CAB431 Tutorial (Week 10): Relevance (Pseudo) Models vs. IR model
****

In Week 9 workshop, we discussed **a relevance model** to build an information filtering system. It selects features ("training_bm25_wk9.py") to represent the user information need in a file "Model_w5_R102.dat", it then uses the selected features to rank documents in Test_set.

In some real applications, it is hard to get a training set. However, we can use Pseudo Relevance feedback as we mentioned in the lecture notes. In the week 8 workshop, we also discussed how to use a query and an IR model to rank a collection of documents, and then use the top-ranked documents as relevant examples to generate a training set.

Task 1. Design a Pseudo Relevance Model to Rank documents using an initial query Q and generate a training set.

(1) Design python function *bm25*(*coll*, *q*, *df*) to calculate BM25 score for all documents in the "Training_set", where *coll* is the output of *coll.parse_rcv_coll*(*coll_fname*, *stop_words*), see week 9 solution if you do not know this function; *q* is a query (e.g., q = "Convicts, repeat offenders"), and *df* is a dictionary of term document-frequency pairs.
(2) Call function *bm25*() in the main function and save the result into a text file, PRModel_R102.dat, in which each row includes the document number and the corresponding BM25 score, and sorted it in descendent order.
(3) Extend the main function to generate a training set $\underline{D}$ which includes both $\underline{D}^+$ (positive – likely relevant documents) and $\underline{D}^-$ (negative – likely irrelevant documents) in the given un-labelled document set *U* (e.g., *U* = Training_set). The output of this function is a file "PTraining_benchmark.txt", which has the following sample output:

R102 73038 1
R102 26061 1
R102 65414 1
R102 57914 1
…
R102 86912 0
R102 86929 0
R102 11922 0
…

(4) Re-run the week 9 solution (the two .py files) by replacing "Training_benchmark.txt" with "PTraining_benchmark.txt", you may get the following output:

At position 1, precision= 1.0, recall= 0.034482758620689655
At position 2, precision= 1.0, recall= 0.06896551724137931
At position 3, precision= 1.0, recall= 0.10344827586206896
At position 4, precision= 1.0, recall= 0.13793103448275862

At position 5, precision= 1.0, recall= 0.1724137931034483
At position 10, precision= 0.6, recall= 0.20689655172413793
At position 11, precision= 0.6363636363636364, recall= 0.2413793103448276
At position 12, precision= 0.6666666666666666, recall= 0.27586206896551724
At position 13, precision= 0.6923076923076923, recall= 0.3103448275862069
At position 14, precision= 0.7142857142857143, recall= 0.3448275862068966
At position 16, precision= 0.6875, recall= 0.3793103448275862
At position 17, precision= 0.7058823529411765, recall= 0.41379310344827586
At position 18, precision= 0.7222222222222222, recall= 0.4482758620689655
At position 19, precision= 0.7368421052631579, recall= 0.4827586206896552
At position 20, precision= 0.75, recall= 0.5172413793103449
At position 21, precision= 0.7619047619047619, recall= 0.5517241379310345
At position 22, precision= 0.7727272727272727, recall= 0.5862068965517241
At position 23, precision= 0.782608695652174, recall= 0.6206896551724138
At position 24, precision= 0.7916666666666666, recall= 0.6551724137931034
At position 25, precision= 0.8, recall= 0.6896551724137931
At position 26, precision= 0.8076923076923077, recall= 0.7241379310344828
At position 27, precision= 0.8148148148148148, recall= 0.7586206896551724
At position 28, precision= 0.8214285714285714, recall= 0.7931034482758621
At position 30, precision= 0.8, recall= 0.8275862068965517
At position 31, precision= 0.8064516129032258, recall= 0.8620689655172413
At position 32, precision= 0.8125, recall= 0.896551724137931
At position 33, precision= 0.8181818181818182, recall= 0.9310344827586207
At position 34, precision= 0.8235294117647058, recall= 0.9655172413793104
---The average precision = 0.7973420115638065

Task 2. Design a BM25 based IR model.

This task needs to rank documents in "**Test_set**" directly by using function *bm25*(*coll*, *q*, *df*), and save the ranking result in IRModel_R102. Then test the ranking result using the similar evaluation methods as we did in week 9 (see "test_eval_bm25_wk9.py"), and you may get the following output:

At position 2, precision= 0.5, recall= 0.034482758620689655
At position 3, precision= 0.6666666666666666, recall= 0.06896551724137931
At position 5, precision= 0.6, recall= 0.10344827586206896
At position 9, precision= 0.4444444444444444, recall= 0.13793103448275862
At position 10, precision= 0.5, recall= 0.1724137931034483
At position 13, precision= 0.46153846153846156, recall= 0.20689655172413793
At position 14, precision= 0.5, recall= 0.2413793103448276
At position 15, precision= 0.5333333333333333, recall= 0.27586206896551724
At position 16, precision= 0.5625, recall= 0.3103448275862069
At position 17, precision= 0.5882352941176471, recall= 0.3448275862068966

At position 18, precision= 0.6111111111111112, recall= 0.3793103448275862
At position 19, precision= 0.631578947368421, recall= 0.41379310344827586
At position 20, precision= 0.65, recall= 0.4482758620689655
At position 21, precision= 0.6666666666666666, recall= 0.4827586206896552
At position 22, precision= 0.6818181818181818, recall= 0.5172413793103449
At position 23, precision= 0.6956521739130435, recall= 0.5517241379310345
At position 24, precision= 0.7083333333333334, recall= 0.5862068965517241
At position 25, precision= 0.72, recall= 0.6206896551724138
At position 26, precision= 0.7307692307692307, recall= 0.6551724137931034
At position 27, precision= 0.7407407407407407, recall= 0.6896551724137931
At position 28, precision= 0.75, recall= 0.7241379310344828
At position 29, precision= 0.7586206896551724, recall= 0.7586206896551724
At position 30, precision= 0.7666666666666667, recall= 0.7931034482758621
At position 31, precision= 0.7741935483870968, recall= 0.8275862068965517
At position 32, precision= 0.78125, recall= 0.8620689655172413
At position 33, precision= 0.7878787878787878, recall= 0.896551724137931
At position 34, precision= 0.7941176470588235, recall= 0.9310344827586207
At position 35, precision= 0.8, recall= 0.9655172413793104
At position 36, precision= 0.8055555555555556, recall= 1.0
---The average precision = 0.6624714303801168

Task 3. Analyse Your Pseudo Relevance Model.

So far, we have discussed three IF models: **the relevance model** (**RM**, see week 9 workshop), **the pseudo relevance model** (**PRM**, Task 1) and **the IR model** (**IRM**, Task 2), for ranking documents in **Test_set** (a set of unlabelled data). Table 1 shows the experimental results.

Table1. The experimental results over average precision

Topic	**RM**	**IRM**	**PRM**
R102	1.00	0.6625	0.7973
…			

Based on the average precision for topic R102, the pseudo relevance model outperforms the IR model, but worse than the relevance model. You can update the pseudo relevance model (e.g., tuning the bm25-threshold parameter or update the BM25 equation) to check if you can find an optimal one or better one. You can discuss your idea with your tutor or send an email to me if you significantly improve the performance.