Final Assignment

Spring 2022

Lecturers: Suzan Verberne, Wessel Kraaij

Information Retrieval

Analyze the impact of BM25's variants on an IR task

1. Introduction and learning goals

BM25 is perhaps the most well-known information retrieval (IR) model among the bag-of-words document retrieval approaches. Although learning-to-rank methods and neural ranking models are commonly employed today, they are most often exploited as part of a multi-stage re-ranking architecture, over candidate documents provided by a first-stage term-matching method employing traditional inverted indexes. This is frequently performed with BM25. As a result, studying the variants of this traditional scoring function, which is a central component of today's search applications, is relevant.

The goals of this assignment are (1) to acquire a more in-depth understanding of the effect of modification on scoring functions on retrieval effectiveness; (2) to learn to work with ElasticSearch; (3) to learn to implement a scoring function based on a mathematical definition.

You work in groups of 3.

2. Tasks

(For more details, see section 4)

1. Read the paper by Kamphuis et al. (2020): "Which BM25 do you mean? A large-scale reproducibility study of scoring variants." In: European Conference on Information Retrieval (pp. 28-34) (link).
2. Read the official website of the TREC Clinical Trials Track to get a holistic view of the IR task of this assignment (link).
3. Install version 7.9.1 of ElasticSearch (link).
4. Download the dataset (see section 3) and index it with ElasticSearch through the Python library.
5. Do a first-stage retrieval run for the Clinical Trials Track queries with the default ElasticSearch retrieval function.
6. Implement two BM25 variants ranking functions for the Clinical Trials Track task.
7. Write a report in which you motivate the choice for the BM25 variants, describe your implementation, compare and discuss the results (see requirements below).

3. Task: TREC Clinical Trial Track

Query by document (QBD) retrieval is a task in which the user enters a text document – instead of few keywords – as a query, and the IR engine finds relevant documents from a text corpus. Examples are patent prior art search and related scientific paper retrieval.

(1) The TREC 2021 Clinical Trials Track The TREC 2021 Clinical Trials Track is a QBD task focused at finding clinical trials that are eligible for a specific patient. Patient-related data is provided as a query in the form of an admission note. Each query contains conditions and observations that describe a patient, and queries are generally longer than those used in traditional ad-hoc retrieval tasks.

(2) Dataset There are 375,580 documents (link), 75 queries (link) and about 11k relevance judgements (link) in the Clinical Trial Track collection[1]. Please note that you need to first install 'ir_datasets' library in Python with 'pip install ir-datasets' command to be able to download the dataset by ir-dataset library. Queries are provided in the following format "query_id \t query_content"[2]. Relevance judgements are available in the qrels format[3].

(3) Evaluation As the relevance labels for Clinical Trial Track are graded, the official metrics for evaluating the performance of the system on this task are: *NDCG@5* and *NDCG@10*. Besides of official metric, you should report *precision@10* and *reciprocal rank*[4]. You can consider both 1 and 2 labels as relevant for computing *precision@10* and *reciprocal rank*. See Table 2 of this report as an example of a report on this dataset and task.

4. Technical Notes

Please note to the following technical notes for your implementation:

- **Implementation language:** We suggest all students to implement in Python exploiting the elasticsearch library ("python -m pip install elasticsearch==7.9.1").
- **Elasticsearch Version:** Version of Elasticsearch for this assignment is 7.9.1 (link)
- **Retrieval and ranking:** You need to first retrieve 10,000 (10k) documents given a query by the default similarity function of Elasticsearch and then re-compute the similarity score on the all 10k candidate documents based on the variant equation.
- **Enable term vector in mapping:** To be able to compute the variant scores you need to have access to the statistical values for each term in the query and candidate documents (e.g. the document frequency for a term of query in the collection). For this purpose, you need to enable 'term_vector' in the mapping of your index to be able to get that statistical information of terms. Read more information on how to do it on Elasticsearch documentation (link).

5. Report requirements

In addition to the default BM25 built in Elasticsearch, each group should select at least two variants of BM25 in the report. The report's final length should be 3–4 pages. We encourage students to write their reports using the single-column ACM proceeding template (overleaf link) or a similar template.

6. Alternative

An alternative option for the final assignment is selecting some area of IR research (or practice) that you are interested in and want to study further. The final report's length must be 4–6 pages. The report should include the description of the selected dataset and the details of the retrieval task. At least three different retrieval models needed to be implemented and one model must be implemented by Elasticsearch. Please be aware that it is not sufficient if you re-run an existing available implementation instead of implementing by yourself.

[1] http://www.trec-cds.org/2021.html

[2] We refer to the tab character by \t

[3] Information about the qrel format file is available on https://trec.nist.gov/data/qrels_eng/

[4] MRR: If you use https://github.com/cvangysel/pytrec_eval for report this metrics