data analytics lab ETH zürich

# assignment 2 - retrieval system

## *introduction*

In this assignment, you will write your own Scala search engine that, from a large collection of documents, returns a list of the most relevant documents for a number of queries. For fast retrieval of candidate documents, we ask you to build an inverted index (see Lecture VIII for details).

All material is available here:

- qrels - a set of relevance judgements for pairs of queries and documents
- 40 training topics (or queries)
- 100k newswire documents

## *libraries*

As in assignment 1, both Breeze and TinyIR (see updated version below) libraries are accepted. No other external libraries (e.g. for search engines) are allowed. If you are unsure whether certain libraries are permissible, please contact us.

## *task*

- Scan the document collection no more than twice.
- Build an inverted index from words to document IDs.
- Implement two scoring systems for each (document, query) pair: one term-based model, one language model
- Retrieve the top 100 documents for each given query and for each scoring model.
- Based on the 40 training topics and qrels, tune your system and compute P, R, F1, AP per query as well as MAP for the entire system (averaged per all queries). See metrics description slides.

## *deliverables*

We ask you to send us one zip archive named exactly ir-practical-2016-2-[groupid].zip containing one folder and 4 files:

- a source code folder containing the entire source code of your system (without the libraries), a build file and a README containing instructions on how to run your code. In particular, the README should mention how to reproduce your predictions for the test set.
- report-[groupid].pdf : a short report (max. 2 pages) that describes your system, your retrieval models as well as your training set performance in terms of MAP. You should also report average (per query) running times when using an inverted index versus when passing the document collection sequentially for each query.
- ranking-[t|l]-[groupid].run : two ranking files (one for the term-based model, one for the language model) of the top 100 most relevant documents for each test query (to be distributed later).

For each of these files, replace [groupid] with your individual group id (as listed in the Google doc).

Please name your files exactly as stated above!!

## *grading*

We will grade by the following scheme (weight in brackets):

- Correctness of models/code (50%)
- Report (30%)
- Performance on test set based on how well your system performs in comparison to everyone else's in terms of system-wide MAP (10%)
- Code readability (10%)
- Bonus: Additional methods that improve your results and are not mentioned in the lecture or tutorial sessions.

## *submission format*

Please submit all three deliverables via email to any of the TA's by Tue 13th Dec 2016, 11:59 PM. If you want me to find your submission and notify you upon receipt, please use the following subject line: ir-practical-2016-2-[groupid]. Since some mail servers refuse sending/receiving large attachments (the typical threshold being in the 15MB area), please make sure in advance that you do not include any unreasonably large amounts of data to ensure receipt of your submission.

## *code*

## *faq*

**Q** *Where exactly do I find the queries ?*

**A** In the topics file, each query can be found under the <title> tag. You are not allowed to use any other information from this file, like the query domain, description, summary, narrative or concepts. These are here just to help you understand the training data and debug your code.

**Q** *I get OutOfMemoryError exceptions.*

**A** You can increase the heap size. But before, think for a second if there is a good reason why the standard Scala heap space (often 256MB) is not sufficient. To increase it, add something like -Xmx2g to the scala command line or in Eclipse at Run→Run Configurations → "your filename" → Arguments → VM arguments.

**Q** *If there are more than 100 positive qrels, I cannot possibly obtain an AP of 1.*

**A** For these cases, it is ok to bound the AP denominator as MIN((TP+FN),100).

**Q** *My inverted index doesn't fit in memory.*

**A** An alternative solution is to keep it on the disk in a database, e.g. using LevelDB.