Assignment 3

Information Retrieval and Text Mining 23/24

Publication: 2023-11-28
Submission Deadline: 2023-12-12
Discussion Session: 2023-12-14

Yarik Menchaca Resendiz, Roman Klinger
irtm-teachers@ims.uni-stuttgart.de

- **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 (or from elsewhere) 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.
- **Submission:** First make a group in Ilias, then submit the PDF. Write all group members on the first 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.
- **Make it understandable:** Do the best you can such that we can understand what you mean. Explain your solutions, comment your code. Print the code in a readable format, write your solutions in a way we can read them.
- **Handwriting:** We typically receive some submissions which are handwritten. That is fine, but if you submit handwritten solutions, make sure that they are well organized, easy to read and to understand, and that there is not doubt about the interpretation of letters. If you think that this might be hard, please typeset the solutions with a computer. We might reduce points if it's really tough for us and cannot read your submission properly.
- **Language:** Please submit your solutions in English. We have limited capacity of correcting German submissions.

Task 1 (TF·IDF, Cosine Similarity), 6 points

You are given the following documents:

- d_1: street bicycles ride on street
- d_2: horses ride on beach
- d_3: bikes ride on street

and the query

- q: "bikes beaches".

Subtask 1 (3 points)

Calculate the tf·idf weights for each term in the documents and write down the corresponding document vector.

Subtask 2 (3 points)

Calculate the cosine similarity between each document and the query and provide the ranking of all documents.

Task 2 (Annotation and Agreement), 7 point

We want to practice the creation of a benchmark collection to evaluate information retrieval systems. Please come up with an information need for which you then collect documents which you then annotate. Do not make the information need too simple – the annotation task won't be fun.

Step 0: Formulate the information need you planned to address in a short paragraph. Then, state the query that you want to use which approximates this information need.

Step 1: Please retrieve the top 20 results for the query from a search engine of your choice.

Step 2: Each member of the group should annotate all 20 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 qualitative analysis of the differences to get an understanding why you did annotate differently?

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

Task 3 (Ranked Evaluation), 4 points

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

- q_1: (128, 7, 9, 3, 35, 32, 41, 64)

The correct result (provided by a human judge) is

- q_1: {3, 34, 41, 64, 128}

Please draw the interpolated recall–precision graph.

Task 4 (Term-at-a-time, Document-at-a-time), 3 points

We discussed in the first lecture of this class how we can efficiently create an inverted index for document-at-a-time processing. We did not discuss in detail how to create an index for term-at-a-time processing.

Please develop an efficient strategy to do so and explain it. Further, explain the differences between the two creation procedures. Which of the two indexing procedures is more efficiently done?

Programming Task (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? How do these parameters influence the results?

Provide examples for (at least) three queries and show the results. Make sure that the examples provide some intuition for the influence of the 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). When you include a parameter to weight tf to idf differently, does that affect the result? Can you show this difference based on example queries and the respective results?