Comp 479/6791 **Project 1** **Fall 2019**

Objectives: Implement SPIMI. Implement rudimentary information retrieval using your SPIMI indexer. Test and analyze your system, discuss how your design decisions influence the results.

Due Date: 24.10.2019

Description:

1. implement the SPIMI algorithm with disk block merging

   SPIMI is an algorithm and has to be implemented as described in the book, no reimagining or shortcuts. To simulate saving and reading to disk, make your program create a directory called DISK. You will need to demonstrate two things to the markers: (a) Assuming a disk block size of 500 Reuter's articles, make it possible to inspect the blocks your SPIMI creates (call simulated disk blocks BLOCK1 ... BLOCK7 ...) (b) make it possible to inspect the merged index vocabulary (simulate distributing your final index over two blocks). Make sure you can demonstrate merging from n blocks.

2. compile an inverted index for Reuters21578 without using any compression techniques, still distributing your index over two blocks and accessing it from there (no hash!)

3. implement the lossy dictionary compression techniques of Table 5.1 in the textbook and compile a similar table for Reuters-21578. Are the changes similar? Discuss your findings. (Note that stemming is not required here, if you run out of time before you get the Porter stemmer to work, that is ok for this assignment, the remaining table is fine.)

4. implement a simple scheme to retrieve matching documents for a few queries (not only single word queries). Techniques from Chapters 1-3 are suitable. Show the queries you used and discuss your findings.

Deliverables:

1. individual project, remember to submit the expectations of originality page on Moodle

2. well documented code

3. well documented sample runs of the queries posted on 22.10.

4. any additional testing or aborted design ideas that show off particular aspects of your project

5. a project report that summarizes your approach, illustrates your design and discusses what you have learned from the project. Note that a summary and commentary on your sample runs has to be included in the report

6. a three slide overview of your system for the lab demo

Test queries:

1. design three test queries:

   (a) a single keyword query,

   (b) a multiple keywords query returning documents containing all the keywords (AND),

   (c) a multiple keywords query returning documents containing at least one keyword (OR), where documents are ordered by how many keywords they contain)

   .

2. run your three test queries to showcase your code and comment on the results in your report

3. exchange one test query each with three different students in the class, run their queries
4. compare your results during lab time
5. report the experiment in your project report

Submissions: submit on Moodle. All code has to run on the lab equipment. You have to demo your project during lab time on 29./30.10. or as agreed with your lab instructor.

Marks:

Preprocessing (text extraction, tokenization)	1pt	Attr1
Spimi implementation	2pts	Attr1
Inverted index	1pt	Attr1
Block handling	1pt	Attr5
Dictionary compression	2pts	Attr1
single keyword query	1pt	Attr1
multiple keyword AND query	1pt	Attr1
multiple keyword OR query	1pt	Attr1

docID hint

Use the NEWID values from the Reuters corpus to make your retrieval comparable.