UNIVERSITY OF SOUTHERN MAINE

Introduction to Information Retrieval, Fall 2024, Assignment 1

Instructor: Behrooz Mansouri

Due: September 26, 2024

In this assignment, you will build your first search engine, a boolean information retrieval system in Python. The goal is to find relevant answers to given questions about traveling.

Data files

- **Answers**: This JSON file will be used as the collection we search over. All the systems in our assignments will use this file as the collection, indexing the answers. Each answer has three properties: *ID* as a unique integer identifier, *Text* is the answer, and *Score* which shows the popularity of the answer in a community question answering (CQA) website. *Score* can be useful in case you want to incorporate popularity in your ranking model (maybe next assignments!).

- **Topics**: There are two JSON files, topics_1, and topics_2. These files contain the input search queries for your system. Each topic has four attributes: *ID* as a unique integer identifier, *Title* is the question title written by the CQA member, *Body* is the detailed question content, and *Tags* are the selected topics by the user. Your system should retrieve the top-100 answers for each topic. You should provide two results files: result_binary_1.tsv (for the topics in topic_1 file), result_binary_2.tsv (for the topics in topic_2 file). The TSV (tab separated value) files have no header and show contain the following columns in each row separated by Tab (TREC standard format): qID Q0 answerID rank score runName

where:

qID	is the query (topic) ID
Q0	is the literal Q0 (Just simply print Q0)
answerID	is the ID of an answer returned for qID
rank	(1-100) is the rank of this answer for this qID
score	is a system similarity score indicating of the quality of the answer to the query
runName	is the identifier for the system (any string)

Note: IR evaluation tool ignore the rank, and they evaluate the results based on the score. It is important that score values have a positive correlation with ranks. If two answers have the same score, the answerID is used for ranking them.

- **Qrel File:** The qrel_1.tsv file provides you with the relevance annotation for topic_1 topics. You can use this to improve your model based on evaluation results.

Designing Boolean Retrieval Model

We have explored Boolean retrieval model in our class (session 4). Now you will develop your boolean retrieval system. Note that each topic has a title, body, and tag, and you can use whatever data you want as the input for your model. However, in your analysis, you should discuss what you have studied and reason about your final model design. For example, one might use only the titles, considering body and tags as noise. In an alternative design, you might use title and body separately and average the system's scores (fusing two systems) as your final proposed system. The choice is yours, however, your system will be run and evaluated on topics_2 file, for which the evaluation is hidden from you. Part of your grade will be based on how effective your system is. Your system should also be efficient and provide retrieval results in a timely manner.

Evaluation Measures

Using the qrel_1 file, you should provide evaluation results for topics in topic_1 file. You will report the P@1, P@5, nDCG@5, MRR, MAP. For metrics other than nDCG, the relevant documents are those with scores 1 and 2 (in the qrel file). You can use any tool that you want for evaluation. However, this tool should be standard, and the results should be reproducible (Trec_eval and Ranx are recommended).

Report and Analysis

You should provide two results files for each system, and make sure they can be passed to Trec_eval and Ranx for evaluation. You do not need to provide an evaluation script.
For topic_1 file, provide a table showing the values for each metric (average over all the topics). Then provide a ski-jump plot based on P@5. For this plot, discuss what your model worked for, and what are the reasons for failures. You should provide at least one pair of successful and failed (topic, answer) in your discussion.

Notes for submission:

---

1. Python file/files (only .py is acceptable, not .ipynb) with codes for all retrieval. Codes should be well-structured with comments to run. You should have a README file that provides clear guidance on how to run your code and get the result files. Explicit file path should not be used in the code, and topic files should be passed as arguments to your code.
2. result_binary_1.tsv and result_binary_2.tsv files having your retrieval result files
3. A .pdf file having your table of results, ski-jump plot, and your analysis
4. Any assumptions made by students should be explicitly mentioned in the submitted

Note: Any submission not in the format explained above will be dropped, resulting in 0, without the possibility of regrading