COL 764 Assignment 2 - Document Reranking Task

2020 - Version 1.0

Deadlines

1. Deadline for final submission of the complete implementation, and the report on the algorithms as well as performance tuning: November 12th 2020, 11:59 PM

Weightage

This assignment is evaluated against 70 marks. The tentative breakup of marks is given in the end of the document.

Instructions

1. This programming assignment is to be done by each student individually. Do not collaborate -either by sharing code, algorithm, and any other pertinent details- with each other.
2. All programs have to be written either using Python/Java/C/C++ programming languages only. Anything else requires explicit prior permission from the instructor.
3. A single tar/zip of the source code has to be submitted. The zip/tar file should be structured such that
   - upon deflating all submission files should be under a directory with the student's registration number. E.g., if a student's registration number is `20XXCSXX999` then the tarball/zip submission should be named `20xxcsxx999.{tgz|zip}` and upon deflating **all contained files** should be under a directory named `./20xxcsxx999` only. If this is not followed, then your submission will be rejected and will not be evaluated.
   - apart from source files the submission tarball/zip file should contain a build mechansim if needed (allowed build systems are Maven and Ant for Java, Makefile for C/C++). It is the responsibility of each student to ensure that it compiles and generates the necessary executables as specified. **Note that we will use only Ubuntu Linux machines to build and run your assignments.** So take care that your file names, paths, argument handling etc. are compatible.
4. You *should not* submit data files, index files, dictionaries etc. If you are planning to use any other "special" library, please talk to the instructor first (or post on Teams).
5. **Note that there will be no deadline extensions. Apart from the usual "please start early" advise, I must warn you that this assignment requires significant amount of implementation effort, as well as some 'manual' tuning of parameters to get good speed up and performance. Do not wait till the end.**

1 Assignment Description

In this assignment the goal is to develop "telescoping" models aimed at improving the precision of results using pseudo-relevance feedback. The dataset we will work with is based on a recently released large collection from Microsoft Bing called MS-MARCO. Although it was originally intended for benchmarking reading comprehension task, it was adapted by TREC for document retrieval task.

Question segment	Percentage of question
Question contains	
YesNo	7.46%
What	34.96%
How	16.8%
Where	3.46%
When	2.71%
Why	1.67%
Who	3.33%
Which	1.79%
Other	27.83%
Question classification	
Description	53.12%
Numeric	26.12%
Entity	8.81%
Location	6.17%
Person	5.78%

Figure 1: Classification of Queries Based on their Answer-type

In MS MARCO, unlike other benchmark query collections, the questions correspond to actual search queries that users submitted to Bing, and therefore may be more representative of a natural distribution of information need that users may want to satisfy using, say, an intelligent assistant.

1.1 Data Description

Document Collection: There are 3,213,835 documents in the collection, released as four tab-separated columns containing strings: docid, url, title, body. Note that the body text has been already preprocessed and stripped of all the HTML tags. The details of the files that are released is given Table 1.

Queries: There are a total of 3,72,406 queries – real-world complex queries submitted through Bing and Cortana. Note that unlike older query logs, these queries are slightly longer in length, include stop words, and are often expressed as questions. See the distribution of questions/queries based on answer-type classification in Figure 1. These queries are split into two sets of 3,67,013 training queries, and 5,193 validation queries which are released to you.

Pseudo-relevance Collection: Unlike previous assignment where you were supposed to build an index and a retrieval system, in this task you are only supposed to rerank the top-100 documents that are retrieved through an initial BM25 model. Note that BM25 model used for initial retrieval used Krovetz stemmer and had removed stopwords (specific stopwords that were removed is not important, but can be provided if needed).
For each query in the training and validation set, you are provided upto top-100 documents from BM25 retrieval round. *Note that in this assignment we are not looking for an implementation of index or an end-to-end retrieval system. You can work with the that you need not build an index and a retrieval system for this initial retrieval from the document collection.*

2 Task

Starting from the top-100 documents for each query, you are supposed to implement the following methods for reranking them to get higher-relevant documents in the higher ranks.

Filename	Size	Records	Description
msmarco-docs.tsv.gz	22GB	3,213,835	tsv: docid, url, title, body
msmarco-doctrain-queries.tsv.gz	15MB	367,013	tsv: qid, query
msmarco-doctrain-top100.tsv.gz	1.8GB	36,701,116	TREC submission: qid, "Q0", docid, rank, score, runstring
msmarco-doctrain-qrels.tsv.gz	8.2MB	367,013	TREC qrels format
msmarco-docdev-queries.tsv.gz	216 KB	5,193	tsv: qid, query
msmarco-docdev-top100.gz	27 MB	519,300	TREC submission: qid, "Q0", docid, rank, score, runstring
msmarco-docdev-qrels.tsv	112 KB	5,193	TREC qrels format

Table 1: Details of the files released for this assignment.

2.1 Task 1: Probabilistic Retrieval Query expansion

You should try expansion by terms starting from 1 to 10 (in steps of 1 word each). Your submission should show the performance variation over this range. You can apply term reweighting (using appropriate estimates of p_i and u_i) if you would like.

2.2 Task 2: Relevance Model based Language Modeling

Use Lavrenko and Croft's relevance model using:

1. Unigram Model with Dirichlet Smoothing
2. Bigram Model with Dirichlet Smoothing with Unigram Backoff

2.3 Evaluation Metrics

Although the shared qrels for training and validation queries are only judged on binary relevance grades (0: non-relevant, 1: relevant), the evaluation will be over 4-point relevance graded qrels.
We will use **nDCG** and **Mean Reciprocal Rank (MRR)** metrics for evaluation.

2.4 Statistical Significance Tests:

Apart from providing the individual task performances, it is important that you also conduct Wilcoxon Paired Rank Test for evaluations. Details of this will be posted in a separate document after 1 week.

2.5 Program Structure

You are expected to submit three implementations:

1. Probabilistic Retrieval Reranking -
   `prob_rerank [query-file] [top-100-file] [collection-file][expansion-limit]`
2. Language Model -
   `lm_rerank [query-file] [top-100-file] [collection-file] [model=uni|bi]`

where,

`query-file`:	file containing the queries in the same tsv format as given in Table 1 for queries file
`top-100-file`:	a file containing the top100 documents in the same format as train and dev top100 files given, which need to be reranked
`collection-file`:	file containing the full document collection (in the same format as msmarco-docs file given)
`expansion-limit`:	is a number ranging from 1—15 that specifies the limit on the number of additional terms in the expanded query
`model=uni|bi`:	it specifies the unigram or the bigram language model that should be used for relevance language model.

2.5.1 Output format

We will again make use of trec_eval tool for generating nDCG and MRR scores. Therefore, as before, it is important that you follow the exact format specification as required by the tool. To reiterate – White space is used to separate columns. The width of the columns in the format is not important, but it is important to have exactly six columns per line with at least one space between the columns.

Lines of results are of the following form

```
1 Q0 pid1    1 2.73 runid1
1 Q0 pid2    1 2.71 runid1
1 Q0 pid3    1 2.61 runid1
1 Q0 pid4    1 2.05 runid1
1 Q0 pid5    1 1.89 runid1
```

where:

- the first column is the topic (query) number.
- the second column is currently unused and should always be Q0.
- the third column is the official identifier of the retrieved passage in context of passage ranking task, and the identifier of the retrieved document in context of document ranking task.
- the fourth column is the rank the passage/document is retrieved.
- the fifth column shows the score (integer or floating point) that generated the ranking. This score must be in descending (non-increasing) order.
- The sixth column is the ID of the run you are submitting.

2.6 Submission Plan

All your submissions should strictly adhere to the formatting requirements given above.

- Submission of your source code on 8th November, and the final version of results.
- You should also submit a README for running your code, and a PDF document containing the implementation details as well as the results nDCG and MRR scores with various parameters that have been specified (preferably as a table).

2.7 Tentative breakup of marks assignment

In general, a submission qualifies for evaluation if and only if it adheres to the specifications given above (arguments, structure, use of external libraries, correct output format, input format adherence, etc.). Given this requirement, the marks assignment for correct implementation of:

probabilistic prf	20
Relevance LM - Unigram	15
Relevance LM - Bigram	10
Statistical Significance Tests	3 x 5 = 15
README and algorithmic details documentation	10
Total	70