The Tenth International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2018), May 20 - 24, 2018 – Nice/France

Tutorial: How to build a Search-Engine with Common Unix-Tools
Andreas Schmidt

Exercise II

.

Download (if not already done in exercise 1) the dataset BBCSport from http://mlg.ucd.ie/files/datasets/bbcsport-fulltext.zip und unzip the content in a directory. In this exercise an inverted index should be created. To be able to rank the documents, the number of times a word appears in the document should be saved for each document. This corresponds to the structure as presented on slide 42.

1. Create an intermediate file *doc-term.txt*, which contains two columns. The first column is the filename and the second column is a lowercase word or a numeric expression like 2018, from the file. Sort the entries in the file, according the filename, and as a second criteria, the word. (use `grep` with –o option, `tr, ort`).

2. Count the number of times, each word (or numeric expression) appears in each file. This can be done with the command `uniq`. Write the result in the format below, using `awk's` printf-capability.
   `<filename>:<word>:<number-of-occurrence>`
   Write the result in file `doc-term-count.idx`

3. Assign each document in the folder "tennis" a unique ID of the form %03d and write the result in a file called *doc-id.mapping*. The result should look like:
   `<doc-id>:<document-filename>`
   Hint: use `ls`, `cat` with –n option and `awk` for formatting purpose.

   Then use this file to replace the filenames with the unique document identifiers (use `join`). The target format should look like this:
   `<doc-id>:<word>:<number-of-occurrence>.`
   The result should be written in a file called `doc_id-term-count.idx.`

   Alternatively you can use the fact, that each file-path in file `doc-term-count.idx`, already has a well formatted unique numeric part (use `ls`, `sed`).

4. Write the inverted index: For each term in the file `doc_id-term-count.idx` write a file with the name *invIndex/<term>.idx*. An entry in such a file should have the following format:

   ```
   <doc-id>:<number-of-occurrence>
   ```

   Use `awk` to perform this task. Inside the awk script use >> file redirection.

   Check: The index-file *invIndex/bogdanovic.idx* should have the following content:

   ```
   $ cat ./invIndex/bogdanovic.idx
   011:3
   017:3
   029:5
   080:1
   099:1
   ```

5. Formulate a query: Get all documents, containing the terms “anna” and “martinez”. As ranking criteria use the sum of occurrences of the search terms. Show the document path of the matching documents by decreasing relevance. (use `join`, `awk`, `sort`). The result of the query should look like this:

   ```
   d:/data/bbcsport/tennis/084.txt        4
   d:/data/bbcsport/tennis/056.txt        2
   ```

Hint: The solutions for this exercise can be found here:
http://www.smiffy.de/dbkda-2018/IR-exercise-2-solution.pdf