Assignment 4

Fall 2017
CS834 Introduction to Information Retrieval
Dr. Michael Nelson

Orkun Krand

December 4, 2017

1 Question 8.3

1.1 Question

For one query in the CACM collection (provided at the book website), generate a ranking using Galago, and then calculate average precision, NDCG at 5 and 10, precision at 10, and the reciprocal rank by hand.

1.2 Methodology

Using Galago, I was able to create an index of the CACM corpus. Afterwards, I used Galago to search for the queries. I used `galago.py` to issue the queries and find the required values. I utilized Requests and Beautiful Soup to issue the queries and parse the results.

Precision is the true positive divided by the total positive (true and false) while recall is true positive divided by true positive and false negative. In order to get the precision at specific rankings, I had to calculate cumulative precision at each rank. For the average precision, I computed the sum of precisions at each rank for the retrieved relevant documents and then divided this sum by the same set. NDCG was calculated by dividing DCG by IDCG. The two values were computed using the formulas in the textbook[1]. Reciprocal rank is 1 divided by the rank of the first relevant document.

1.3 Results

Using query 13 (code optimization for space efficiency), I got the following results:

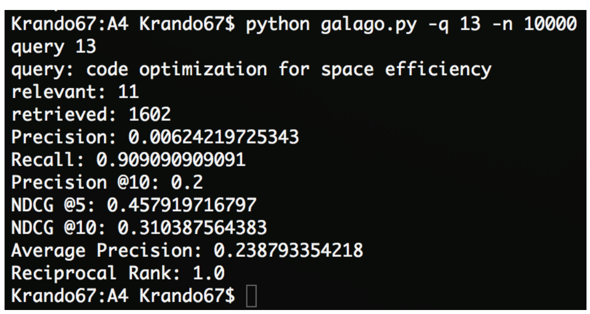

Figure 1: Query 13 Results

2 Question 8.4

2.1 Question

For two queries in the CACM collection, generate two uninterpolated recall-precision graphs, a table of interpolated precision values at standard recall levels, and the average interpolated recall-precision graph.

2.2 Methodology

Deciding to go with queries 9 and 10, the following graphs were created using `graph.R` and the table was created using `ipr.py`. In order to minimize clutter, I joined the two uninterpolated recall-precision graphs into one graph, distinguished by color.

2.3 Results

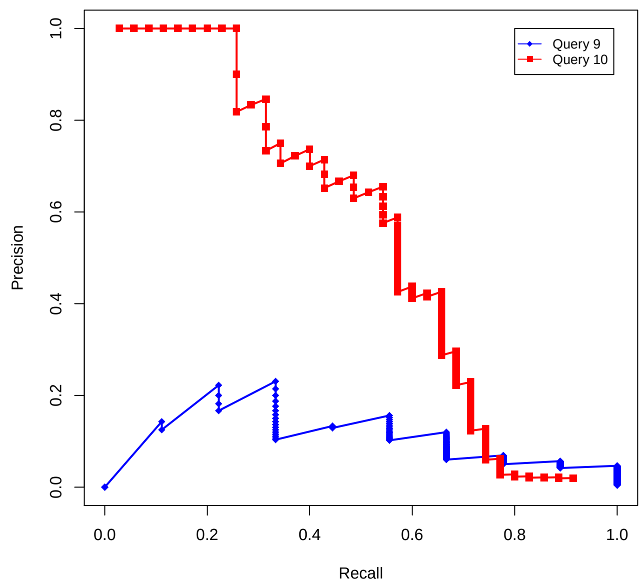

Figure 2: Uninterpolated recall-precision graphs for queries 9 and 10

Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Query 9	0.231	0.231	0.231	0.231	0.156	0.156	0.12	0.0693	0.0567	0.0466	0.0466
Query 10	1	1	1	0.846	0.714	0.655	0.426	0.229	0.0237	0.0198	0.0198
Average	0.615	0.615	0.615	0.538	0.435	0.406	0.273	0.149	0.0402	0.0332	0.0332

Table 1: Interpolated precision values at standard recall levels

3 Question 8.5

3.1 Question

Generate the mean average precision, recall-precision graph, average NDCG at 5 and 10, and precision at 10 for the entire CACM query set.

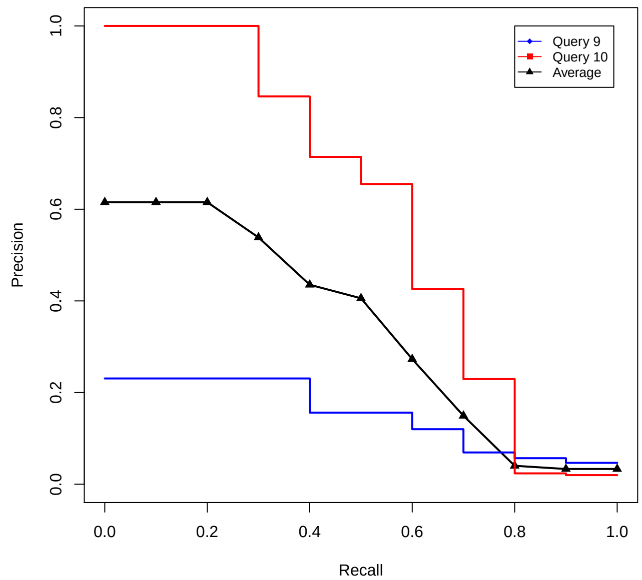

Figure 3: Average interpolated recall-precision graphs for queries 9 and 10

3.2 Methodology

`entirecacm.py` was created to calculate the values for the entire query set. `graph.R` was modified to produce the requested recall-precision graph.

3.3 Results

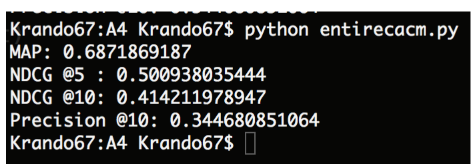

Figure 4: Entire CACM Results

4 Question 8.7

4.1 Question

Another measure that has been used in a number of evaluations is R-precision. This is defined as the precision at R documents, where R is the number of relevant documents for a query. It is used in situations where there is a large variation in the number of relevant documents per query. Calculate the average *R-precision* for the CACM query set and compare it to the other measures.

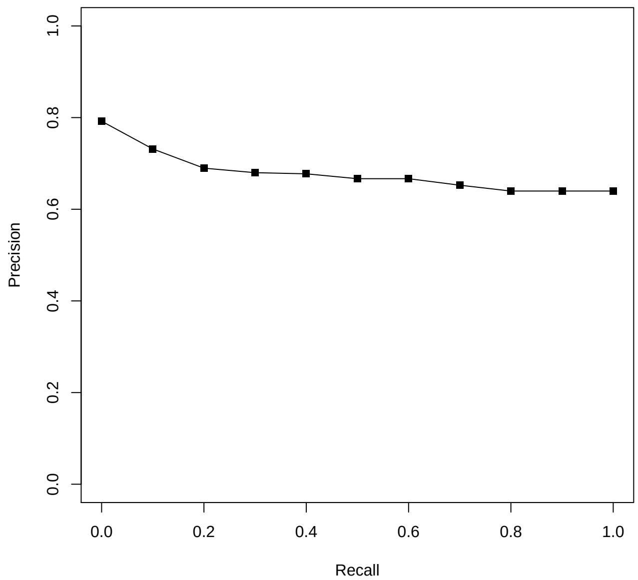

Figure 5: CACM queries recall-precision graph

4.2 Methodology

Using `Rprecision.py`, which was run over all documents, I was able to get the R precision at any query I want. Other measures were displayed along with the result for easy comparison.

4.3 Results

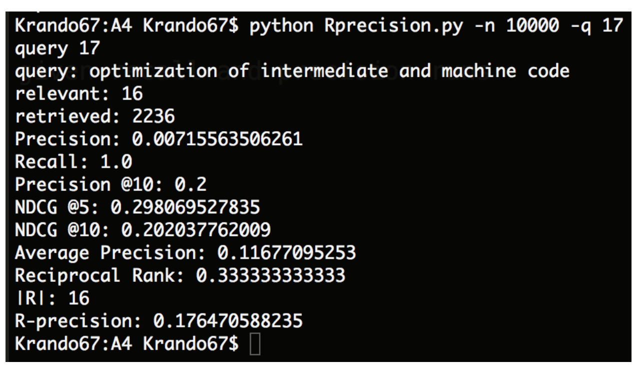

Figure 6: R precision for query 17

At all queries I tried, I observed that R-precision value is very close to the average precision. So it seems that it's a good candidate for an alternative. However, it may not be as stable with a small set of relevant documents.

5 Question 8.9

5.1 Question

For one query in the CACM collection, generate a ranking and calculate BPREF. Show that the two formulations of BPREF give the same value.

5.2 Methodology

Two functions which calculate the two equations for BPREF found in the textbook[1] were added to `galago.py`.

5.3 Results

After spending time writing the code for BPREF which can be found in `bpref.py`, I had to run it at least 3 times. The results are below. While the two BPREF values weren't exactly the same they were very close.

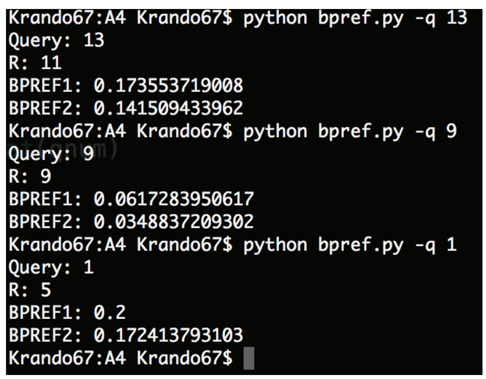

Figure 7: BPREF values for multiple queries

References

[1] Croft, William Bruce, et al. *Search Engines: Information Retrieval in Practice*. Pearson, 2010.