Recuperação de Informação / Information Retrieval

2016/2017 MIECT/MEI, DETI, UA

Assignment

For this assignment you will create an information retrieval engine. This will consist of various modules, namely corpus reader, document processor, tokenizer, indexer, and searcher. The work should be handed in according to the schedule defined below.

Submission 1: **05/10/2016**

Modelling: classes and main methods definition.

a) Keep in mind modularity and flexibility.

b) Describe your classes, main methods, and data flow in the report.

Submission 2: **26/10/2016**

Implement a simple corpus reader, tokenizer, and Boolean indexer.

a) Develop your own tokenizer from scratch. Integrate the Porter stemmer (http://snowball.tartarus.org/) and a stopword filter in your code.

b) Index a small corpus (to be defined later) and submit a text file with the resulting index, following the scheme: term,document frequency,list of documents.

Submission 3: **23/11/2016**

Implement an indexer based on the vector-space model, using the tf-idf weighting scheme and *lnc.ltc* strategy, as described in the slides.

a) Write your index to disk so that the searcher module can efficiently load it.

b) Index the corpus *(to defined later on)*.

Submission 4: **21/12/2016**

Implement a ranked retrieval method.

a) Load the index from disk.

Extra

i. Allow phrase search (e.g. “fishing quota”) and proximity searches (e.g. “fishing quota”~10) in your indexing/retrieval methods.

ii. Support multiple fields and a combination of distinct fields in the queries (e.g. “world cup” category:Sport).

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** (preferably) or Netbeans
- 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.