ASSIGNMENT DESCRIPTION FOR THE COURSE *INFORMATION RETRIEVAL*

JAN HIDDERS AND LARS VAN HOLSBEEKE

1. Introduction

The assignment consists of building a small information retrieval system that allows the indexing and retrieval of documents. Several aspects need to be built into the system, for which there can be one or more options. Which option you have to do depends on the description of the assignment for your group. You can use any tools or libraries that are available, as long as you can explain why and how you used them to achieve the desired effect or implement the requested technique.

2. Overview of aspects

Aspect 1: Querying: Has the following options:

(a) Implement full boolean querying supporting AND, OR and NOT.

(b) Implement positional indexes and proximity queries with "within n words" or "near" operator.

Aspect 2: Normalization: Has the following options:

(a) Include the use of stemming to improve recall, and show the improvement.

(b) Include the use of soundex to deal with misspellings and improve recall. Demonstrate the improvement.

Aspect 3 Wild-cards: Has the following options:

(a) Build and use a permuterm index over the dictionary for * wild-cards.

(b) Build and use B-tree over the dictionary to deal with * wild-cards.

Aspect 4 Ranking: Has the following options:

(a) Implement cosine ranking, using a heap to get the top k ranked documents.

(b) Implement ranking based on Euclidian distance between normalized vectors, using a heap to get the top k ranked documents.

Aspect 5 Ranking optimization: Has the following options:

(a) Implement cosine ranking with high-idf query terms only for optimization of top-k ranking. Demonstrate the improvement.

(b) Implement cosine ranking with presence of multiple query terms for optimization of top-k ranking. Demonstrate the improvement

Aspect 6 User feed-back, Query expansion and XML Retrieval: Has the following options:

(a) Implement the *Rocchio* feedback method.

(b) Implement query expansion based on a thesaurus.

(c) Implement and demonstrate the vector space model for XML information retrieval.

3. Assignment of aspects to groups

Group	Asp. 1	Asp. 2	Asp. 3	Asp. 4	Asp. 5	Asp. 6
1	a	a	a	a	b	c
2	b	b	a	b	a	a
3	a	b	b	a	a	c
4	b	a	b	b	b	b

4. Presentation

The built system must be presented in front of fellow students, the assistant and the professor for evaluation. The presentation will consist of a short introduction on how the system was built (5 minutes), and a short demo of the implemented features (15 minutes). This will be followed by a brief interrogation and discussion with the professor and assistant. All students are expected to attend the presentations, and will also be asked to give feed-back (in public) and grade (anonymously) the work of the other groups. The grades of peer groups will determine 20% of the grade for the assignment.

5. Report

At least two working days before the presentation, a short report (maximum 15 pages) describing your work should be handed in electronically. The report should contain the following:

- Cover page with (1) group number, (2) name and student number of group members
- Overview of the executed aspects and the implemented option.
- Overview of used technologies, (open-source) libraries, APIs, literature and ideas from similar systems.
- Overview of how the workload was spread among the members of the group.
- For each implemented option:
  - Achieved functionality. Possibly including some examples.
  - Global description of implementation: used software / components, global architecture, design decisions.
  - Used data sets.
  - Evaluation and discussion of the final result.

6. Resources

- Potential datasets:
  - *Wikipedia* articles and articles from the *Communication of the ACM* journal: `http://www.search-engines-book.com/collections//`
  - The INEX 2009 collection, containing XML documents representing annotated Wikipedia pages: `http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/software/inex/`
- Software:
  - Lucene: *a high-performance, full-featured text search engine library written entirely in Java.* `http://lucene.apache.org/core/`

- PyLucene (Python version of Lucene):
  `http://lucene.apache.org/pylucene/index.html`
- Solr: *a standalone enterprise search server with a REST-like API, based upon lucene.* `http://lucene.apache.org/solr/`