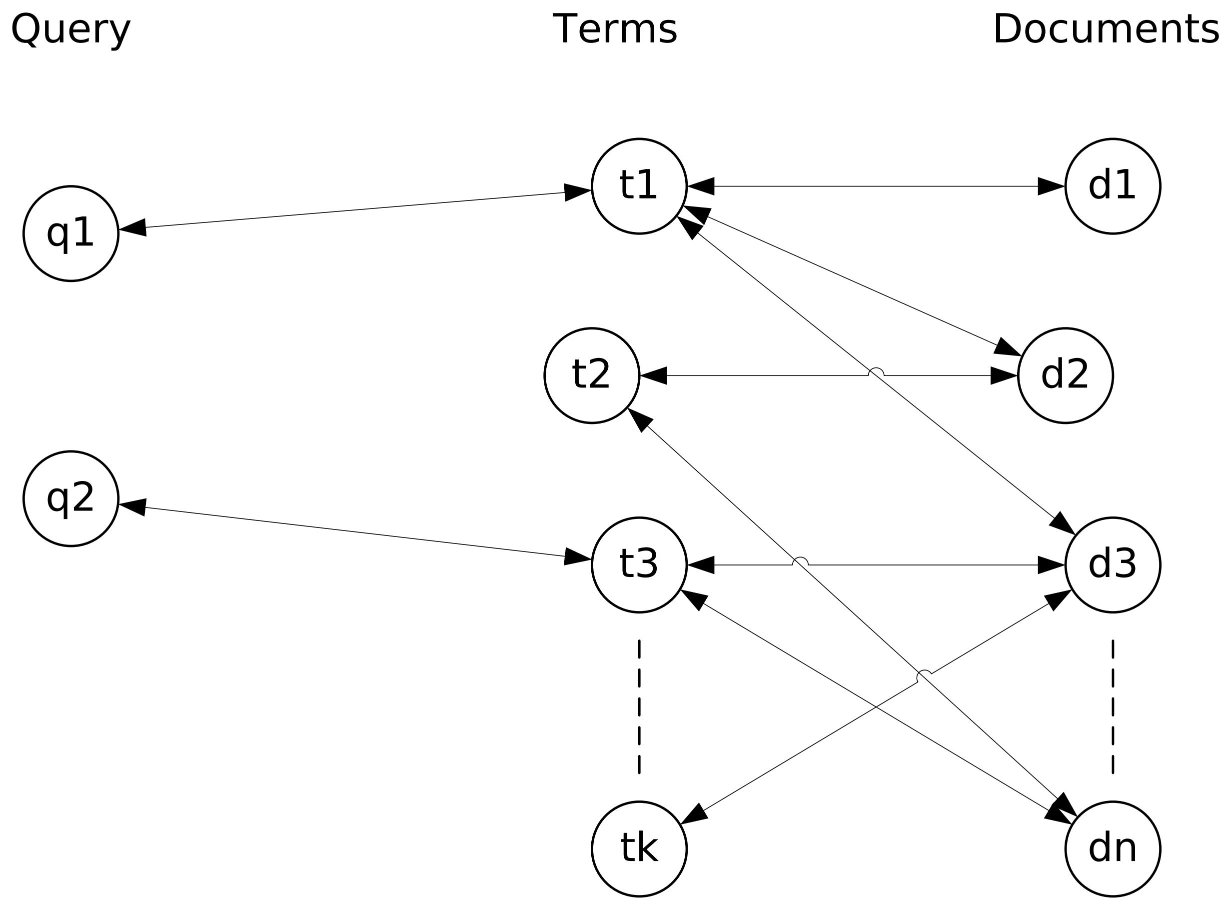
	c1	c2	c3	c4	c5	m1	m2	m3
c2	0.91							
c3	1.00	0.91						
c4	1.00	0.88	1.00					
c5	0.85	0.99	0.85	0.81				
m1	-0.85	-0.56	-0.85	-0.88	-0.45			
m2	-0.85	-0.56	-0.85	-0.88	-0.44	1.00		
m3	-0.85	-0.56	-0.85	-0.88	-0.44	1.00	1.00	
m4	-0.81	-0.50	-0.81	-0.84	-0.37	1.00	1.00	1.00

0.92	
-0.72	1.00

Algebraic Models: Neural Network M. / Associative Retrieval

- Neural Network:
  - Neurons emit signals to other neurons
  - Graph interconnected by synaptic connections
- Three levels:
  - Query terms, terms & documents

Algebraic Models: Neural Network M. / Associative Retrieval



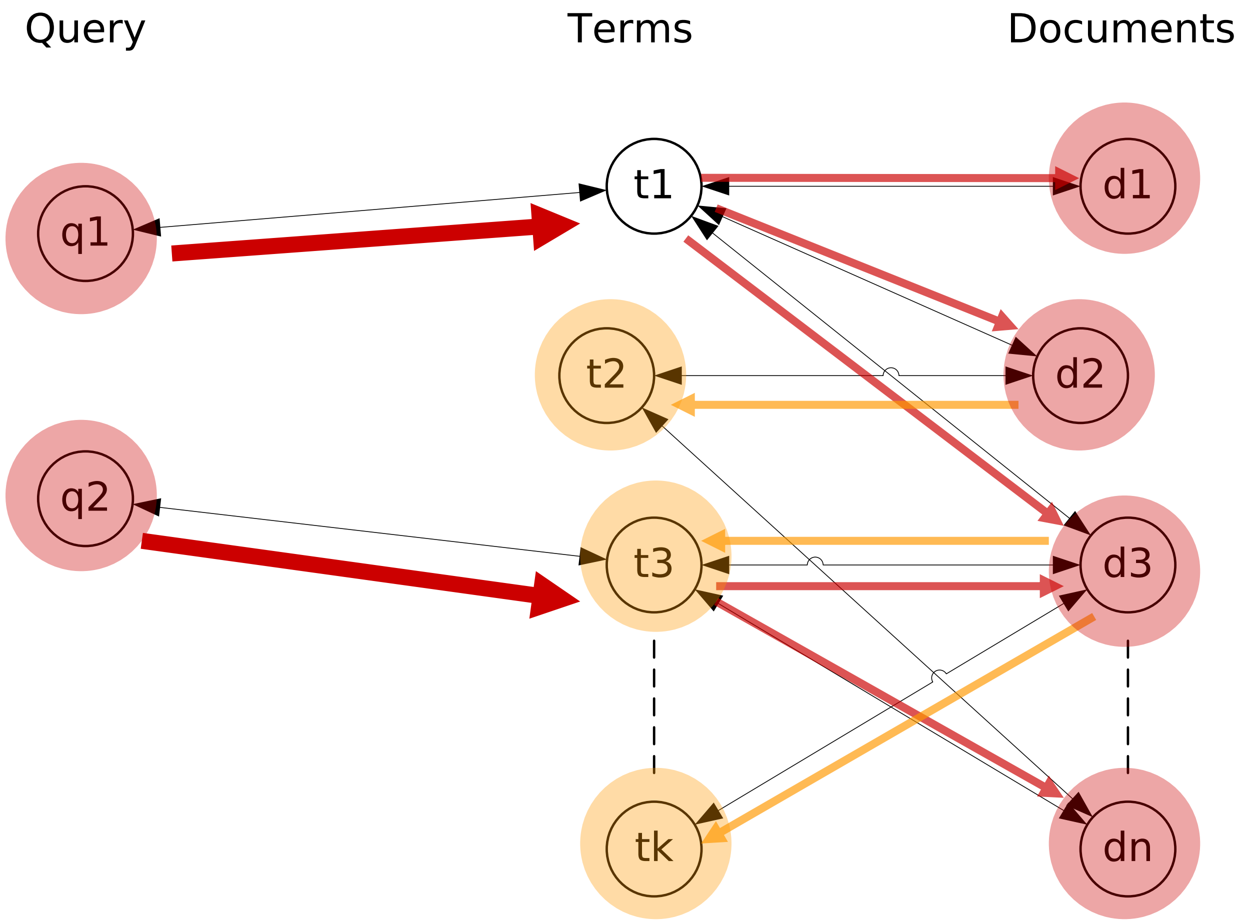
Algebraic Models: Neural Network M. / Associative Retrieval

- Query term is "activated"
  - Usually with weight 1
  - Query term weight is used to "weaken" the signal
- Connected terms receive signal
  - Term weight "weakens" the signal
- Connected documents receive signal
  - Different activation sources are "combined"

Algebraic Models: Neural Network M. / Associative Retrieval

- First round query terms -> terms -> docs
  - Equivalent to vector model
- Further rounds increase retrieval performance

Algebraic Models: Neural Network M. / Associative Retrieval



Algebraic Models: Neural Network M. / Associative Retrieval

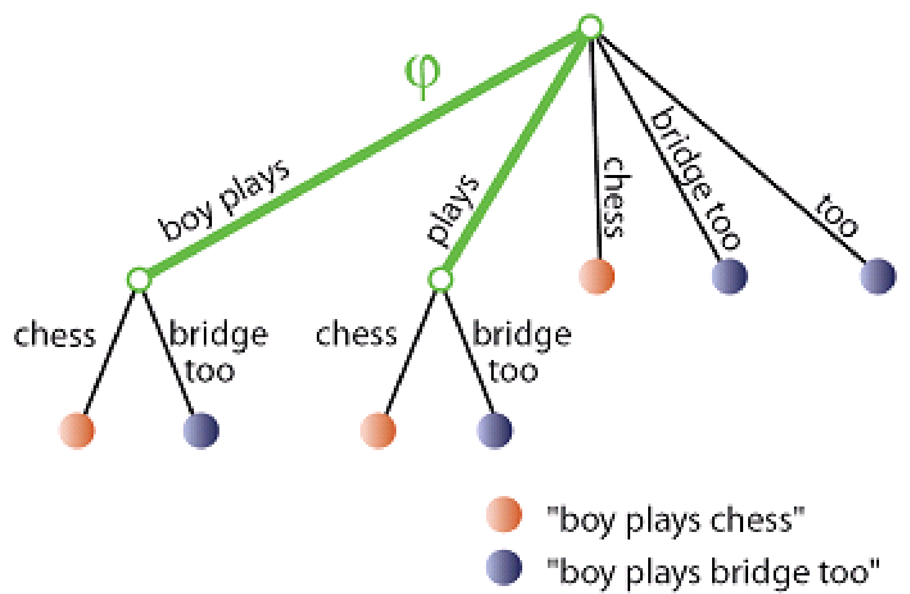
- Advantages
  - Works on generic digraphs
  - Edges can be created on the fly
  - Nodes can be re-weighted on the fly
- Disadvantages
  - Graph might be too big for main memory
  - Tuning of weights is complicated
  - Selection of appropriate concepts: Back-propagation etc.

Alternative Model: Suffix Tree Retrieval M.

- Operates on document suffixes:
  - "The quick brown fox" has the suffixes:
    - *The quick brown fox, quick brown fox, brown fox, fox*
- Integrates word order
  - Therefore terms are not independent
- Builds a tree with the suffixes

Alternative Model: Suffix Tree Retrieval M.

- Example
  - d1 = “boy plays chess”
  - d2 = “boy plays bridge too”



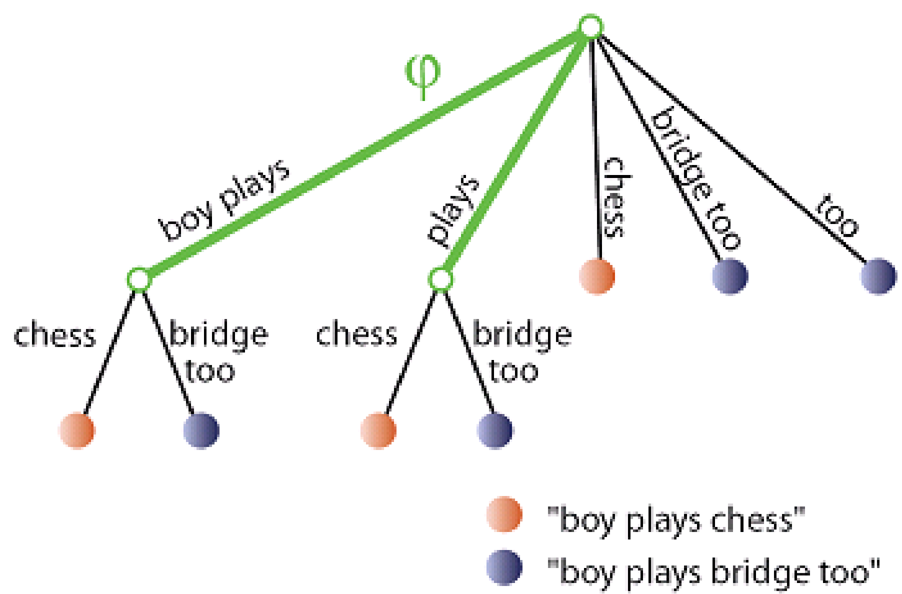
Alternative Model: Suffix Tree Retrieval M.

- Similarity is assessed based on traversed edges in the tree
- Different metrics used as relevance function:
  - Jaccard coefficient
  - TF*IDF weighting

Alternative Model: Suffix Tree Retrieval M.

- Jaccard coefficient
  - Two document d^+ ands d^-
  - Edge sets E^+, E^-: traversed upon insertion of d^+, d^-

$$\varphi_{ST} = \frac{\left| E^+ \cap E^- \right|}{\left| E^+ \cup E^- \right|}$$



Information Retrieval Basics: Agenda

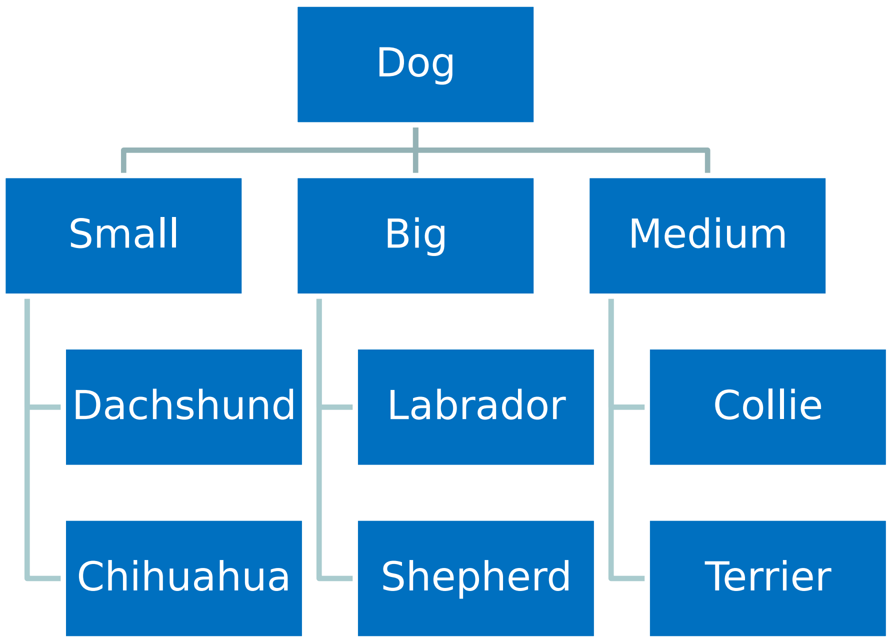
- Probabilistic Model
- Other Retrieval Models
- **Common Retrieval Methods**
  - **Query Modification**
  - Co-Occurrence
  - Relevance Feedback
- Retrieval Evaluation
- The Lucene Search Engine
- Exercise 02

Query Modification

- Query expansion
  - General method to increase either
    - number of results
    - or accuracy
  - Query itself is modified:
    - Terms are added (co-occurrence, thesaurii)

Query Expansion

- Integrate existing knowledge
  - Taxonomies
  - Ontologies
- Modify query
  - Related terms
  - Narrower terms
  - Broader terms



Term Reweighting

- To improve accuracy of ranking
- Query term weights are changed
  - Note: no terms are added / removed
  - Result ranking changes

Information Retrieval Basics: Agenda

- Probabilistic Model
- Other Retrieval Models
- **Common Retrieval Methods**
  - Query Modification
  - **Co-Occurrence**
  - Relevance Feedback
- Retrieval Evaluation
- The Lucene Search Engine
- Exercise 02

Co-Occurrence

- Try to quantify the relation between terms
  - Based on how often they occur together
  - Not based on the position
- Let M_{ij} be the document term matrix
  - with t rows (terms) and N cols (docs)
- $M*M^t$ ($t \times t$) is the “co-occurrence” matrix

Co-Occurrence: Example

	d1	d2	d3	d4	d5
computer	7	7	0	8	3
pda	5	1	4	0	3
cellphone	0	1	5	0	0
wlan	6	1	0	0	4
network	1	2	0	6	0

7	5	0	6	1
7	1	1	1	2
0	4	5	0	0
8	0	0	0	6
3	3	0	4	0

Co-Occurrence: Example

	computer	pda	cellphone	wlan	network
computer	171	51	7	61	69
pda	51	51	21	43	7
cellphone	7	21	26	1	2
wlan	61	43	1	53	8
network	69	7	2	8	41

Co-Occurrence: Weighting

- Inverse term frequency: $itf_j = log(t/t_j)$
  - t_j .. number of distinct index terms in doc d_j
  - t … number of terms
- $f_{i,j}$ … Raw frequency of term i in doc j

$$w_{i,j} = \frac{\left(0,5 + 0,5 \cdot \frac{f_{i,j}}{\max_j (f_{i,j})}\right) \cdot itf_j}{\sqrt{\sum_{l=1}^{N} \left(0,5 + 0,5 \cdot \frac{f_{i,l}}{\max_l (f_{i,l})}\right)^2 \cdot itf_j^2}}$$

Co-Occurrence & Query Expansion

	computer	pda	cellphone	wlan	network
computer	171	51	7	61	69
pda	51	51	21	43	7
cellphone	7	21	26	1	2
wlan	61	43	1	53	8
network	69	7	2	8	41

Query: *cellphone* → Query: *cellphone OR pda*

Information Retrieval Basics: Agenda

- Probabilistic Model
- Other Retrieval Models
- **Common Retrieval Methods**
  - Query Modification
  - Co-Occurrence
  - **Relevance Feedback**
- Retrieval Evaluation
- The Lucene Search Engine
- Exercise 02

Relevance Feedback

- Popular Query Reformulation Strategy:
  - User gets list of docs presented
  - User marks relevant documents
  - In practice ~10-20 docs are presented
  - Query is refined, new search is issued
- Proposed Effect:
  - Query moves more toward relevant docs
  - Away from non relevant docs
  - User does not have to tune herself

Relevance Feedback

- $D_r \subset D$... set of relevant docs identified by the user
- $D_n \subset D$ ... set of non relevant docs
- $C_r \subset D$ ... set of relevant docs
- $\alpha,\ \beta,\ \gamma$ ... tuning parameters

Relevance Feedback

- Considering an optimal query
  - Unlikely and therefore hypothetical
- Which vector retrieves C_r best?

$$\vec{q}_{OPT} = \frac{1}{|C_r|} \cdot \sum_{\forall \vec{d}_j \in C_r} \vec{d}_j - \frac{1}{N - |C_r|} \cdot \sum_{\forall \vec{d}_j \notin C_r} \vec{d}_j$$

Relevance Feedback

Rochio: $\vec{q}_m = \alpha \cdot \vec{q} + \frac{\beta}{|D_r|} \cdot \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \frac{\gamma}{|D_n|} \cdot \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$

Ide: $\vec{q}_m = \alpha \cdot \vec{q} + \beta \cdot \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \cdot \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$

Ide-Dec-Hi: $\vec{q}_m = \alpha \cdot \vec{q} + \beta \cdot \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \max_{non-relevant}(\vec{d}_j)$

Relevance Feedback

- Rochio
  - Based on q_{OPT}, α was 1 in original idea
- Ide
  - $\alpha=\beta=\gamma=1$ in original idea
- Ide-Dec-Hi
  - $\max_{\text{non-relevant}}$ ... highest ranked doc of D_n

- All three techniques yield similar results ...

Relevance Feedback

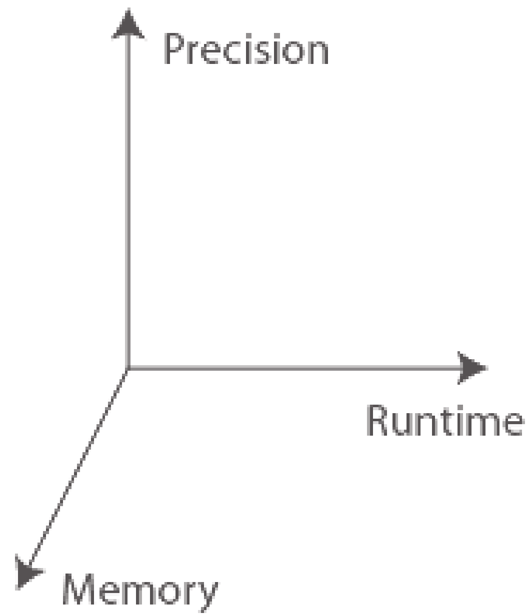
- Evaluation issues:
  - Boosts retrieval performance
  - Relevant documents are ranked top
  - But: Already marked by the user
- Evaluation remains complicated issue

Information Retrieval Basics: Agenda

- Probabilistic Model
- Other Retrieval Models
- Common Retrieval Methods
  - Query Modification
  - Co-Occurrence
  - Relevance Feedback
- **Retrieval Evaluation**
- The Lucene Search Engine
- Exercise 02

Retrieval Evaluation: Motivation

- Compare **objectively** different
  - Search engines
  - Models & Weighting Schemes
  - Methods & Techniques
- Scope
  - Academic
  - Commercial & Industrial
- Different aspects
  - Runtime, Retrieval performance



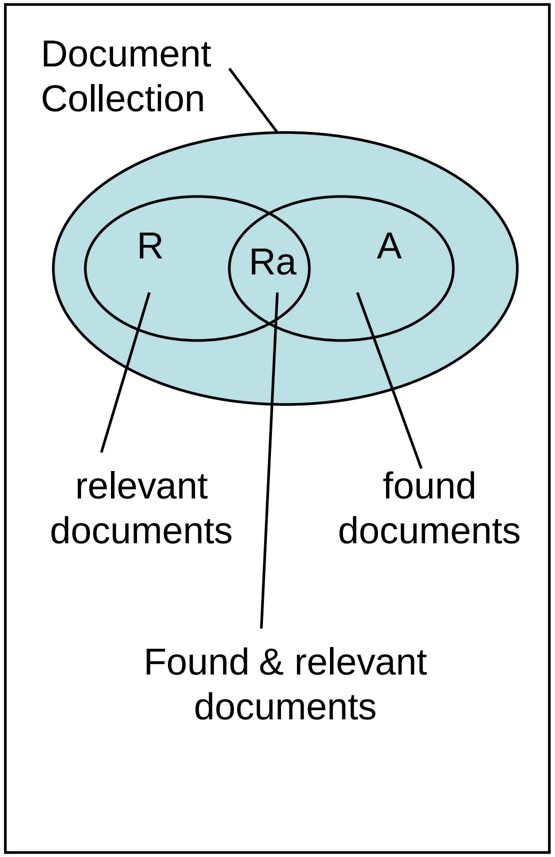
Retrieval Evaluation

- Comparability issues:
  - 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

Metrics: 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**

Metrics: Precision & Recall

- With a query all documents of D have been found (5% of D would be relevant)
  - Precision & Recall?

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

Example

- D = {D00, D01, … D99}
- Query 1:
  - Result Set 1: {**D2, D14, D25**, D76**, D84, D98**}
  - Relevant Docs {D1, D2, D14, D22, D23, D25, D84, D89, D90, D98}
- Query 2:
  - Result Set 1: {**D10, D14**, D60, D63, D77, D95}
  - Relevant Docs {D10, D14}

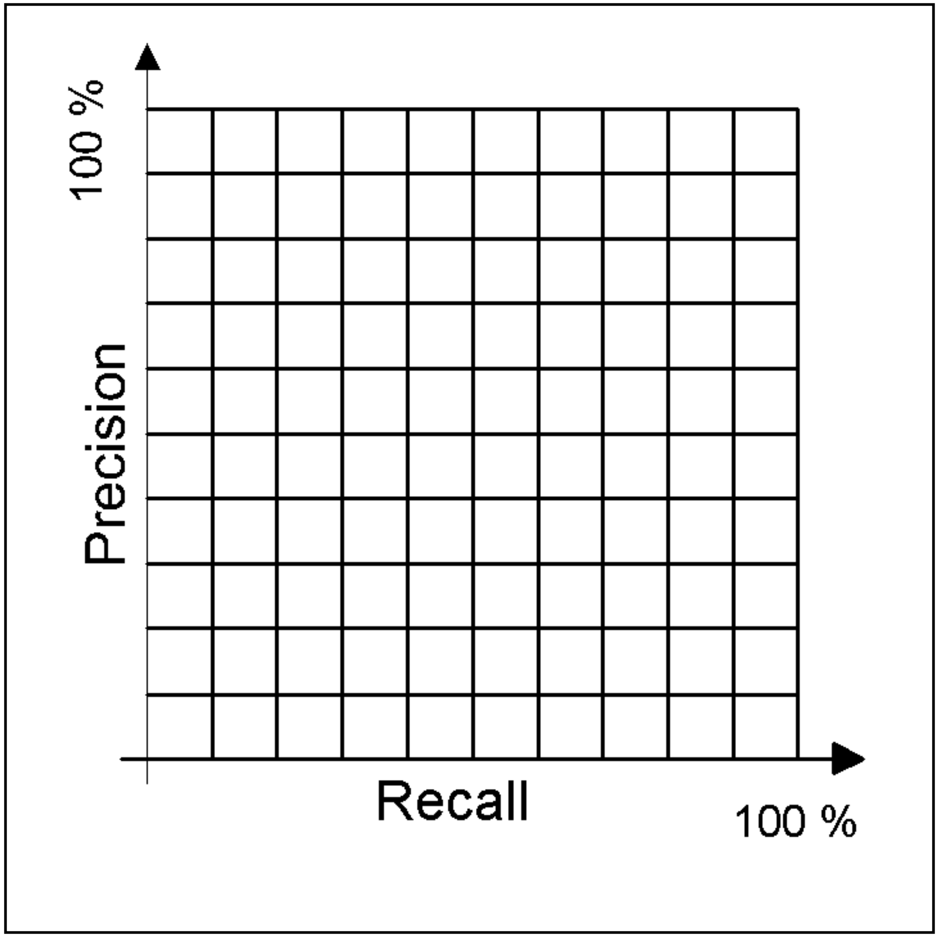
Recall vs. Precision Plot

- 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

01. d123 *
02. d84
03. d56 *
04. d6
05. d8
06. D9 *
07. d511
08. d129
09. d187
10. d25 *
11. d38
12. d48
13. d250
14. d113
15. d3 *

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

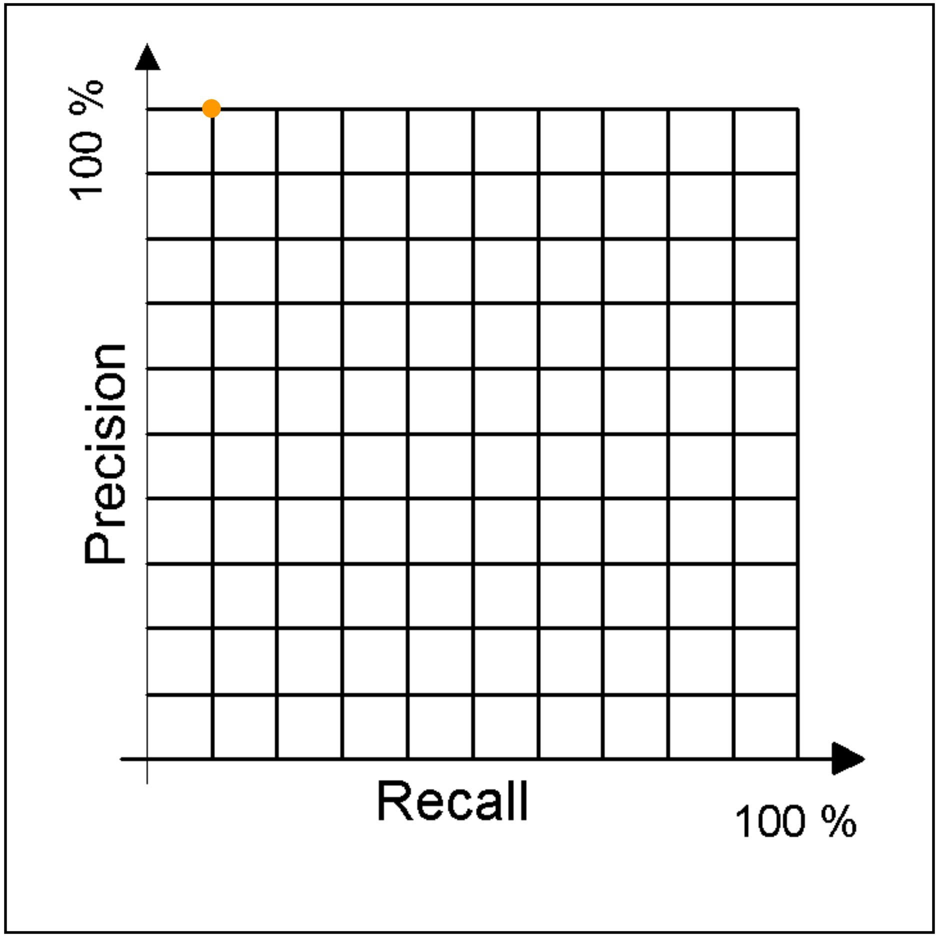


Recall vs. Precision Plot

01. d123 *
02. d84
03. d56 *
04. d6
05. d8
06. D9 *
07. d511
08. d129
09. d187
10. d25 *
11. d38
12. d48
13. d250
14. d113
15. d3 *

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

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

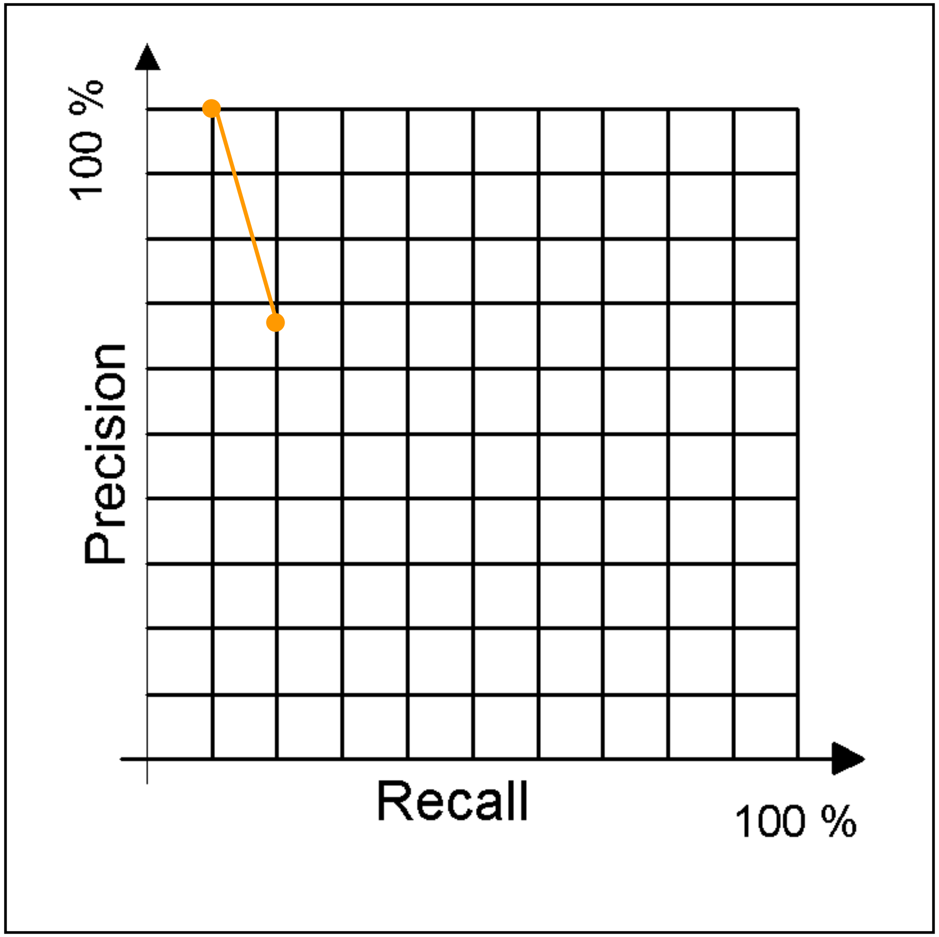


Recall and Precision

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 *

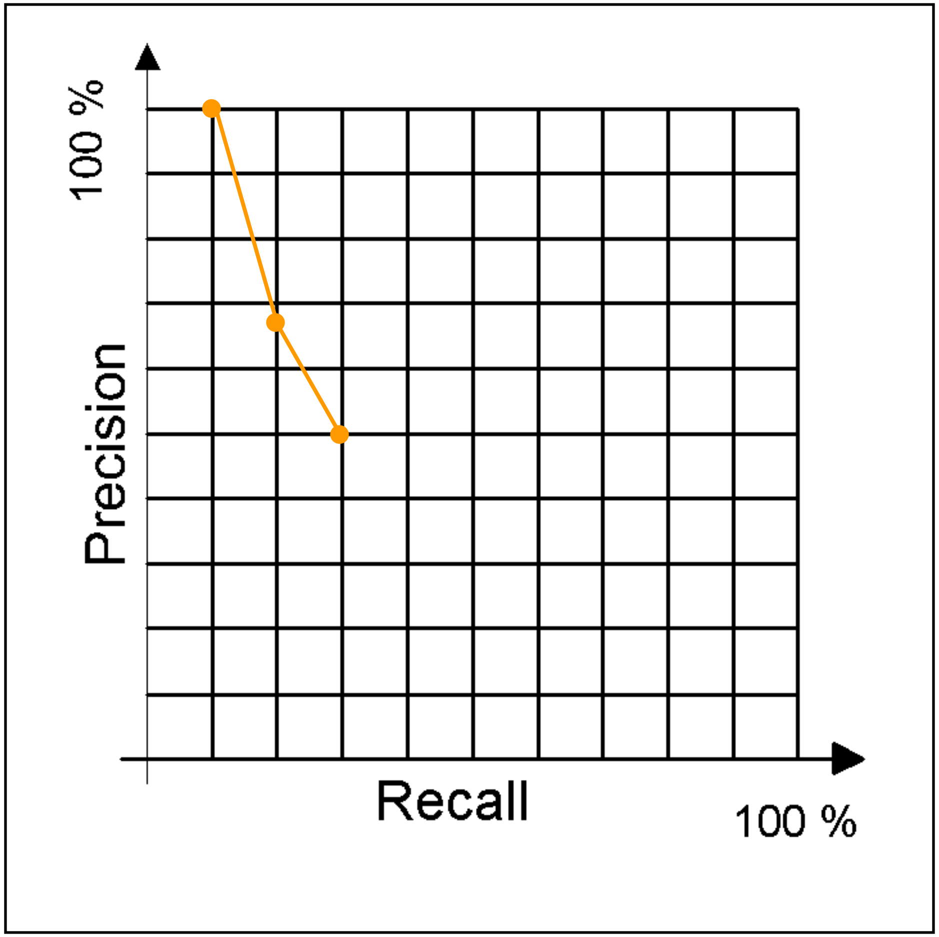
$$\text{Recall} = \frac{|Ra|}{R} = \frac{2}{10}$$

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



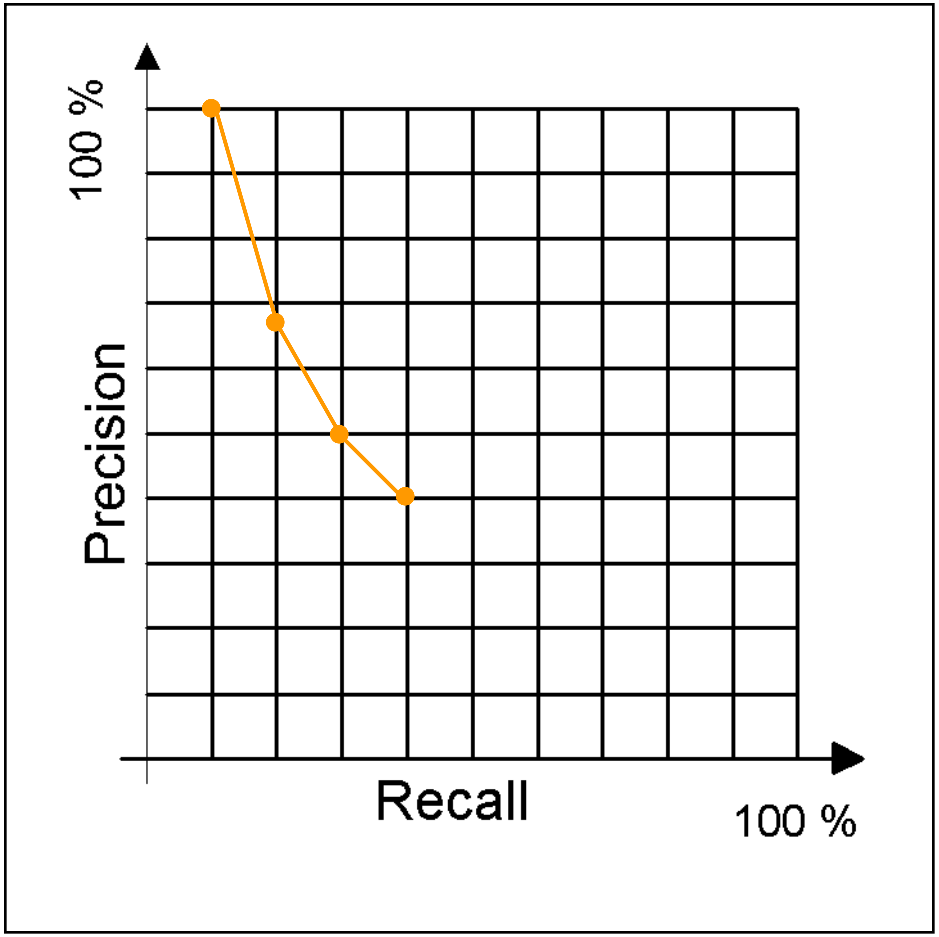
Recall and Precision

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 *



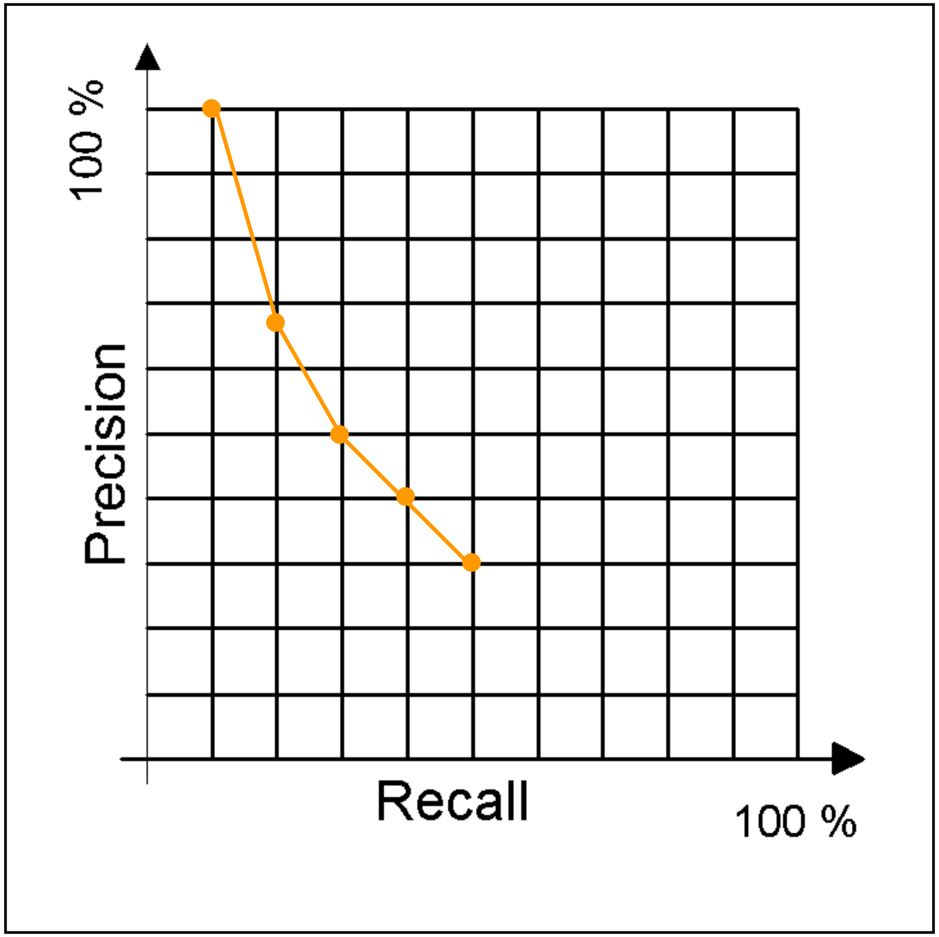
Recall and Precision

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 *



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 *



F-Measure

$$E(j) = 1 - \frac{1 + b^2}{\frac{b^2}{recall(j)} + \frac{1}{precision(j)}}$$

F(j) = 1 – E(j) … van Rijsbergen

- Lower values -> lower performance
- If b=1, F(j) is average
- If b=0, F(j) is precision
- If b=inf, F(j) is recall
- b=2 is a common choice

Mean Average Precision (MAP)

- Find average precision for each query
- Compute mean AP over all queries
  - Macroaverage: All queries are considered equal
- For average recall-precision curves
  - Average at standard recall points

Mean Average Precision (MAP)

Example: Query Q1:

1. D12 (relevant) -> Precision: 1
2. D61
3. D39 (relevant) -> Precision: 2/3
4. D75 (relevant) -> Precision: 3/4
5. D66
6. D14 (relevant) -> Precision: 4/6
7. D52
8. D33 (relevant) -> Precision: 5/8

- Average Precision: (1+2/3+…)/5=0.742

Mean Average Precision (MAP)

- Compute MAP:
  - Q1: 0,742
  - Q2: 0,633
  - Q3: 0,874
  - Q4: 0,722
- MAP = (0,742 + 0,633 + ..) / 4 = 0,743

Test Collections & Initiatives

- Aim:
  - Provide data, topic & results
- Prominent Initiatives
  - Text Retrieval Conference (TREC)
  - INitiative for the Evaluation of XML Retrieval (INEX)
  - Cross Language Evaluation Forume (CLEF)

The TREC Collection

- Aim: Support IR Research on big data collections with
  - Test collection
  - Uniform measures and methods
  - Platform for comparison & challenges
- TREC collection size increases steadily
- Several different tracks:
  - Ad hoc, Web, Blog, Confusion, Genomics Track, Question Answering, Spam, Terabyte
- Examples:
  - Spam: ~ 91.000 messages (300 MB zipped)
  - Ad hoc has 5 sets:
    - e.g. Disk 5: 260.000 documents (1 GB zipped)

Summary: Evaluation

- Lots of measures exist besides Precision & Recall
- Selection based on Use Case & Scenario
- Initiatives & Collections allow comparison
- Also user centered evaluation methods exist
- collections & initiatives are criticized:
  - Handling of outliers, significance of differences, ...

Information Retrieval Basics: Agenda

- Probabilistic Model
- Other Retrieval Models
- Common Retrieval Methods
  - Co-Occurrence
  - Relevance Feedback
- Retrieval Evaluation
- **The Lucene Search Engine**
- Exercise 02

Lucene

- A **Java** text search engine
  - .NET Implementation exists
  - Also used in PHP, etc.
- Initiated by Doug Cutting
  - Now paid by Yahoo!

Lucene

- Implements an **inverted list**
  - Stores term -> document
    - Per field (e.g. title, content, ...)
    - And additional information (count, position, length, etc.)
  - File format & storage.
- Preprocesses input
  - Stemming, etc.
- Provides search & index update
  - Query, Ranking

Lucene: Basic Usage

Let lucene-{version}.jar and lucene-demos-{version}.jar be in your classpath

- To index files type:
  - java org.apache.lucene.demo.IndexFiles [dir]
- To search in the index type:
  - java org.apache.lucene.demo.SearchFiles

Lucene: Queries

- Lucene has an extensive query parser
  - Parses text to internal representation
- Lucene supports several types of queries
  - Field based: title:”multimedia information”
  - Boolean clauses: multimedia AND image
  - Wildcards: te?t OR te*t
  - Fuzzy search: roam~ (e.g. *foam* and *roams*)
  - Proximity search: “java apache”~10
  - Term boosting: java^4 apache

Lucene File Format

- Definitions:
  - An index contains a sequence of documents.
  - A document is a sequence of fields.
  - A field is a named sequence of terms.
  - A term is a string.
- Lucene uses
  - different types of fields:
    - stored, indexed, tokenized
  - Sub-indexes (segments, upon insertion)

Lucene: Usage

- IndexWriter
  - Writes documents to the index
  - Uses Analyzer
- IndexSearcher
  - Searching documents in an index
  - Same Analyzer as for indexing needed
  - A Hits object is returned
- Document
  - Groups fields to logical unit

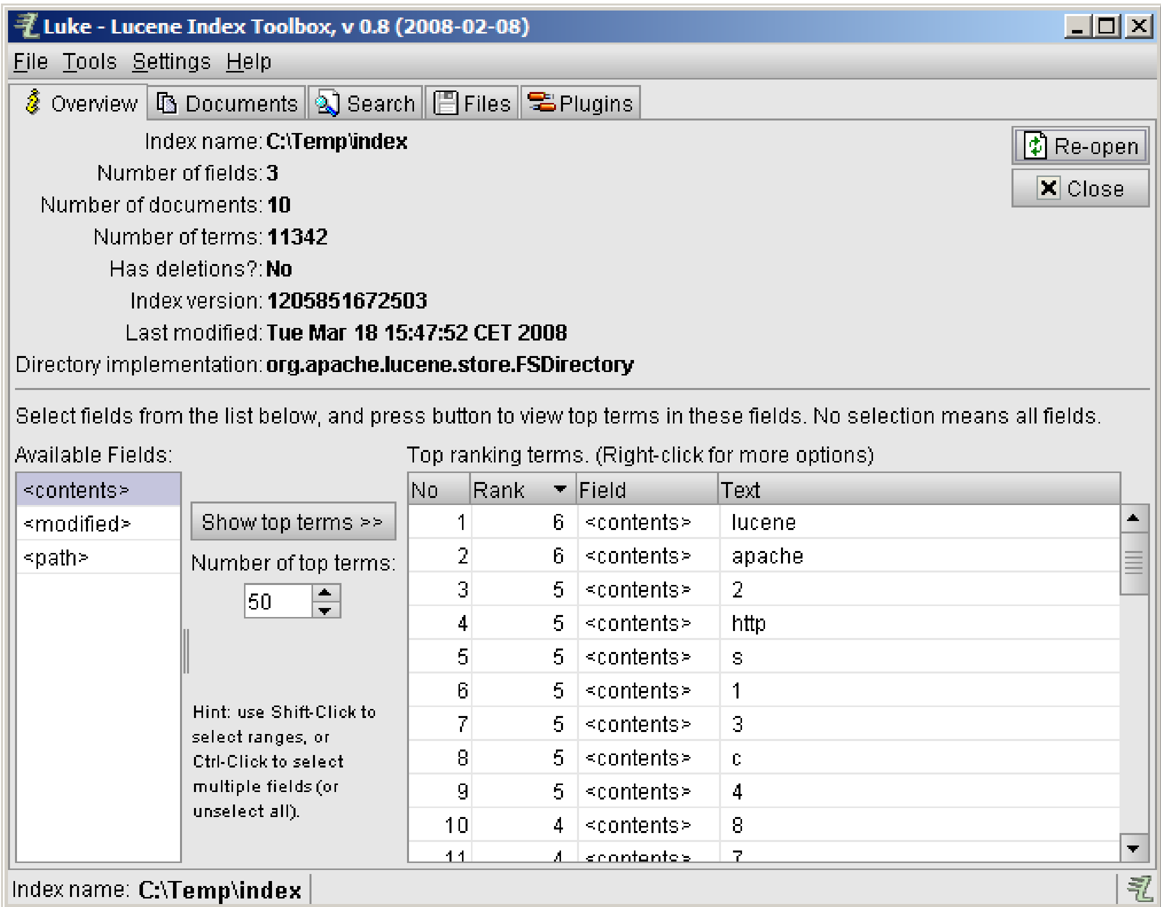
Lucene: Features

- It’s really fast & stable
  - Even compared to commercial products
- Handles multiple indexes
  - MultiReader, distributed search
- Has strong development support
  - Yahoo! & Apache (top level project)
- Lots of Stemmers, Tokenizers, etc.
  - English, German, Korean, Chinese, ...

Lucene: Projects & Tools

- Nutch
  - Open source internet search engine
- Lucene .NET
  - Source code port to .NET
- Solr
  - Search server supporting web services, REST, ..
- Luke
  - GUI index management tool

Luke Demo …


Luke - Lucene Index Toolbox, v 0.8 (2008-02-08)
File Tools Settings Help
Overview Documents Search Files Plugins
Re-open
Close
Index name: C:\Temp\index
Number of fields: 3
Number of documents: 10
Number of terms: 11342
Has deletions?: No
Index version: 1205851672503
Last modified: Tue Mar 18 15:47:52 CET 2008
Directory implementation: org.apache.lucene.store.FSDirectory
Select fields from the list below, and press button to view top terms in these fields. No selection means all fields.
Available Fields:
<contents>
<modified>
<path>
Show top terms >>
Number of top terms:
50
Hint: use Shift-Click to select ranges, or Ctrl-Click to select multiple fields (or unselect all).
Top ranking terms. (Right-click for more options)
No Rank Field Text
1 6 <contents> lucene
2 6 <contents> apache
3 5 <contents> 2
4 5 <contents> http
5 5 <contents> s
6 5 <contents> 1
7 5 <contents> 3
8 5 <contents> c
9 5 <contents> 4
10 4 <contents> 8
Index name: C:\Temp\index

Information Retrieval Basics: Agenda

- Probabilistic Model
- Other Retrieval Models
- Common Retrieval Methods
  - Co-Occurrence
  - Relevance Feedback
- Retrieval Evaluation
- The Lucene Search Engine
- **Exercise 02**

Exercise 02 (a)

- Retrieval Evaluation
  - Gegeben sind eine Collection von 35 Dokumenten
  - Eine Query und eine (ungeordnete) Liste von relevanten Dokumenten zur Query
  - Die gereihten Ergnislisten zweier Suchmaschinen (A1 und A2)
- Ihre Aufgabe
  - Berechnen Sie die Precision & Recall
  - Zeichnen Sie einen Precision vs. Recall Plot
  - Vergleichen Sie die Retrievalperformance der beiden Suchmaschinen.

Exercise 02 (b)

- Berechnen Sie die Assoziationsmatrizen
  - Term 2 Term & Document 2 Document
  - Finden Sie einen Term für eine Query Expansion der Query “Amsel”
  - Finden Sie das Document, das dem Document d1 am ähnlichsten ist.
  - Tipps:
    - Excel MMULT (siehe Hilfe)
    - Kopieren -> Inhalte einfügen (transponiert)

Nicht vergessen!

- Ergebnisse schicken!

Frohe Ostern!