Information Retrieval Programming Assignment 5: Learning-to-Rank

Shayan Amani
(993550898, sa1149)

Soheil Gharatappeh
(933639024, sg1147)

Source Code

Please refer to our GitHub repository for the source code.

To run the project, you need to have the folder `outputs/pa5` downloaded in your system.

Produce Features for Learning to Rank

1. First, we need to train our model, using the feature data we got from the ranking files (find features here). To do that with Coordinate Ascent method for MAP we enter the following command:

   ```
   $ java -jar bin/RankLib.jar -train outputs/pa5/RankLibFeatures
   -ranker 4 -metric2t MAP -tolerance .001 -save outputs/pa5/RankLibModel
   -qrel outputs/pa5/qrelfile
   ```

   The weights are:

   ```
   1:0.3226744658228911 2:0.14228732869911265 3:0.086533586782299
   4:-0.44850461869569724
   ```

2. Now, we can obtain our new ranking entering the following command:

   ```
   java -jar bin/RankLib.jar -load outputs/pa5/T1RLTrainedModel -rank
   outputs/pa5/T1RLFeatures -score outputs/pa5/T1ScoreModel
   ```

   Our new ranking is:

Doc Number	Score
5	0.164234643248655
2	0.122639045653859
3	0.106482577952791
6	0.101809480103402
4	0.080668616455723
7	0.035571832174778
9	0.024188846133279
11	0.018497352052406
1	-0.039296566090507
12	-0.089700925075787
10	-0.092205928571239
8	-0.119549063886959

The MAP value on our new ranking is 1.

Combining Different Ranking Functions

1. To implement a k-fold cross validation, we need to use `[ -kcv <k> ]` option in our command. So, if we enter:

   `$ java -jar bin/RankLib.jar -kcv 5 outputs/pa5/RankLibFeatures -ranker 4 -metric2t MAP -save outputs/pa5/RankLibModel` we can have a 5-fold cross validation.