Recuperação de Informação / Information Retrieval
2017/2018 MIECT/MEI, DETI, UA

Assignment 4

Submission deadline: **20 December 2017**

For this assignment, you will apply and evaluate methods for relevance feedback and thesaurus based query expansion. Use the same corpus as in previous assignments.

1. Implement and evaluate the Rocchio relevance feedback method.
    a. Use 'explicit' relevance feedback. For this, consider the real relevance (*gold standard*) of the first 10 documents in your retrieved results as user feedback. Suggestion: You may assign different feedback weights depending on the level of relevance of each document (1 to 4).
    b. Use 'implicit' feedback. For this, consider the first 10 documents in your retrieved results as positive feedback.
    c. Calculate and compare the average (across all queries) NDCG obtained with these options and with your baseline implementation from assignment 3.

Note: you need to keep a document cache (in memory or disk) to know which terms occur in each document.

2. Implement query expansion using an automatically generated word association thesaurus.
    a. Generate a thesaurus of word associations from the full Cranfield collection. For this, generate word embeddings[1] from the collection, and use the similarity between words to expand the query (i.e. add similar words to the query).

       Use the word2vec implementation from https://deeplearning4j.org/word2vec or https://radimrehurek.com/gensim/models/word2vec.html

       Once you have calculated the word embedding vectors, you can obtain the most similar words to each word in the query. In deeplearning4j, for example, you can do:

       ```
       most_similar = vec.wordsNearest("aeroelastic", 3)
       ```

    b. Calculate and compare the average (across all queries) NDCG obtained with this approach and compare to previous approaches.

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.

[1] Word embeddings are vector representations of words. https://deeplearning4j.org/word2vec

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.