VK Multimedia Information Systems

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

Dienstags, 16.00 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/
Addendum: BM25

- 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

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

\[
 W(\tilde{d}, q, C) = \sum_j w_j(\tilde{d}, C) \cdot q_j
\]
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.
Proabilistic 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: $P(d_j \text{ relevant for } q)/P(d_j \text{ not relevant for } q)$

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 $q$:
  \[ P(R | \tilde{d}_j) \]
- Probability that $d_j$ is not relevant for $q$:
  \[ P(\bar{R} | \tilde{d}_j) \]
Probabilistic Model: Definition

- Similarity \( q \) and \( d_j \):
  \[
  \text{sim}(d_j, q) = \frac{P(R | \tilde{d}_j)}{P(R | \tilde{d}_j)}
  \]

- Using Bayes‘ Rule:
  \[
  \text{sim}(d_j, q) = \frac{P(R | \tilde{d}_j)}{P(R | \tilde{d}_j)} = \frac{P(\tilde{d}_j | R) \times P(R)}{P(\tilde{d}_j | \bar{R}) \times P(\bar{R})}
  \]

- Probability for randomly selecting \( d_j \) out of \( R \)
  \[
  P(\tilde{d}_j | R)
  \]

- Probability for a randomly selected document to be in \( R \)
  \[
  P(R)
  \]
Probabilistic Model: Definition

- As \( P(R) = P(\bar{R}) \)
  \[
  \text{sim}(d_j, q) \approx \frac{P(\bar{d}_j | R)}{P(\bar{d}_j | \bar{R})}
  \]

- Assuming independent index terms:
  \[
  \text{sim}(d_j, q) \approx \frac{\prod_{g_i(\bar{d}_j)=1} P(k_i | R) \times \prod_{g_i(\bar{d}_j)=0} P(\bar{k}_i | R)}{\prod_{g_i(\bar{d}_j)=1} P(k_i | \bar{R}) \times \prod_{g_i(\bar{d}_j)=0} P(\bar{k}_i | \bar{R})}
  \]

- \( P(k_i | R) \) .... Probability that \( k_i \) is in a randomly selected document from \( R \)

- \( P(\bar{k}_i | R) \) .... Probability that \( k_i \) is not in a randomly selected document from \( R \)

- the same for \( P(k_i | \bar{R}) \), \( P(\bar{k}_i | \bar{R}) \)
Probabilistic Model: Definition

Simplification based on

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

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

- Problems
  - \( R \) is not know at query time
  - Therefore we cannot calculate \( P(k_i \mid R) \) and \( P(k_i \mid \overline{R}) \)
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 $\sim$ distribution of index terms $k_i$ in $D$

$$P(k_i \mid R) = 0.5 \quad P(k_i \mid \bar{R}) = \frac{n_i}{N}$$

- $n_i$ ... 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.

\[
P(k_i \mid R) = \frac{V_i}{V} \quad P(k_i \mid \overline{R}) = \frac{n_i - V_i}{N - V}
\]
Probabilistic Model: Starting Probabilities (iii)

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

\[
P(k_i \mid R) = \frac{V_i + 0.5}{V + 1} \quad P(k_i \mid \bar{R}) = \frac{n_i - V_i + 0.5}{N - V + 1}
\]

- or not constant:

\[
P(k_i \mid R) = \frac{V_i + \frac{n_i}{N}}{V + 1} \quad P(k_i \mid \bar{R}) = \frac{n_i - V_i + \frac{n_i}{N}}{N - V + 1}
\]
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)

![Diagram showing geometric interpretation of Extended Boolean Model with vectors and distances labeled: \(d_1\) and \(d_2\).]
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 $K \times S \times D^t$
  - $K$ .. matrix of eigenvectors from term-to-term (co-occurrence) 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.

- With $M_{ij} = K*S*D^t \ldots$
- Only the $s$ largest singular values from $S$:
  - Others are deleted
  - Respective columns in $K$ and $D^t$ remain
- $M_s = K_s*S_s*D_s^t \ldots$
  - $s < r$ 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 $N \times N$ 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 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

Example LSA ...

\[
\{X\} =
\begin{array}{ccccccccc}
& c1 & c2 & c3 & c4 & c5 & m1 & m2 & m3 & m4 \\
\hline
\text{human} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\text{interface} & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{computer} & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{user} & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
\text{system} & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\
\text{response} & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\text{time} & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\text{EPS} & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\text{survey} & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\text{trees} & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
\text{graph} & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\text{minors} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
\end{array}
\]

Example LSA ...

\[
\{W\} =
\begin{array}{cccccccc}
0.22 & -0.11 & 0.29 & -0.41 & -0.11 & -0.34 & 0.52 & -0.06 \\
0.20 & -0.07 & 0.14 & -0.55 & 0.28 & 0.50 & -0.07 & -0.01 \\
0.24 & 0.04 & -0.16 & -0.59 & -0.11 & -0.25 & -0.30 & 0.06 \\
0.40 & 0.06 & -0.34 & 0.10 & 0.33 & 0.38 & 0.00 & 0.00 \\
0.64 & -0.17 & 0.36 & 0.33 & -0.16 & -0.21 & -0.17 & 0.03 \\
0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 \\
0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 \\
0.30 & -0.14 & 0.33 & 0.19 & 0.11 & 0.27 & 0.03 & -0.02 \\
0.21 & 0.27 & -0.18 & -0.03 & -0.54 & 0.08 & -0.47 & -0.04 \\
0.01 & 0.49 & 0.23 & 0.03 & 0.59 & -0.39 & -0.29 & 0.25 \\
0.04 & 0.62 & 0.22 & 0.00 & -0.07 & 0.11 & 0.16 & -0.68 \\
0.03 & 0.45 & 0.14 & -0.01 & -0.30 & 0.28 & 0.34 & 0.68 \\
\end{array}
\]

\[
\{S\} =
\begin{array}{cccccccc}
3.34 & \\
2.54 & 2.35 \\
1.64 & 1.50 \\
1.31 & 0.85 \\
0.56 & 0.36 \\
\end{array}
\]

\[
\{P\} =
\begin{array}{cccccccc}
0.20 & 0.61 & 0.46 & 0.54 & 0.28 & 0.00 & 0.01 & 0.02 \\
-0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 \\
0.11 & -0.50 & 0.21 & 0.57 & -0.51 & 0.10 & 0.19 & 0.25 \\
-0.95 & -0.03 & 0.04 & 0.27 & 0.15 & 0.02 & 0.02 & 0.01 \\
0.05 & -0.21 & 0.38 & -0.21 & 0.33 & 0.39 & 0.35 & 0.15 \\
-0.08 & -0.26 & 0.72 & -0.37 & 0.03 & -0.30 & -0.21 & 0.00 \\
0.18 & -0.43 & 0.24 & 0.26 & 0.67 & -0.34 & -0.15 & 0.25 \\
-0.01 & 0.05 & 0.01 & -0.02 & -0.06 & 0.45 & -0.76 & 0.45 \\
-0.06 & 0.24 & 0.02 & -0.08 & -0.26 & -0.62 & 0.02 & 0.52 \\
\end{array}
\]
Example LSA ...

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

<table>
<thead>
<tr>
<th></th>
<th>c1</th>
<th>c2</th>
<th>c3</th>
<th>c4</th>
<th>c5</th>
<th>m1</th>
<th>m2</th>
<th>m3</th>
<th>m4</th>
</tr>
</thead>
<tbody>
<tr>
<td>human</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>interface</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>computer</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>user</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>system</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>response</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>time</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EPS</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>survey</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>trees</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>graph</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>minors</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Example LSA ...

Correlations between titles in raw data:

<table>
<thead>
<tr>
<th></th>
<th>c1</th>
<th>c2</th>
<th>c3</th>
<th>c4</th>
<th>c5</th>
<th>m1</th>
<th>m2</th>
<th>m3</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2</td>
<td>-0.19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c3</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c4</td>
<td>0.00</td>
<td>0.00</td>
<td>0.47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c5</td>
<td>-0.33</td>
<td>0.58</td>
<td>0.00</td>
<td>-0.31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m1</td>
<td>-0.17</td>
<td>-0.30</td>
<td>-0.21</td>
<td>-0.16</td>
<td>-0.17</td>
<td>0.67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m2</td>
<td>-0.26</td>
<td>-0.45</td>
<td>-0.32</td>
<td>-0.24</td>
<td>-0.26</td>
<td></td>
<td>0.77</td>
<td></td>
</tr>
<tr>
<td>m3</td>
<td>-0.33</td>
<td>-0.58</td>
<td>-0.41</td>
<td>-0.31</td>
<td>-0.33</td>
<td>0.52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m4</td>
<td>-0.33</td>
<td>-0.19</td>
<td>-0.41</td>
<td>-0.31</td>
<td>-0.33</td>
<td>-0.17</td>
<td>0.26</td>
<td>0.56</td>
</tr>
</tbody>
</table>

Correlations in two dimensional space:

<table>
<thead>
<tr>
<th></th>
<th>c1</th>
<th>c2</th>
<th>c3</th>
<th>c4</th>
<th>c5</th>
<th>m1</th>
<th>m2</th>
<th>m3</th>
<th>m4</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2</td>
<td>0.91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c3</td>
<td>1.00</td>
<td>0.91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c4</td>
<td>1.00</td>
<td>0.88</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c5</td>
<td>0.85</td>
<td>0.99</td>
<td>0.85</td>
<td>0.81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m1</td>
<td>-0.85</td>
<td>-0.56</td>
<td>-0.85</td>
<td>-0.88</td>
<td>-0.45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m2</td>
<td>-0.85</td>
<td>-0.56</td>
<td>-0.85</td>
<td>-0.88</td>
<td>-0.44</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m3</td>
<td>-0.85</td>
<td>-0.56</td>
<td>-0.85</td>
<td>-0.88</td>
<td>-0.44</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m4</td>
<td>-0.81</td>
<td>-0.50</td>
<td>-0.81</td>
<td>-0.84</td>
<td>-0.37</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>

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

Query

$$q_1$$

Terms

$$t_1$$

$$t_2$$

$$t_3$$

$$t_k$$

Documents

$$d_1$$

$$d_2$$

$$d_3$$

$$d_n$$
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

Query

Terms

Documents

q1

t1

d1

t2

d2

t3

d3

q2

tk
	dn

\[ q_1 \rightarrow t_1 \rightarrow d_1 \]

\[ q_2 \rightarrow t_2 \rightarrow d_2 \]

\[ q_2 \rightarrow t_3 \rightarrow d_3 \]

\[ q_2 \rightarrow t_k \rightarrow d_n \]
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
  - $d_1 = \text{“boy plays chess”}$
  - $d_2 = \text{“boy plays bridge too”}$
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 $d^+$, $d^-$

$$\varphi_{ST} = \left| \frac{E^+ \cap E^-}{E^+ \cup E^-} \right|$$
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)
● $M \times M^t$ ($t \times t$) is the “co-occurrence” matrix
Co-Occurrence: Example

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
</tr>
</thead>
<tbody>
<tr>
<td>computer</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>pda</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>cellphone</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wlan</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>network</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

The table on the right shows the co-occurrence matrix for the given terms.
Co-Occurrence: Example

<table>
<thead>
<tr>
<th></th>
<th>computer</th>
<th>pda</th>
<th>cellphone</th>
<th>wlan</th>
<th>network</th>
</tr>
</thead>
<tbody>
<tr>
<td>computer</td>
<td>171</td>
<td>51</td>
<td>7</td>
<td>61</td>
<td>69</td>
</tr>
<tr>
<td>pda</td>
<td>51</td>
<td>51</td>
<td>21</td>
<td>43</td>
<td>7</td>
</tr>
<tr>
<td>cellphone</td>
<td>7</td>
<td>21</td>
<td>26</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>wlan</td>
<td>61</td>
<td>43</td>
<td>1</td>
<td>53</td>
<td>8</td>
</tr>
<tr>
<td>network</td>
<td>69</td>
<td>7</td>
<td>2</td>
<td>8</td>
<td>41</td>
</tr>
</tbody>
</table>
Co-Occurrence: Weighting

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

$$w_{i,j} = \sqrt{\frac{(0.5 + 0.5 \cdot \frac{f_{i,j}}{\max_j (f_{i,j})}) \cdot \text{itf}_j}{\sum_{l=1}^{N} (0.5 + 0.5 \cdot \frac{f_{i,l}}{\max_l (f_{i,l})})^2 \cdot \text{itf}_j^2}}}$$
# Co-Occurrence & Query Expansion

<table>
<thead>
<tr>
<th></th>
<th>computer</th>
<th>pda</th>
<th>cellphone</th>
<th>wlan</th>
<th>network</th>
</tr>
</thead>
<tbody>
<tr>
<td>computer</td>
<td>171</td>
<td>51</td>
<td>7</td>
<td>61</td>
<td>69</td>
</tr>
<tr>
<td>pda</td>
<td>51</td>
<td>51</td>
<td>21</td>
<td>43</td>
<td>7</td>
</tr>
<tr>
<td>cellphone</td>
<td>7</td>
<td>21</td>
<td>26</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>wlan</td>
<td>61</td>
<td>43</td>
<td>1</td>
<td>53</td>
<td>8</td>
</tr>
<tr>
<td>network</td>
<td>69</td>
<td>7</td>
<td>2</td>
<td>8</td>
<td>41</td>
</tr>
</tbody>
</table>

**Query:** cellphone

**Result:** 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: \( \tilde{q}_m = \alpha \cdot \tilde{q} + \frac{\beta}{|D_r|} \sum_{\forall d_j \in D_r} \tilde{d}_j - \frac{\gamma}{|D_n|} \sum_{\forall d_j \in D_n} \tilde{d}_j \)

Ide: \( \tilde{q}_m = \alpha \cdot \tilde{q} + \beta \sum_{\forall d_j \in D_r} \tilde{d}_j - \gamma \sum_{\forall d_j \in D_n} \tilde{d}_j \)

Ide-Dec-Hi: \( \tilde{q}_m = \alpha \cdot \tilde{q} + \beta \sum_{\forall d_j \in D_r} \tilde{d}_j - \gamma \max_{\text{non-relevant}} (\tilde{d}_j) \)
Relevance Feedback

- Rochio
  - Based on $q_{OPT}$, $\alpha$ 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 documents
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

- $D = \{D00, D01, \ldots, D99\}$
- **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

Rq={d3, d5, d9, d25, d39, d44, d56, d71, d89, d123} \rightarrow 10
Recall vs. Precision Plot

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

Precision $= \frac{|Ra|}{A} = \frac{1}{1}$
Recall and Precision

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

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


Precision

Recall

100 % 100 %
## Recall and Precision

<table>
<thead>
<tr>
<th>Rank</th>
<th>Document</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>d123</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>d84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>d56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>d6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>d8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>D9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>d511</td>
<td></td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>d129</td>
<td></td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>d187</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>d25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>d38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>d48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>d250</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>d113</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Graph showing the relationship between Recall and Precision](#)
F-Measure

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

\[ F(j) = 1 - E(j) \] ... van Rijsbergen

- Lower values -> lower performance
- If \( b=1 \), \( F(j) \) is average
- If \( b=0 \), \( F(j) \) is precision
- If \( b=\infty \), \( F(j) \) 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: \( \frac{1+2/3+\ldots}{5}=0.742 \)
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
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 Ergebnislisten 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!

(CC) by annia316 ☺ - http://www.flickr.com/photos/annia316