ifis
Institut für Informationssysteme
Technische Universität Braunschweig

Information Retrieval and Web Search Engines
Summer Semester 2010

Prof. Dr. Wolf-Tilo Balke and Joachim Selke

Homework Assignment 3

Due to June 3, 2010
(34 points in total)

Remember: If you have any problems or questions regarding this assignment, please let us know. We are happy to help!

Note: For this assignment, please use the *stemmed* version of the Reuters collection, which is available for download on the lecture website.

Exercise 3.1 (Binary Independence Retrieval)

Answer the query “taxes reagan” using the binary independence retrieval model (you may estimate the term $\Pr(D_i = 1 \mid D \in R_q)$ by 0.9 as proposed by Croft and Harper). Compare the results to the ones generated by the vector space model (using TF–IDF and cosine similarity; see Exercise 2.4). Which model works better (in your opinion)? (8 points)

Exercise 3.2 (Latent Semantic Indexing)

a) LSI has been reported to work better if it is applied to a transformation of the term–document matrix (rather than to the term–document matrix itself).[1] Therefore, please take the filtered and stemmed Reuters matrix *TD* from Assignment 2 (this matrix is available for download on our website) and replace each entry $td_{i,j}$ by its corresponding log entropy

$$td'_{i,j} = \left(1 + \frac{\sum_{r=1}^{n} \frac{td_{i,r}}{f_i} \cdot \ln\left(\frac{td_{i,r}}{f_i}\right)}{\ln(n)}\right) \cdot \ln\left(td_{i,j} + 1\right),$$

where n is the number of documents in the collection and f_i is the total number of times term i occurs in the whole collection.

If you did this and saved the new matrix as `TDLSI`, the command `TDLSI(1:200, 1:200)` should return the following:

```
ans =
 (131,12)      0.2519
 (108,39)      0.5051
 (121,73)      0.5089
 (107,192)     0.5487
```

(10 points)

[1] Source: `http://en.wikipedia.org/wiki/Latent_semantic_indexing`.

Hint: The MATLAB commands `spdiags` (to rescale the rows of a matrix by multiplying it with a sparse diagonal matrix) and `spfuns` (to apply a function to each nonzero entry of a sparse matrix) might be helpful.

b) Perform LSI on the transformed term–document matrix you just created by computing its rank-100 approximation. Do it as shown in the lecture by creating two new matrices U'_{100} and V'_{100}. (3 points)

Hint: The MATLAB command `svds` will be helpful (be careful, the matrix `V` returned by MATLAB is the transpose of the matrix we called V in the lecture).

c) Take a look at the first five latent dimensions generated by LSI by inspecting which terms get the highest and lowest coordinates in each dimension. Try to assign a meaningful concept name to each dimension! (5 points)

d) Answer the query "taxes reagan" using LSI (on the matrices U'_{100} and V'_{100} using cosine similarity). Compare the results to the ones generated by the vector space model (on the term–document matrix using TF–IDF and cosine similarity; see Exercise 2.4). Which model works better (in your opinion)? (8 points)

Hint: The MATLAB command `pdist2` might be helpful.