IFN647 – Assignment 2

The methodology you will use for Assignment 2 includes the following tasks:

- **Task 1** – Design an Okapi BM25-Based IR Model (***BM25IR***) that ranks documents in each dataset using the corresponding query (topic) for all 50 datasets.
- **Task 2** – Design a Language Model-Based Ranking Model (***LMRM***) that ranks documents in each dataset using the corresponding query (topic) for all 50 datasets.
- **Task 3** – Based on the knowledge you gained from this unit about *learning to rank*, design a Pseudo-Relevance Ranking Model (***PRRM***) to rank documents in each dataset using the corresponding query (topic) for all 50 datasets.
- **Task 4** – Use Python to implement three models: ***BM25IR***, ***LMRM*** and ***PRRM***, and test them on the given 50 datasets for the corresponding 50 queries (topics).
- **Task 5** – Use three effectiveness measures to evaluate the three models.
- **Task 6** – Recommend a model based on significance test and your analysis and describe a possible application scenario.

Assignment 2 Data Collection

It is a subset of RCV1 data collection. It is only for IFN647 students who will be supervised by Prof. Yuefeng Li. Due to copyright and privacy issues, please do not release this data collection to others.

DataSets-1.zip file – It includes 50 Datasets (folders "Dataset101" to "Dataset150") for 50 queries R101 to R150.

"Queries-1.txt" file – It contains definitions for 50 queries (numbered from R101 to R150) for the 50 datasets, where each <Query> element (<Queru>...</Query>) defines a query (topic), including query number (<num>), title (<title>), description (<desc>) and narrative (<narr>).

Example of query R103 "Ferry Boat sinkings" **for Dataset103** is defined as follows:

```
<Query>

<num> Number: R103
<title> Ferry Boat sinkings

<desc> Description:
Documents will report on any sinkings of Ferry Boats throughout
the world.

<narr> Narrative:
Documents that identify any instances where a ferry boat has sunk
or capsized are relevant; only boats identified as ferries should be
considered relevant.

</Query>
```

"EvaluationBenchmark-1.zip" file – It includes relevance judgements (where file "Dataset101.txt" is the benchmark for dataset "Dataset101", etc.) for all documents used in the 50 datasets, where "1" in the third column of each .txt file indicates that the document (the second column) is relevant to the corresponding query (the first column); and "0" means the document is non-relevant.

Assignment 2 Specification

Task 1: Design an Okapi BM25-based IR Model (***BM25IR***) that ranks documents in each dataset using the corresponding query for all 50 datasets.

Inputs: 50 long queries (topics) in *Queries-1.txt* and the corresponding 50 datasets (*Dataset*101, *Dataset*102, …, *Dataset*150).
Output: 50 ranked document files (e.g., for Query *R107*, the output file name is "BM25IR_R107Ranking.dat") for all 50 datasets and save them in the folder "RankingOutputs".

For each long query (topic) *Q*, you need to use the following equation to calculate a BM25 score for each document *D* in the corresponding dataset:

$$score_{BM25}(D,Q) = \sum\nolimits_{i \in Q} IDF_i \cdot \frac{(k_1+1)f_i}{K+f_i} \cdot \frac{(k_2+1)qf_i}{k_2+qf_i}$$

where $IDF_i = log_2(1 + \frac{N-n_i+05}{n_i+0.5})$, *Q* is the title of the long query, $k_1 = 1.2$, k_2=500, $b = 0.75$, $K = k_1*((1-b) + b*dl\ /avdl)$, *dl* is document *D*'s length and *avdl* is the average length of a document in the dataset.

Formally describe your design for ***BM25IR*** in an algorithm to rank documents in each dataset using corresponding query (topic) for all 50 datasets. When you use the above scoring function to rank the documents of each dataset, you also need to answer what the query feature function and document feature function are.
Hint: You can start this task by designing a loop that uses 50 datasets and, in each iteration, provides steps for one dataset and produces what you want.

Task 2: Design a Language Model-based Ranking Model (***LMRM***) that ranks documents in each dataset using the corresponding query for all 50 datasets. For each long query (topic) *Q*, the score of each document in the corresponding dataset needs to be calculated using the Jelinek-Mercer (JM) smoothing technique.

Inputs: 50 long queries (topics) in *Queries-1.txt* and the corresponding 50 datasets (*Dataset*101, *Dataset*102, …, *Dataset*150).
Output: 50 ranked document files (e.g., for Query *R107*, the output file name is "LMRM_R107Ranking.dat") for all 50 datasets and save them in the folder "RankingOutputs".

For each long query (topic) *Rx* (e.g., *R*107), you need to use the following JM smoothing equation to calculate a score (a conditional probability) for each document *D* in the corresponding dataset:

$$score_{JMS}(D,Q) = \sum_{i=1}^{n} log_2((1-\lambda)\frac{f_{q_i,D}}{|D|} + \lambda\frac{c_{q_i}}{|C|})$$

where $f_{qi,D}$ is the number of times query word q_i occurs in document *D*, |*D*| is the number of word occurrences in *D*, c_{qi} is the number of times query word q_i occurs in the dataset *Datasetx* (e.g., *Dataset*107) |*Datasetx*| is the total number of word occurrences in dataset *Datasetx*, and parameter λ = 0.4.

Formally describe your design for ***LMRM*** in an algorithm to rank documents in each dataset using corresponding query (topic) for all 50 datasets.

Task 3. Based on the knowledge you gained from this unit, design a Pseudo-Relevance Ranking Model (***PRRM***) to rank documents in each dataset using the corresponding query for all 50 datasets.

Inputs: 50 long queries (topics) in *Queries-1.txt* and the corresponding 50 datasets (*Dataset*101, *Dataset*102, …, *Dataset*150).
Output: 50 ranked document files (e.g., for Query *R107*, the output file name is "PRRM _R107Ranking.dat") for all 50 datasets and save them in the folder "RankingOutputs".

Formally describe your design for ***PRRM*** in an algorithm to rank documents in each dataset using the corresponding query for all 50 datasets. Your approach should be generic that means it is feasible to be used for other queries (topics). You also need to discuss the differences between ***PRRM*** and the other two models (***BM25IR*** and ***LMRM***).

Task 4. Use Python to implement three models: ***BM25IR***, ***LMRM*** and ***PRRM***, and test them on the given 50 datasets for the corresponding 50 queries (topics).

Design Python programs to implement these three models. You can use a .py file (or a .ipynb file) for each model. Describe the Python package or module (or any open-source software) you used; and the data structures used to represent a single document and a set of documents for each model (you can use different data structures for different models).

For each query, your python programs will produce ranked results and save them into .dat files. For example, for query R107, you save the ranked results of three models (***BM25IR***, ***LMRM*** and ***PRRM***) into "BM25IR_R107Ranking.dat", "LMRM_R107Ranking.dat", and "PRRM_R107Ranking.dat", respectively by using the following format, where the first column is the document *id* (the *itemid* in the corresponding XML document) and the second column is the document score (or probability).

Example ranked documents in LMRM_R109Ranking.dat:

```
(Doc_ID        JMS_Score)

26073 -18.996575728706063
```

```
16953 -19.575365189410793
64476 -19.678721175668624
67717 -20.09535092620512
16575 -20.162511954860065
61540 -20.213316916978787
24340 -21.03770080008531
23398 -21.129021576597964
65289 -21.26902415270888
78626 -21.30648543085368
34684 -21.38488106015813
4933 -21.426280939680773
29314 -21.453484478557137
25832 -21.81950191688901
15776 -22.177398774592348
73598 -22.24085467999975
56519 -22.348774530157616
58676 -22.35078288378288
55187 -22.41526043740232
51139 -22.898756504415203
68812 -22.95611466641387
62293 -22.968990948991497
31530 -22.992543612875522
67144 -22.995214424601535
29729 -23.097055865843032
58651 -23.16596365209444
51841 -23.168835543751797
56735 -23.262508748630015
11402 -23.266725263105656
46566 -23.641819173586104
82229 -23.68014267919321
58504 -23.690778903086088
51780 -23.693716896982615
59340 -23.849979797354955
61554 -23.920015527506834
58582 -23.934652536825737
58428 -24.214999668638654
…
```

You also need to test the three models on the given 50 datasets for the 50 queries (topics) by printing out the top 12 documents for each dataset (in descending order). The output will also be put in the appendix of your final report.

The following is an example of some outputs of BM25IR.

```
…

Query_R102 (Document_ID BM25_Score):

73038 7.478123852368341
58476 6.9803130377238745
26061 6.925868691292516
57914 6.629632855119393
78836 6.206149027309603
76635 6.131239831821994
12769 5.90446895919291
```

```
12767 5.754815530264548
65414 5.234924008221947
25096 5.10323883896895
82227 4.995162279532815
24515 4.788076046512906

Query_R103 (Document_ID BM25_Score):

14314 6.898944476213828
81463 6.736307830421904
27426 5.559197287164682
27106 5.227018970726709
54533 4.892531202397255
59459 4.756761002837321
83370 4.459669025048385
20159 4.155358391944065
26385 3.895697036183199
14069 3.6383109011247283
26386 3.4058865667491736
79396 3.3652517315025676

…

Query_R136 (Document_ID BM25_Score):

79605 5.765896170980428
35104 5.681349440930214
41487 5.244927857268396
43962 4.621207638459407
19488 4.317859701309451
8768 4.0980870514029855
42907 3.519348394686479
46422 3.461321059179448
48692 3.092456180885655
15404 2.9861240705652383
83238 2.928888166531443
3641 2.648343768402267

…

Query_R150 (Document_ID BM25_Score):

…
```

Task 5. Use three effectiveness measures to evaluate the three models.

In this task, you need to use the relevance judgments (**EvaluationBenchmark-1.zip**) to compare with the ranking outputs in the folder of "RankingOutputs" for the selected effectiveness metric for the three models.

You need to use the following three different effectiveness measures to evaluate the document ranking results you saved in the folder "RankingOutputs".

(1) Average precision (and *MAP*),

(2) Precision at rank position 12 (precision@12) and their average, and
(3) Discounted cumulative gain at rank position 12 (p = 12), DCG_{12} (and *their average*)

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

where rel_i = 1 if the document at position i is relevant; otherwise, it is zero.

Evaluation results can be summarized in tables or graphs. For example, Tables 1 to 3 show you example summary tables for the results of the average precision, preciosn@12, and discounted cumulative gain (DCG_{12}).

Table 1. The performance of 3 models on Average Precision

Topic	BM25IR	LMRM	PRRM
R101	…	…	…
R102	0.720	…	…
R103	0.383		
…			
R136	0.365		
…	…	…	…
R150	…	…	…
MAP	…	…	…

Table 2. The performance of 3 models on *precision@*12

Topic	BM25IR	LMRM	PRRM
R101	…	…	…
R102	0.583	…	…
R103	0.417		
…			
R136	0.333		
…	…	…	…
R150	…	…	…
Average	…	…	…

Table 3. The performance of 3 models on DCG_{12}

Topic	BM25IR	LMRM	PRRM
R101	…	…	…
R102	3.687	…	…
R103	1.500		
…			
R136	2.134		
…	…	…	…
R150	…	…	…
Average	…	…	…

Task 6. You need to conduct a significance test to compare models. You can choose a t-test to perform a significance test based on the evaluation results (e.g., in Tables 1, 2 and 3). You can compare models between ***BM25IR*** and ***LMRM***, ***BM25IR*** and ***PRRM***, and ***LMRM*** and ***PRRM***.

Based on t-test results (p-value and t-statistic), you can recommend a model (You want the proposed "PRRM" to be the best because it is your own model). You can perform the t-test using a single effectiveness measure or multiple measures. Generally, using more effectiveness measures provides stronger evidence against the null hypothesis.

(1) Recommend a model based on significance test and your analysis.
(2) Describe a possible application scenario for applying the recommendation model and any potential ethical issues.

Note that if the t-test is unsatisfactory, you can use the evaluation results to refine **PRRM** mode. For example, you can adjust parameter settings or update your design and implementation.

Assignment 2 Requirements

- The following are the frameworks/libraries that you could use for assignment 2:
  (a) sklearn
  (b) nltk
  (c) pandas
  (d) numPy
  (e) Matplotlib
- If you want to use another package or library, you need to get your tutor's approval.
- You can re-use or update your assignment 1 code, the workshop solutions or review question solutions.
- Your programs should be well laid out, easy to read and well commented.
- All items submitted should be clearly labelled with your name and student number.
- Marks will be awarded for design (algorithms), programs (correctness, programming style, elegance, commenting) and evaluation results, according to the marking guide.
- You will lose marks for missing or inaccurate statements of completeness or user manual, and for missing sections, files, or items (see the **final report and poster templates** on Canvas).
- Your results do not need to be the same as the sample outputs.
- We expect **all team members to participate equally** in this assessment project. If you have different contributions, you should provide the percentages and your signature on the cover page of the final report. If you have team conflict issues, your individual contributions will be assessed through peer review and tutor review.
- You can find "What to submit", and "Making Rubric" on Canvas.
- See the marking guide for more details.

END OF ASSIGNMENT 2