Information Retrieval: Assignment 1

Prof. Toon Calders, Ewoenam Tokpo
{toon.calders, ewoenamkwaku.tokpo}@uantwerpen.be

Deadline: 08/11/2021

This project is to be executed in groups of 2 students. For further inquiries about the project, please email ewoenamkwaku.tokpo@uantwerpen.be

1 Introduction

The goal of this assignment is to use an existing library, Lucene, to develop a document search and retrieval system.

At the end of the project, you should understand some of the fundamental concepts of information retrieval such as indexing, query processing, search, and scoring, and how such concepts can be applied in real-world applications. It is expected that you will understand the operational structure and details of Lucene, and importantly, you should be able to implement a simple information retrieval system with Lucene that can index, search and retrieve text documents.

2 Lucene

The Lucene library is available in Java [1] or Python PyLucene [2]. We strongly recommend the use of the Java version of Lucene (version 7.3) for this project; Lucene in java is far more straightforward than using PyLucene of which the installation often goes wrong. See installation details of Lucene here.

3 Dataset

For this project, you will be working with a specific set of documents. The dataset can be directly downloaded from large dataset. A small dataset is available for test purposes during development.

The dataset is an extract from the much larger Question Answering Dataset from MS Marco. Each file in the folder corresponds to a document.

Three sets of queries are provided; two to test your system during development and one for the final output:

1. A small test set of queries that can be used with the ***small dataset***. The corresponding result list can be used to evaluate your results. You can use this for a quick test of your implementation. You do not have to submit the results from this query list.

[1] https://lucene.apache.org/
[2] https://lucene.apache.org/pylucene/

2. A large test set of queries that can be used with the ***large dataset***. The corresponding result list can be used to evaluate your results. You can use this for a more thorough test of your implementation. You do not have to submit the results from this query list.

3. The main set of queries that must be used for the final evaluation. The result list from these queries must be submitted alongside your implementation (see Section Assignment: 1(c)). Your implementation will be evaluated based on these query results.

4 Assignment

The objective of this project is to get familiar with Lucene and to study the feasibility of using Lucene to add retrieval capabilities to large text datasets. The requirements are as follows:

1. Implementation of a document retrieval system with Lucene. The implementation should cover the following aspects:

   (a) Retrieval system:
   - Document analysis and indexing
   - Query processing
   - Document search and retrieval

   (b) Using the ***small dataset*** together with the query list, experiment with various query types, search, and scoring techniques. Based on your observations, find a suitable combination for the provided dataset. You can use the query lists and the result lists provided in the previous section to improve the performance of your system.

   (c) Using the ***large dataset*** and the ***main set of queries*** (see previous section), for each query, return the list of top-10 most relevant documents (in descending order of relevance). Save this output file as ***result.csv***. The expected format for your submission is a csv file consisting of two columns named *Query_number* and *Document_number*. Each row of the file should correspond to a query_number-document_number pair, starting from the first query; see this file for an example of the format. It is possible some queries will have less than 10 returned documents; in that case, the number of rows corresponding to that query number will be less than 10. Include *result.csv* in the code repository.

2. Write a report of approximately 8 pages in length. The report should cover the following:

   (a) Background on Lucene. Some guideline questions are below:
   - What is Lucene?
   - What are the main functionalities of Lucene?
     - Which types of indices are available? How does Lucene store the index?
     - Discuss the various types of queries.
     - How is search and scoring done in Lucene? The different score models, etc.

   (b) Details of your implementation:
   - Discuss the major aspects of the implementation of your retrieval system.
   - Discuss the results/observations from 1(b).

5 Deliverables

1. The code of your project. **Do not** include bulky software libraries or large datasets in emails. The preferred way to share code is via a link to a GitHub repository by including the link in your report. Mae sure the repository is accessible with the link. Code should be readable; that is: include comments, use appropriate naming conventions for functions and variables.

2. The report in pdf format, to be submitted via BlackBoard. **Do not** submit zip-files, word documents, etc., only the submission of a single pdf file will be accepted.

A Note on Plagiarism

There is absolutely nothing wrong with using existing materials, you will even be commended for not reinventing the wheel, as long as you are not violating the copyright of other authors. Nevertheless, it is expected from you to clearly indicate whenever you used material that was not created by yourself. Clearly indicate in your submissions which parts constitute original work, which parts are taken from other works, and which parts were adapted from external sources. These sources have to be properly acknowledged in all your submissions. Concretely, this means at least the following guidelines are observed:

- Papers, books, webpages, blogs, etc. that were inspected while making the assignment will be referenced in a separate section "References". Citations to these materials are included in the text where appropriate.

- Text fragments exceeding one sentence that are copied from other sources are clearly marked as such. You could for instance include quoted text, definitions, etc. in italics, followed by a reference. An example of how to do this: Bela Gip (2014) defines plagiarism as *"The use of ideas, concepts, words, or structures without appropriately acknowledging the source to benefit in a setting where originality is expected"*

  References: (at the end of the document) Gipp, Bela. Citation-based plagiarism detection. Springer Vieweg, Wiesbaden, 2014. 57-88.

- When using code from other sources, indicate so in the report, and in the source code. This could for instance be done by adding a comment with a reference to the source of the function for each function that was copied from another source. It is recommended to include a separate folder "sources" in your GitHub repository with the original files from other authors that you used. Include source in the message of your commits.