Chair for Algorithms
and Data Structures
Prof. Dr. Hannah Bast
Elmar Haußmann

Information Retrieval
WS 2015/2016

http://ad-wiki.informatik.uni-freiburg.de/teaching

UNI FREIBURG

Exercise Sheet 8

Submit until Tuesday, December 15 at 4:00pm

Exercise 1 (5 points)

Copy your code from Exercise Sheet 2 (or from the master solutions for that sheet if you prefer) to a new folder *sheet-08*. Extend your code by a method *preprocessing_vsm* that builds the term-document matrix from the inverted index (using BM25 scores as entries).

Then add a method *process_query_vsm* such that the result list is obtained via a multiplication of the term-document matrix with the query vector (and not via merging of the inverted lists, like in Exercise Sheet 2).

Note 1: You can use the code from the lecture as an orientation. The only difference is that you should use BM25 scores, whereas in the lecture we used simple tf scores.

Note 2: For debugging, it might be useful to keep your old *process_query* and *merge* methods around. However, please remove them in the final version of your submission: they will only make it harder for your tutor to understand your code and give you meaningful feedback.

Exercise 2 (5 points)

Extend your code by a method *process_query_lsi* that computes results by projecting both the query vector and the term-document matrix into the latent space (Variant 2 from the lecture). The necessary preprocessing (in particular, the computation of the SVD) should be done in a method *preprocessing_vsm*. For your term-document matrix only use the m most frequent terms to keep the matrix small. The dimension k of the latent space and the number m of terms should be second and third command-line arguments (after the file name).

Exercise 3 (5 points)

Repeat the evaluation of your system on the benchmark from Exercise Sheet 2, without LSI (using your code from Exercise 1), with LSI (using your code from Exercise 2), and using a linear combination of the two (with parameter λ, as explained in the lecture). Play around with the parameters m, k and λ and post your best results in the result table on the Wiki. Leave the BM25 parameters constant for the three runs.

[please turn over in low dimension]

Exercise 4 (5 points)

Add a method *related_term_pairs* that computes the term-term association matrix T explained in the lecture (based on the SVD), and returns the 50 term pairs (only unique pairs of different terms please) with the highest values in that matrix. Write these term pairs to a file *term_pairs.txt* (one term pair + value in T per line) and commit that file to the SVN as part of your solution. Briefly(!) discuss in your *experiences.txt* why you think that LSI "found" some of these term pairs.

Add your code and your file *term_pairs.txt* from Exercise 4 to a new sub-directory *sheet-08* of your folder in the course SVN, and commit it. As usual, make sure that *compile*, *test*, and *checkstyle* run through without errors on Jenkins. And let us partake in your adventures for this sheet in the usual *experiences.txt*.