IFN647 – Assignment 2

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

- **Task 1** – Design a BM25-based IR model (***BM25***) that ranks documents in each data collection using the corresponding topic (query) for all 50 data collections.
- **Task 2** – Design a Jelinek-Mercer based Language Model (***JM_LM***) that ranks documents in each data collection using the corresponding topic (query) for all 50 data collections.
- **Task 3** – Based on the knowledge you gained from this unit, design a pseudo-relevance model (***My_PRM***) to rank documents in each data collection using the corresponding topic (query) for all 50 data collections.
- **Task 4** – Use Python to implement three models: ***BM25***, ***JM_LM*** and ***My_PRM***, and test them on the given 50 data collections 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.

Data Collection

It is a subset of RCV1 data collection. It is only for IFN647 students who will be supervised by Prof. Yuefeng Li. Please do not release this data collection to others.

Data_Collection.zip file – It includes 50 Datasets (folders "Data_C101" to "Data_C150") for 50 queries R101 to R150.

"the50Queries.txt" file – It contains definitions for 50 queries (numbered from R101 to R150) for the 50 data collections, where each <top> element (<top>...</top>) defines a query (topic), including query number (<num>), title (<title>), description (<desc>) and narrative (<narr>).

Example of query R102 - "`Convicts, repeat offenders`" is defined as follows:

```
<top>

<num> Number: R102
<title>Convicts, repeat offenders

<desc> Description:
Search for information pertaining to crimes committed by people who
have been previously convicted and later released or paroled from
prison.

<narr> Narrative:
Relevant documents are those which cite actual crimes committed by
"repeat offenders" or ex-convicts.  Documents which only generally
discuss the topic or efforts to prevent its occurrence with no
specific cases cited are irrelevant.

</top>
```

"EvaluationBenchmark.zip" file – It includes relevance judgements (where file "dataset101.txt" is the benchmark for data collection "Data_C101", etc.) for all documents used in the 50 data collections (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 Specification

Task 1: Design a BM25-based IR model (***BM25***) that ranks documents in each data collection using the corresponding topic (query) for all 50 data collections.

Inputs: 50 long queries (topics) in *the50Queries.txt* and the corresponding 50 data collections (*Data_C*101, *Data_C*102, …, *Data_C*150).
Output: 50 ranked document files (e.g., for Query *R107*, the output file name is "BM25_R107Ranking.dat") for all 50 data collections and save them in the folder "RankingOutputs".

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

$$\sum_{i \in Q} \log \frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)} \cdot \frac{(k_1+1)f_i}{K+f_i} \cdot \frac{(k_2+1)qf_i}{k_2+qf_i}$$

where *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, the base of the log function is 10. Note that BM25 values can be negative, and you may need to update the above equation to produce non-negative values but keep the resulting documents in the same rank order.

Formally describe your design for ***BM25*** in an algorithm to rank documents in each data collection using corresponding query (topic) for all 50 data collections. When you use the BM25 score to rank the documents of each data collection, you also need to answer what the query feature function and document feature function are.

Task 2: Design a Jelinek-Mercer based Language Model (***JM_LM***) that ranks documents in each data collection using the corresponding topic (query) for all 50 data collections.

Inputs: 50 long queries (topics) in *the50Queries.txt* and the corresponding 50 data collections (*Data_C*101, *Data_C*102, …, *Data_C*150).
Output: 50 ranked document files (e.g., for Query *R107*, the output file name is "JM_LM_R107Ranking.dat") for all 50 data collections and save them in the folder "RankingOutputs".

For each long query (topic) *Rx*, you need to use the following equation to calculate a conditional probability for each document *D* in the corresponding data collection (dataset):

$$p(Rx|D) = \prod_{i=1}^{n} ((1-\lambda)\frac{f_{q_i,D}}{|D|} + \lambda\frac{c_{q_i}}{|Data\_Cx|})$$

where $f_{q_i,D}$ is the number of times query word q_i occurs in document D, $|D|$ is the number of word occurrences in D, c_{q_i} is the number of times query word q_i occurs in the data collection *Data_Cx*, |*Data_Cx*| is the total number of word occurrences in data collection *Data_Cx*, and parameter $\lambda = 0.4$.

Formally describe your design for ***JM_LM*** in an algorithm to rank documents in each data collection using corresponding query (topic) for all 50 data collections. When you use the probabilities to rank the documents of each data collection, you also need to answer what the query feature function and document feature function are.

Task 3. Based on the knowledge you gained from this unit, design a pseudo-relevance model (***My_PRM***) to rank documents in each data collection using the corresponding topic (query) for all 50 data collections.

Inputs: 50 long queries (topics) in *the50Queries.txt* and the corresponding 50 data collections (*Data_C*101, *Data_C*102, …, *Data_C*150).
Output: 50 ranked document files (e.g., for Query *R107*, the output file name is "My_PRM_R107Ranking.dat") for all 50 data collections and save them in the folder "RankingOutputs".

Formally describe your design for ***My_PRM*** in an algorithm to rank documents in each data collection using corresponding query (topic) for all 50 data collections. Your approach should be generic that means it is feasible to be used for other topics (queries). You also need to discuss the differences between My_PRM and the other two models (BM25 and JM_LM).

Task 4. Use Python to implement three models: ***BM25***, ***JM_LM*** and ***My_PRM***, and test them on the given 50 data collections 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. For each long query, your python programs will produce ranked results and save them into .dat files. For example, for query R107, you can save the ranked results of three models into "BM25_R107Ranking.dat", "JM_LM_R107Ranking.dat", and "My_PRM_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).

JM_LM_R107Ranking.dat:

```
71157 4.646997830368691e-09
51576 2.482952802628265e-09
79950 5.693526515139723e-10
77936 4.948756407556077e-10
59244 4.899766717079565e-10
67107 1.2699172048493836e-10
67411 1.2699172048493836e-10
31404 1.1845590141686954e-10
```

```
67673 1.1588524383403817e-10
69377 9.91507340327138e-11
37330 9.808623426595576e-11
71159 7.708887959394712e-11
78164 7.694632354679383e-11
66686 7.487428797770515e-11
28185 7.381770800812406e-11
36212 6.578663298681318e-11
9462 6.430588586817597e-11
41791 6.24004183888205e-11
61532 6.111906054763072e-11
17736 5.7876326010935544e-11
17930 5.653765867655625e-11
58209 5.610656241633241e-11
31761 5.392544836460531e-11
72535 5.253219736109066e-11
18297 5.227009502668939e-11
…
```

JM_LM_R109Ranking.dat:

```
26073 1.9118811474434787e-06
16953 1.2800528530096943e-06
64476 1.1915565266579784e-06
67717 8.926816301185424e-07
16575 8.520775317783744e-07
61540 8.225935429292426e-07
24340 4.645377577898009e-07
23398 4.360444131295939e-07
65289 3.957179508800271e-07
78626 3.8557492531372624e-07
34684 3.6518198686924225e-07
4933 3.548515786425903e-07
29314 3.4822316655628664e-07
25832 2.7019383316239696e-07
15776 2.1083258956747463e-07
73598 2.0176025220913835e-07
56519 1.8721838737889023e-07
58676 1.8695794485615428e-07
55187 1.7878630453477863e-07
51139 1.2787552471190826e-07
68812 1.2289123296331515e-07
62293 1.2179928930763065e-07
31530 1.1982700262014253e-07
67144 1.1960537621808574e-07
29729 1.1145341314943189e-07
…
```

JM_LM_R124Ranking.dat:

```
25328 0.00014790815622152154
9981 0.0001413272837935355
74036 9.795227363584739e-05
21511 9.0441011111624e-05
53830 8.340282947445954e-05
7225 7.312606618634215e-05
19285 6.981634407236306e-05
65633 5.6720708729330295e-05
2345 5.318557360190391e-05
2346 5.259462753302758e-05
4528 5.0669826748284e-05
```

```
65604 4.4606138618874727e-05
20350 4.4566608008677795e-05
46194 3.5830956476360504e-05
4546 3.5509194431959494e-05
61262 3.097474377511931e-05
30653 2.964664844764463e-05
68718 2.873254185005243e-05
14377 2.8390609465851963e-05
13147 2.751914542720251e-05
2415 2.6550914875520497e-05
2725 2.4228663005340617e-05
74846 2.3621577960060493e-05
68716 2.3298848613484753e-05
69236 1.9021927692821243e-05
…
```

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).

You also need to test the three models on the given 50 data collections for the 50 queries (topics) by printing out the top 15 documents for each data collection (in descending order). The output will also be put in the appendix of your final report.

The following is an example of the output of the BM25 model.

Appendix for BM25 Model

```
Query101 (DocID Weight):
…

…
Query107 (DocID Weight):
51576 2.765639300454872
71157 2.386400256599359
77936 2.273108908141271
79950 2.144090202267175
59244 1.8983172340327779
67107 1.6074284655326316
67411 1.6074284655326316
86459 1.128090008008347
31404 0.5807469321665876
41791 0.56351553901341
69377 0.5625603334593126
37330 0.5584756327943026
18297 0.5497333978620387
18529 0.548367073995152
78164 0.5467090324257456
…

Query109 (DocID Weight):
16953 1.7689441459684538
26073 1.64507953070341
61540 1.6346667531459578
4933 1.4873010135488298
64476 1.4002858970645484
16575 1.3834259539490046
```

```
67717 1.3624423453277654
78626 1.2863554347587505
34684 1.2429864221583735
23398 1.2168610353589324
73598 1.1528165978316172
15776 1.1440484476415609
51139 1.0559625097248324
24340 1.046927985206203
29314 1.0339222055290365
…

Query124 (DocID Weight):
21511 1.1101374087659885
19285 1.0454841639375547
25328 0.9942153902538324
74036 0.9940621652520794
65604 0.953091969948307
4528 0.9347773178026801
9981 0.8855131437342743
7225 0.8461955158446076
53830 0.8214559599900999
61262 0.7981291422494461
30653 0.788682520221377
46194 0.7860765283415733
14377 0.7860219825872141
13147 0.6520068520161371
78418 0.5791182459289543
…

Topic R150 (DocID Weight):
…
```

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

In this task, you need to use the relevance judgments (**EvaluationBenchmark.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*@10 (and *their average*), and
(3) Discounted cumulative gain at rank position 10 (p = 10), DCG_{10} (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 several example summary tables for the average precision, preciosn@10, and discounted cumulative gain (DCG_{10}) at rank position 10.

Table 1. The performance of 3 models on average precision (*MAP*)

Topic	BM25	JM_LM	My_PRM
R101	…	…	…
…		…	…
R107	0.303	…	…
R108			
R109	0.641		
…			
R124	0.391		
…	…	…	…
R150	…	…	…
MAP	…	…	…

Table 2. The performance of 3 models on *precision*@10

Topic	BM25	JM_LM	My_PRM
R101	…	…	…
…		…	…
R107	0.200	…	…
R108			
R109	0.600		
…			
R124	0.400		
…	…	…	…
R150	…	…	…
Average	…	…	…

Table 3. The performance of 3 models on DCG_{10}

Topic	BM25	JM_LM	My_PRM
R101	…	…	…
…		…	…
R107	1.131	…	…
R108			
R109	3.751		
…			
R124	2.190		
…	…	…	…
R150	…	…	…
Average	…	…	…

Task 6. Recommend a model based on significance test and your analysis.

You need to conduct a significance test to compare models. You can choose a t-test to perform a significance test on the evaluation results (e.g., in Tables 1, 2 and 3). You can compare models between ***BM25*** and ***JM_LM***, ***BM25*** and ***My_PRM*** and ***JM_LM*** and ***My_PRM***. Based on t-test results (p-value and t-statistic), you can recommend a model (You want the proposed

"My_RPM" 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.

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

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.
- 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" in Canvas (**IFN647 Assignment 2 Specification and Requirements**).

END OF ASSIGNMENT 2