Recuperação de Informação / Information Retrieval

2018/2019 MIECT/MEI, DETI, UA

Assignment 2

Submission deadline: **12 November 2018**

For this assignment, you will create a weighted (tf-idf) indexer and ranked retrieval system. Use the same corpus as in assignment 1.

[REQUIRED, 16 points]

1. Extend your indexer to apply tf-idf weights to terms.
   Reuse the tokenizer that incorporates stemming and stopword filtering from Assignment 1.
   Save the resulting index to a file using the following format (one term per line):
   term;doc_id:term_weight;doc_id:term_weight;…

2. Create a class that implements a ranked retrieval method using the index created in 1.
   You should consider the memory availability when loading the index (or segments) from disk.

[EXTRA, 4 points]

3. Positional index and phrase search.
   a. Extend your indexer to store term positions.
      The resulting index should be saved with the following format (one term per line):
      term;doc_id:term_weight:pos1,pos2,pos3,…;doc_id:term_weight:pos1,pos2,pos3,…

   b. Extend your retrieval method to allow phrase and proximity search.

Note:

Your assignment will be evaluated in terms of: modelling, class diagram, code structure, organization and readability, correct use of data structures, submitted results, and report. See suggestions and submission instructions below.

Suggestions:

- Write **modular** code
- Favour **efficient** data structures
- Add **comments** to your code
- Follow the **submission instructions**

Submission instructions:

- To manage your project please use **Maven**
- At each submission, include a small **Report** including:
    - Your project's **class diagram**
    - A description of each class and main methods, identifying where these are called
    - A block diagram and a high-level (but sufficiently detailed) description of the overall processing pipeline (data flow diagram)
    - Complete instructions on how to run your code, including any parameters that need to be changed
    - A list of any external libraries that are needed to run the code
    - Efficiency measures: total indexing time; maximum amount of memory used during indexing; total index size on disk
    - A short commentary/assessment of your own work, describing features or implementation decisions that you consider the most relevant/positive (or otherwise)
- Make sure you **include your name and student number** in the code and in the report.
- Make sure all your programs compile and run correctly.
- Submit your assignment by the due date using Moodle.