Information Retrieval and Data Mining (IRDM) SS 2015
Prof. Dr.-Ing. Sebastian Michel
MSc. Koninika Pal
TU Kaiserslautern, FB Informatik – Lehrgebiet Informationssysteme
Sheet 3: Handout 20.05.2015, Presentation 02.06.2015

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

http://dbis.informatik.uni-kl.de

Assignment 1: BIM with Term Dependence Tree (1 P.)

(a) Consider the query {*q:="Michael Jordan computer science"*} with the four terms $t_1 = Michael, t_2 = Jordan, t_3 = computer, t_4 = science$. An initial query evaluation returns the documents $d_1, \ldots, d_{10}$ that are intellectually evaluated by a human user. The occurrences of the terms $t_1, \ldots, t_4$ in the documents as well as the relevance feedback of the user are depicted in the following table, where "1" points out a relevant document and "0" points out a non-relevant document.

	t_1	t_2	t_3	t_4	**Relevant**
d_1	**1**	0	**1**	0	0
d_2	**1**	**1**	0	0	0
d_3	**1**	0	0	0	0
d_4	0	**1**	**1**	**1**	**1**
d_5	**1**	**1**	**1**	**1**	**1**
d_6	0	**1**	0	**1**	**1**
d_7	0	**1**	**1**	0	0
d_8	**1**	0	**1**	**1**	**1**
d_9	**1**	**1**	0	0	0
d_{10}	**1**	**1**	0	0	0

Consider the following document d_{11}

	t_1	t_2	t_3	t_4
d_{11}	0	**1**	0	**1**

Compute the similarities of document d_{11} to the given query using the probabilistic retrieval model with relevance feedback according to the formula by *Robertson & Spärck-Jones* with Lidstone smoothing ($\lambda = 0.5$) but considering maximum spanning tree created from the term dependence tree for relevant and non-relevant documents. The similarity of a document is calculated using the formula

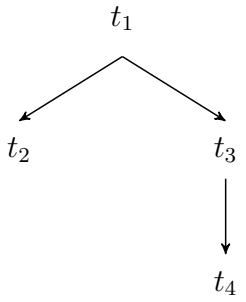
$$sim(d,q) = \sum_{t \in q} d_t \log \frac{p_{t|\text{parent}_t}}{1 - p_{t|\text{parent}_t}} + \sum_{t \in q} d_t \log \frac{1 - q_{t|\text{parent}_t}}{q_{t|\text{parent}_t}}$$

where, $p_{t|\text{parent}_t}$ and $q_{t|\text{parent}_t}$ are considered as conditional probability of that term t appears in relevant document with respect to whether or not its parent term (denoted as parent_t) appears in d, respectively for irrelevant documents in case of $q_{t|\text{parent}_t}$. For instance, for d_{11} and t_2 we have

$$p_{t_2|\text{parent}_{t_2}} = \frac{|t_2 = 1 \cap t_1 = 0 \cap R = 1| + 0.5}{|t_1 = 0 \cap R = 1| + 1}$$

note that t_1 does not appear in d_{11}. We compute q_t analogously. In principle, for the root term we simply take $p_{t|\text{parent}_t}$ equals to p_t, but in this example t_1 does anyway not appear in d_{11}.

The maximum spanning tree for both relevant and non-relevant documents looks as follows:



Assignment 2: Language Model with different Smoothings (1 P.)

Suppose we want to search in the following collection of Christmas cookie recipes. The numbers in the table below indicate raw term frequencies.

	milk	pepper	raisins	sugar	cinnamon	apples	flour	eggs	clove	jelly
d_1	4	0	0	4	0	1	1	0	0	0
d_2	1	1	0	2	0	0	0	0	1	0
d_3	3	1	0	2	0	0	0	2	0	0
d_4	1	2	1	1	2	0	2	1	0	0
d_5	2	0	2	0	1	0	5	2	1	2
d_6	1	0	0	0	0	0	1	1	0	2
d_7	2	1	0	0	1	0	0	0	0	1
d_8	0	0	3	2	0	1	0	4	0	0

(a) Determine the top-3 documents including their query likelihoods for the query

$$q_1 = \langle \text{sugar}, \text{raisins}, \text{cinnamon} \rangle$$

using the multinomial model (i.e., $P(q|d) = \prod_{t \in q} P(t|d)$) with MLE probabilities $P(t|d)$.

(b) Determine the top-3 documents when using Jelinek-Mercer smoothing ($\lambda = 0.5$).

(c) Determine the top-3 documents when using Dirichlet smoothing (for a suitable α)

Assignment 3: Latent Semantic Indexing (1 P.)

We suggest to use R (as briefly mentioned in the lecture) to solve this assignment. Alternatively, you can use Python or your favorite language/tool, but be able to demonstrate your approach/solution.
Consider the following term-document matrix.

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}
human	2	1	1	0	0	0	0	0	0	1	0	0
genome	1	2	0	1	0	0	0	0	0	0	0	0
genetic	1	2	1	2	1	0	0	0	0	0	1	0
molecular	0	1	2	1	0	0	0	1	0	0	0	1
host	0	0	0	0	1	1	2	0	0	0	0	0
bacteria	0	0	0	0	1	2	1	1	0	0	0	0
resistance	0	1	0	1	0	1	3	2	0	0	0	0
disease	0	0	1	1	1	2	2	3	0	0	0	0
computer	1	0	0	0	0	0	0	0	2	2	1	0
information	0	0	1	0	0	0	2	2	3	0	1	0
data	1	0	0	0	0	0	1	0	1	1	1	2

Here, we want to understand the topic space of the collection of these documents using LSI.

(a) How many dimension of the topic space you want to reduce to remove noise without loosing valuable information? Explain the justification behind your answer.

(b) Determine top-3 similar documents for following query using LSI on the reduced topic space according to the dimensions you have chosen in part (a):

$$q = \langle 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0 \rangle$$

(c) Determine the most related word to *gene* which appears in document d_1, d_2, d_4, d_5, and d_{11} i.e.,

$$gene = \langle 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0 \rangle$$

.