Assignment 3

Information Retrieval and Text Mining 17/18

2017-12-01; to be submitted 2017-12-19

Roman Klinger, Florian Strohm

- **Deadline:** These assignment will be discussed on 2017-12-21. Submissions are accepted via Ilias until end of day 2017-12-19.
- **Groups:** Working in groups of up to three people is encouraged, up to four people is allowed. More people are not allowed. Copying results from one group to another is not allowed. Changing groups during the term is allowed.
- **Grading:** Passing the assignments is a requirement for participation in the exam in all modules IRTM can be part of. Altogether 80 points need to be reached. There are five assignments with 20 pen & paper points and 10 programming points each. That means, altogether, 150 points can be reached. Explain all solutions.
- **Submission:** First make a group in Ilias, then submit the PDF. Write all group members on each page of the PDF. Only submit *one* PDF file. If you are technically not able to make a group (it seems that happens on Ilias from time to time), do not submit a PDF multiple times by multiple people – only submit it once. Submission for the programming tasks should also be in the same PDF.

Task 1 (Cohen's Kappa), 6 point

Please choose two from the four queries:

- buy ebike stuttgart
- how to pass exam
- why do we celebrate christmas
- *a query of your choice*

Step 1: Please receive the top 20 results for these two queries from a search engine of your choice.

Step 2: Each member of the group should annotate all 40 query-document pairs individually. Do not discuss while annotating. (if you are alone in a group, annotate twice, once as soon as possible, and then again some days later without looking at the previous result)

Step 3: Calculate a pairwise Cohen's kappa (if you are two members in the group, the result is only one kappa value, if you are three members A,B,C, calculate kappa for A-B, A-C, B-C, etc.)

Step 4: Discuss: In which cases did you annotate differently? Is the inter-annotator agreement high or low? Why? Can you do a qualitativ analysis of the differences?

Please explain each step including the results. Step 2 could be shown as a table of URLs with all annotations.

Task 2 (Language Models, Jelinek-Mercer smoothing), 3 points

Given the documents:

- d_1: Xerox reports a profit but revenue is down
- d_2: Lucene narrows quarter loss but revenue decreases further

Given the query:

- q: revenue down

Use Jelinek-Mercer smoothing with $\lambda = 0.3$. Which document is more likely for the given query? What would the result be for other values of λ? (Explain!)

Task 3 (Ranked Evaluation), 4 points

The ranked result list for a query given to a user is:

- q_1: 126, 8, 9, 1, 34, 31, 40, 63

The correct result (provided by a human judge) is

- q_1: 1, 33, 40, 63, 126

Please draw the interpolated recall-precision graph.

Task 4 (Probabilistic Ranking), 4 points

Explain, in your own words, what the motivation for probabilistic ranking is (in comparison to TF-IDF vector spaces with cosine similarity). How would you implement language models for ranking efficiently? Can you use ideas of the inverted index?

Task 5 (IDF and Stop word lists), 3 points

When you use IDF (in combination with TF): What are advantages and disadvantages compared to using stop word lists?

Programming Task 3 (10 points)

Choose between one of the following subtasks:

Subtask 1: Ranking in your existing retrieval system

Add a ranking method to your implementation. Describe your approach in plain text and submit the code added for ranking (not the whole system's code any more, but enough to understand how it works. What parameters does your ranking method have (if any)? How do these parameters influence the results?

Provide examples for (at least) three queries and show the results (also providing an intuition of the influence of parameters).

Subtask 2: Cosine Similarity based on TF·IDF

Implement a method which takes two texts as input (from the data provided in assignment 1) and outputs a similarity score based on cosine with TF-IDF values. Pick (at least) three texts and rank all other texts decreasing by similarity. List the texts of the top 100 results. Interpret your findings (do not use an existing library to calculate TF-IDF vectors or cosine).