Information Retrieval: Assignment 1
Document Search and Ranked Retrieval (self-implementation)

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

Deadline: 27/10/2024

This project is to be executed in groups of 2 to 3 students. For further inquiries about the project, please email ewoenamkwaku.tokpo@uantwerpen.be

1 Introduction

The goal of this assignment is to ***self-implement***, a functional document Search and Retrieval system. For this project, the use of external search and retrieval libraries like Lucene, Solr, or Elastic Search is ***prohibited***. You are, however, allowed to use generic libraries and packages such as ScikitLearn, Pandas, etc. The project can also be implemented with any programming language of choice.

2 Assignment

The objective of this project is to get familiar with fundamental search and retieval techniques in Information Retrieval (IR). At the end of the project, you should understand some fundamental concepts of information retrieval such as indexing, query processing, search, and scoring, as well as how such concepts can be implemented.

2.1 Dataset

For this project, you will work with a given 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 dev set of queries that can be used with the ***small dataset***. The corresponding result list (unranked) 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.

2. A large dev set of queries that can be used with the ***large dataset***. The corresponding result list (unranked) can be used to evaluate your results. This dataset will be used for the evaluation section of your report (see Project Specifications).

3. The test set of queries that must be used for the final ***Test output***. The output file for these queries must be submitted alongside your implementation (see Project Specifications).

2.2 Project Specifications

The goal of this project is to implement a document retrieval system and to apply ***search*** and ***ranked retrieval techniques*** using both small and large datasets. The following steps outline the requirements for the project.

1. **Document search and retrieval system implementation:** Build a document retrieval system that handles document analysis, indexing, query processing, document search, and retrieval. Ensure the system covers the following fundamental aspects:
   - **Document Analysis and Indexing**: Implement a simple inverted-indexing mechanism.
   - **Query Processing**: Develop a query processing system that takes a query and prepares it for matching against indexed documents.
   - **Document Search and Retrieval**: Implement a *vector space model (VSM)* that retrieves relevant documents based on an input query. The model should generate scores that rank the retrieved documents by relevance.

2. **Extensions (for additional scores)**: You can implement one or more of the following extensions for extra scores. Beside these extensions, you are also free to implement other extensions you desire.
   (a) **Positional indexing for phrase queries**: Implement a positional indexing mechanism to enable your retrieval system handle phrase queries.
   (b) **Advanced index construction and management:** Implement advanced indexing features like single-pass in-memory indexing (SPIMI), distributed indexing, or dynamic indexing.
   (c) **Preprocessing schemes**: Implement extra text processing aside from tokenization to enhance search and retrieval. E.g. stop word removal, stemming/lemmatization, case formatting, unknown character removal, word normalization. You can use this blog post as a reference.

3. **Report**: Prepare a short report (about 3 to 4 pages in total) covering:
   (a) Conceptual details of the key components of your system.
   (b) Implementation details of the key components of your system.
   (c) Evaluation: Use the ***large dev dataset*** and its corresponding ***result list*** to evaluate the performance of your implementation.
      i. Compute Mean Average Precision at k (MAP@K): The average precision of search results across all queries; for $k = 3, 10$.
      ii. Compute Mean Average Recall at k(MAR@K): The average recall of search results across all queries; for $k = 3, 10$.

(d) Analysis and discussion of the evaluation results.

(e) Include a GitHub link of the implementation code in the report.

4. **Test output**: Test your approach on the ***large dataset*** with the provided ***test set of queries***.

   (a) Return the top 10 most relevant documents (ranked in descending order of relevance) for each query.

   (b) Save this output as ***result.csv*** in the format *(Query_number, doc_number)* provided in the example *result list* files. Include this file in the GitHub repository.

   The report should be submitted as a PDF file via Blackboard. Do not submit zip files, Word documents, or bulky attachments. All code should be placed in a 'src' directory, and result files in a 'results' directory in the GitHub repository.

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.