Evaluation reports:

My code for testing it: (you can find it on `evaluation.py`)
I collected the initial two sample queries, while Mahan Beyhaghi colledted the remaining ones. We queried the IMDB website, from which we collected the top 15 results. These results were then assigned scores ranging from 20 to 5, based on their respective positions in the search results. We utilized these scores to calculate the Discounted Cumulative Gain (DCG) and Normalized Discounted Cumulative Gain (NDCG) metrics.

```
if __name__ == '__main__':
    queries = ['dune', 'harry potter', 'spiderman', 'matrix', 'batman']
    actual = [[('tt15239678', 20), ('tt0087182', 19), ('tt1160419', 18),
('tt0142032', 17), ('tt31378509', 16),
               ('tt0287839', 15), ('tt10466872', 14), ('tt1935156', 13),
('tt15331462', 12), ('tt11835714', 11),
               ('tt12451788', 10), ('tt14450978', 9), ('tt31613341', 8),
('tt0099474', 7), ('tt31613353', 6)],
               [('tt0241527', 20), ('tt0330373', 19), ('tt0304141', 18),
('tt0295297', 17), ('tt1201607', 16),
                ('tt0373889', 15), ('tt0417741', 14), ('tt0926084', 13),
('tt13918446', 12), ('tt16116174', 11),
                ('tt1756545', 10), ('tt15431326', 9), ('tt3731688', 8),
('tt2335590', 7), ('tt7467820', 6)],
                [("tt0145487", 20), ("tt10872600", 19), ("tt0948470", 18),
("tt1872181", 17), ("tt2705436", 16),
                 ("tt0112175", 15), ("tt12122034", 14), ("tt0413300", 13),
("tt4633694", 12), ("tt2250912", 11),
                 ("tt6320628", 10), ("tt9362722", 9), ("tt0316654", 8),
("tt0076975", 7), ("tt16360004",6)],
                 [("tt0133093", 20), ("tt10838180", 19), ("tt0234215", 18),
("tt0242653", 17), ("tt0106062", 16),
                  ("tt30849138", 15), ("tt0410519", 14), ("tt9847360", 13),
("tt31998838", 12), ("tt30749809", 11),
                  ("tt0365467", 10), ("tt0364888", 9), ("tt11749868", 8),
("tt0303678", 7), ("tt0274085", 6)],
                  [("tt0096895", 20), ("tt1877830", 19), ("tt0059968", 18),
("tt0372784", 17), ("tt0103359", 16),
                   ("tt0118688", 15), ("tt0103776", 14), ("tt0112462", 13),
("tt2975590", 12), ("tt19850008", 11),
                   ("tt0147746", 10), ("tt0398417", 9), ("tt0035665", 8),
("tt4116284", 7), ("tt0060153", 6)]
                  ]
```

The rest of the Code:

```python
methods = ['ltn.lnn', 'ltc.lnc', 'OkapiBM25']
predicted = {method: [] for method in methods}

for query in queries:
    for method in methods:
        search_term = query
        search_max_num = 10
        search_weights = [1, 1, 1]
        result = utils.search(
                    search_term,
                    search_max_num,
                    method,
                    search_weights,
                )
        query_predicted = []
        for res in result:
            query_predicted.append(res[0])
        predicted[method].append(query_predicted)

for method in methods:
    evaluation = Evaluation(method)
    evaluation.calculate_evaluation(actual, predicted[method], queries)
```

Outputs:

```
Name: ltn.lnn

Queries: dune - harry potter - spiderman - matrix - batman

Evaluation Metrics:
--------------------
All Precisions: [0.8333, 0.5000, 0.9000, 0.7500, 0.2857]
Mean Precision: 0.6538
All Recalls: [0.3333, 0.3333, 0.6000, 0.4000, 0.1333]
Mean Recall: 0.3600
All F1s: [0.4762, 0.4000, 0.7200, 0.5217, 0.1818]
Mean F1: 0.4599
Average Precision (AP): [1.0000, 1.0000, 0.9765, 0.6968, 0.5000]
Mean Average Precision (MAP): 0.8347
All Discounted Cumulative Gains (DCG): [585576.4504, 1186346.9292,
458287.0892, 915856.1489, 387238.5365]
Mean DCG: 706661.0308
All Normalized Discounted Cumulative Gains (NDCG): [0.3651, 0.7351,
0.2840, 0.5682, 0.2406]
Mean NDCG: 0.4386
All Reciprocal Ranks (RR): [1.0000, 1.0000, 1.0000, 0.5000, 0.5000]
Mean Reciprocal Rank (MRR): 0.8000
--------------------
```

Outputs:

```
Name: ltc.lnc

Queries: dune - harry potter - spiderman - matrix - batman

Evaluation Metrics:
--------------------
All Precisions: [0.8333, 0.2000, 0.9000, 0.7500, 0.2857]
Mean Precision: 0.5938
All Recalls: [0.3333, 0.1333, 0.6000, 0.4000, 0.1333]
Mean Recall: 0.3200
All F1s: [0.4762, 0.1600, 0.7200, 0.5217, 0.1818]
Mean F1: 0.4119
Average Precision (AP): [0.7100, 0.3095, 0.9765, 0.9151, 0.4500]
Mean Average Precision (MAP): 0.6722
All Discounted Cumulative Gains (DCG): [400619.6351, 5493.3333,
632611.0630, 830974.1627, 381494.4698]
Mean DCG: 450238.5328
All Normalized Discounted Cumulative Gains (NDCG): [0.2498, 0.0034,
0.3920, 0.5155, 0.2370]
Mean NDCG: 0.2795
All Reciprocal Ranks (RR): [0.5000, 0.3333, 1.0000, 1.0000, 0.5000]
Mean Reciprocal Rank (MRR): 0.6667
--------------------
```

Outputs:

```
Name: OkapiBM25

Queries: dune - harry potter - spiderman - matrix - batman

Evaluation Metrics:
--------------------
All Precisions: [0.8333, 0.7000, 0.9000, 0.7500, 0.2857]
Mean Precision: 0.6938
All Recalls: [0.3333, 0.4667, 0.6000, 0.4000, 0.1333]
Mean Recall: 0.3867
All F1s: [0.4762, 0.5600, 0.7200, 0.5217, 0.1818]
Mean F1: 0.4919
Average Precision (AP): [0.7100, 0.9129, 1.0000, 0.7345, 0.4167]
Mean Average Precision (MAP): 0.7548
All Discounted Cumulative Gains (DCG): [400619.6351, 1039407.7958,
809742.8366, 834670.4512, 318593.6378]
Mean DCG: 680606.8713
All Normalized Discounted Cumulative Gains (NDCG): [0.2498, 0.6441,
0.5018, 0.5178, 0.1980]
Mean NDCG: 0.4223
All Reciprocal Ranks (RR): [0.5000, 1.0000, 1.0000, 0.5000, 0.3333]
Mean Reciprocal Rank (MRR): 0.6667
--------------------
```