Information Retrieval and Data Mining (IRDM) SS 2015
Prof. Dr.-Ing. Sebastian Michel
MSc. Koninika Pal
TU Kaiserslautern, FB Informatik – Lehrgebiet Informationssysteme
Sheet 1: Handout 22.04.2015, Presentation 06.05.2015

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

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

Assignment 1: Stopword Removal and Edit Distance (1 P.)

(a) Consider the problem of building a search engine for an English document collection consisting of 10^{42} term occurrences. We would like to estimate how much space we can save by removing stopwords from the collection. As discussed in the lecture, one way to determine stopwords is to select the k most frequent words. Assuming that the space consumption (e.g., on hard disk) of your system is proportional to the number of term occurrences that it indexes, how much space (as a percentage) can you save by choosing $k = 10$? How much by choosing $k = 100$?

(b) Compute the Hamming distance, longest common subsequence, and Levenshtein distance (including the dynamic-programming table) for the string pairs: (i) *phone* and *alpha*, and (ii) *boolean* and *root*.

(c) The Levenshtein distance assumes uniform cost for edit operations. Extend it to handle non-uniform costs c_I, c_D, and c_R for inserting, deleting, and replacing an arbitrary character. Can you still use dynamic programming?

Hints. *For part (a), you can write code to compute the exact result or estimate results using the harmonic series approximation* $\sum_{i=1}^{n} \frac{1}{i} \approx ln\, n + \gamma$ *where* $\gamma = 0.5772156649$ *is the Euler–Mascheroni constant.*

Assignment 2: Ranking Model (1 P.)

Suppose we want to search the following collection of christmas cookie recipes. Assume that the numbers in the table 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

Consider the query $q = \langle$ sugar, milk $\rangle$, i.e., $q = \langle\, 1\,0\,0\,1\,0\,0\,0\,0\,0\,0 \,\rangle$ and perform the following tasks.

(a) Rank the documents according to TF*IDF model.

(b) Assume that we use the vector space model with the following TF*IDF variant

$$tf.idf(t,d) = tf(t,d) \times \log \frac{|D|}{df(t)} \,.$$

and determine the top-3 documents according to cosine similarity.

Assignment 3: Retrieval Effectiveness (1 P.)

Consider a search engine executing a query over a corpus with $n = 1,000$ documents and returning a ranked list with $r = 50$ results. A human expert assesses the result list and considers the documents at the following ranks as relevant:

$$2, 3, 5, 6, 7, 8, 10, 12, 14, 19, 21, 23, 29, 30, 37, 42, 46, 48, 50.$$

The same expert also says that the entire corpus contains 100 documents that would be relevant for the given query.

(a) Compute precision, recall, accuracy, $F_{0.5}$, F_1, and F_2 for the methods. What are the overall recall, the precision@20, and the precision@10 (i.e., for the top-10 results only), and the F_1 measure at the top-20? What is the MAP (mean average precision)? Assume that all relevant documents not retrieved in the top 50 are positioned at the bottom ranks of the collection (i.e., on the last ranks until 1000).

(b) Assume we compare the previous ranked list to the result of a Boolean search engine which returns $r' = 10$ documents only, all of which are however relevant. At what ranks does the ranked search engine beat the Boolean search engine in terms of precision, recall, and F_1-measure? (Assume that the Boolean search engine returns all its results in the form of a "pseudo" ranked list consisting of only the 10 documents which it found to be relevant.)

(c) Assume that the expert has given grades 0 (irrelevant), 1 (somewhat relevant), or 2 (very relevant) as shown in the table below:

doc. at rank	2	3	5	6	7	8	10	12	14	19	21	23	29	30	37	42	46	48	50
grade	1	2	2	2	1	1	1	1	1	2	1	1	1	2	1	1	2	1	1

All ranks not shown in the list have grade 0. Compute the DCG and NDCG values for the top-3, top-5, and top-10 results.