Developing an Information Retrieval System with Document Ranking

About

This project aims to augment the Information Retrieval (IR) system developed in the previous assignments by incorporating different Document Ranking strategies. You should use the Cranfield collection as the dataset. You can find that in the original format here or in the TREC XML format with binary tagging here.

Project Overview

In this project, you will implement three different approaches for document ranking, including the vector space model, the binary independence model, and the language model. Then, you need to compare these ranking models resorting to the evaluation criteria in Lecture 7. Key components and functionalities are as follows:

- **Document Preprocessing:** Your project will begin by reading and preprocessing a collection of text documents – for each document you only need to retain the text with TITLE and TEXT tags. The dataset also contains queries and relevant documents to each query which will be useful in the evaluation phase.
- **Document Ranking – Space Vector Model:** You will implement a function for document ranking utilizing the tf-idf weighting approach. The function will take as input a query text and an integer indicating the number of top documents to be retrieved. To this end, you might need to store additional information about the "term frequency" in each document and the "document frequency" of each term. A good representation and sufficient data storage facilitate you in implementing other models too. You can choose between different tf-idf variants and need to discuss why your chosen approach is preferred.
- **Max-Heap – Space Vector Model (Optional):** Reduce the sorting complexity in the page ranking function to $O(n)$ for top-k elements ($k << n$), by implementing a max-heap.
- **Document Ranking – Probabilistic Model:** You will implement a function for document ranking utilizing the Okapi BM25 basic weighting approach. The function will take as input a query text and an integer indicating the number of top documents to be retrieved. You do not need to fine-tune parameters b, k_1, k_3. Nevertheless, you have to argue why the acquired parameters make sense.
- **Long queries – Probabilistic Model (Optional):** Handle long queries by having your function alternatively switch between Okapi BM25 approaches based on the query length.
- **Document Ranking – Language Model:** You will implement a function for document ranking utilizing the language model. The function will take as input a query text and an integer indicating the number of top documents to be retrieved. You can choose between Dirichlet smoothing or Jelinek-Mercer smoothing to avoid zeroes. You do not need to fine-tune parameters λ or α. Albeit, you need to discuss why your chosen methods and parameters are preferred.
- **Compering Document Ranking Model:** You will need to compare these three approaches for document ranking based on an evaluation you learned in Lecture 7. You are provided with a set of queries and their relevant documents, and you need to utilize these queries during your evaluation. You are suggested to use 11-point interpolated average precision for at least a few queries; however, you might resort to other criteria or have a number of queries tested. You may also introduce your own evaluation approach if you are utilizing the Cranfield collection in its original format, although you may also refer to this to change the relevancies of the original format.

Input – Any Document Ranking Model

ranking_input = ["example content example", 2]

- **Query Text:** The user's input query.
- **Number of Top Documents:** An integer specifying the count of top-ranked documents.

Output

expected_results = [{"Id": Document#, "Score": S#}, {"Id": Document#, "Score": S#}]

Deliverables

- Three functions handling document ranking as described.
- A comparison of these models based on an evaluation function on the provided query-relevant document pairs.
- Comprehensive documentation describing the functions' architecture and components.
- A report summarizing key findings, challenges faced, and enhancements made to the IR system during the project's development (you may append this to your previous reports).

References

Document and reference the sources, libraries, and tools used in the project. You can refer to the NLTK library for text processing and explore relevant literature on Information Retrieval techniques.