Exercise 1 - Basic Search System

188.412 Information Retrieval 2015S

March 16, 2015

Abstract

At the end of this exercise, you should have an understanding of the practical issues of creating an inverted index for information retrieval, as well as basic scoring. The exercise can be done in pairs, but each of the members has to be able to answer questions about the implementation provided.

Task

Your task, should you choose to accept it, is to create a basic ranking engine and be able to write and talk about it. Your performance will be graded as follows:

Functionality (50%)

Index

Build two inverted indexes from the document collection 20 Newsgroups Subset as provided on TUWEL (*20_newsgroups_subset.zip*). One is a simple bag-of-words, the other is a term bi-gram index. You may use any format for storing the index.

Vocabulary

Normalize the vocabulary by applying any combination of the techniques described in Chapter 2 of the *Introduciton to Information Retrieval* book (case folding, removing stopwords, stemming). These options should be exposed as parameters in the index creation phase.

Search

Implement a basic search functionality and provide it as a command line interface (CLI) — no GUI is required. The CLI allows the user to enter provide search parameters and a file with a topic to search for. The system then returns a list of documents ranked by a similarity function of your choice and based on a variant of term frequency ranking. Different components (e.g. the use of the bag-of-words or the bi-gram index, scoring method) need to be exposed as parameters in the command line.

The search engine must take as a parameter a topic file. You will be provided with a set of 20 topics to search for. These topics are in the same format as

the collection itself, so it should be easy to parse it using the same parser you created for the indexing part.

As a recommendation, also regarding the future exercises, try to implement topic processing and the actual search as two separate tasks.

Your search should be (moderately) efficient. It is e.g. not ok to perform your collection indexing only at search time, and also loading the inverted index should be reasonably in time, or at least not happen before every search.

For each topic of the 20 provided, the result should be a ranked list of up to 100 documents, where each line is in the following format:

topic Q0 document-id rank score run-name

where

topic is "topic#" where # is a number between 1 and 20;

document-id is an identifier for the document, consisting of the folder name in the newsgroups archive and the file name (e.g. misc.forsale/76050)

rank is an integer indicating the rank of the object in the sorted list (normally, this should be ascending from the first line to the last line)

score the similarity score you calculated (normally, this should be descending from the first line to the last line)

run-name a name you give your experiment (should be the same for the same configuration of index-scoring method, across all 20 topics tested)

Here is a short example of a potential output for a topic:

```
topic1 Q0 misc.forsale\76442 1 2.813525 group5-experiment1
topic1 Q0 misc.forsale\76056 2 1.0114759 group5-experiment1
topic1 Q0 rec.autos\102958 3 0.58848727 group5-experiment1
```

Java-based: You must use Java for your solution. You are allowed to use standard Java API functionality for index and search. For the vocabulary part, you are allowed to use external algorithms (e.g. for stemming). Everything you use, you must be able to explain.

Report (30%)

Describing the prototype in a report. Describe how the index is generated and stored, how the vocabulary is processed and how the search works. The report must explain how to run the prototype. Maximum size: 2 pages or 1000 words.

Hand-in Presentation (20%)

- Prototypes are presented to the coordinator in one of the labs.
- Final deadline is April 14th at 8 am (but you may hand-in earlier). For that you must upload a zipped file to TUWEL (report, source code, and corresponding executable jar file incl. all dependencies).
- Your submission has to be self-contained.

- You must book a time on TUWEL for the presentation.
- You present on your own notebook to the coordinator in the seminar room.
- Seminar rooms for hand-in times are on TISS

Note: You must have 33% of this exercise in order to pass the course.