## System Implementation

NB: please copy and paste the query.py module to the original simplesearch directory before testing it since it has dependencies within this directory.

**Step 1**

**Relevance Scoring**

The implementation for relevance scoring is in the python module named relevanceScoring.py.

In this Python module, I created a simple console-based program to assign relevance scores to documents for each query. The module has three main functions: getQueries, getDocs, and main. the module rotates queries and documents on the console. For each query + document combination, i assign a relevance core through the console. the results are then store in the corresponding relevance file, i.e relevance.1 for query 1. relevance.2 for query2 and relevance.3 for query 3.

**Step 2**

For Step 2, I extended the query.py Python module to include an implementation for blind relevance feedback. The additions to the code include:

1. The first step involved wrapping the main search functionality from the original module into a function. This function is named getResults()
2. A new function relFeedback(results, numTerms) that takes the search results and a specified number of popular terms (numTerms) as input. The function calls getPopularTerms to obtain the most popular terms from the top search results and then updates the system argument field (sys.argv[2]) with the concatenated popular terms. Finally, it calls getResults() again to perform a second search with the updated query.
3. Another new function getPopularTerms(results, k_terms) that extracts the top k_terms popular terms from the documents returned by the search results. It tokenizes the documents into terms, counts the occurrences of each term, and then concatenates the most popular terms into a single string. This string is then returned for use as an updated query.
4. After the initial search, I called the relFeedback(results, 20) function to perform a second search with the concatenated popular terms. This allows the search engine to retrieve more relevant documents based on the blind relevance feedback.
5. Note! The third search involves manually adding the popular query results as additional query terms to the original query and submitting another search.

By extending the query.py module with these additions, I incorporated blind relevance feedback into the search engine, which can potentially improve the retrieval of relevant results for a given query.

NB: Additional documentation is provided in the module file.

**Results**

NB: Calculations are included in the appendix at the end of this document

A Total of 9 queries were submitted to the search engine. These include the original query found in the query file within the testbed directory as variations of the original query, that is, for each query in the directory there are other two related queries. The original query is submitted first then popular terms are extracted from each of the documents returned by the initial query. The ten most popular terms are then concatenated and submitted as a second query. The last query involves concatenating the popular terms with the original search query. This forms part of the relevance feedback system.

Query 1:

MAP: Before (0.7889) vs. After (0.7041)

NDCG: Before (0.9533) vs. After (0.7489)

For Query 1, the precision values tend to decrease after applying blind relevance feedback, whereas the recall values exhibit a mixed pattern. This suggests that the search system retrieved more irrelevant documents after feedback, which led to a reduction in precision. As for recall, the expanded query might have captured additional relevant documents in some instances, while the introduction of noise or irrelevant terms could have hampered recall in others. The decline in both MAP and NDCG indicates that the overall search performance has degraded for Query 1 after applying blind relevance feedback.

Query 2:

MAP: Before (0.55) vs. After (0.75)

NDCG: Before (0.6632) vs. After (0.75)

In the case of Query 2, both precision and recall improve after applying blind relevance feedback. This indicates that the expanded query captured more relevant documents while maintaining a high level of precision. The feedback process likely added valuable terms that were absent from the original query, leading to the retrieval of more relevant documents without significantly increasing the number of irrelevant results. The improvement in both MAP and NDCG for Query 2 demonstrates that blind relevance feedback has had a positive impact on search performance.

Query 3:

MAP: Before (0.5771) vs. After (0.5)

NDCG: Before (0.7113) vs. After (0.6309)

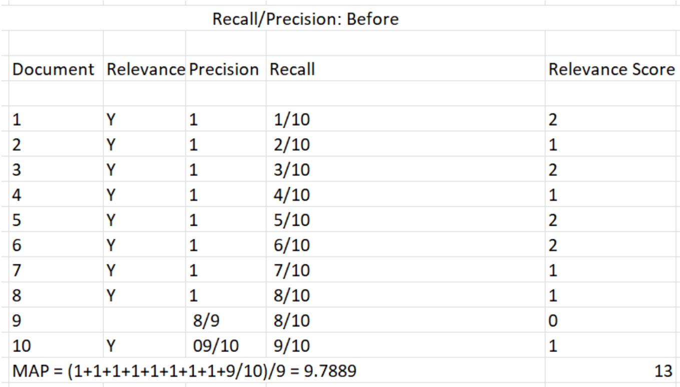
For Query 3, precision values generally decrease after applying blind relevance feedback, while recall values show a mixed trend. Similar to Query 1, the feedback process may have introduced irrelevant terms or noise, causing a decline in precision. The mixed trend in recall values again indicates that the expanded query might have captured additional relevant documents in some cases while hindering recall in others. The decrease in both MAP and NDCG for Query 3 implies that search performance has worsened after applying blind relevance feedback.

In conclusion, the effectiveness of blind relevance feedback depends on the specific query, as evidenced by the mixed results for precision and recall. For Query 2, the feedback process improved search performance across all metrics, while it resulted in a decline for Queries 1 and 3. This underlines the significance of considering the unique context and characteristics of each query when using blind relevance feedback.

These findings suggest that incorporating additional techniques, such as query expansion based on user input, domain knowledge, or other contextual information, might prove more beneficial for consistently enhancing search performance. By refining the query expansion process and carefully managing the trade-off between precision and recall, search systems can aim to provide a better user experience and more relevant results.

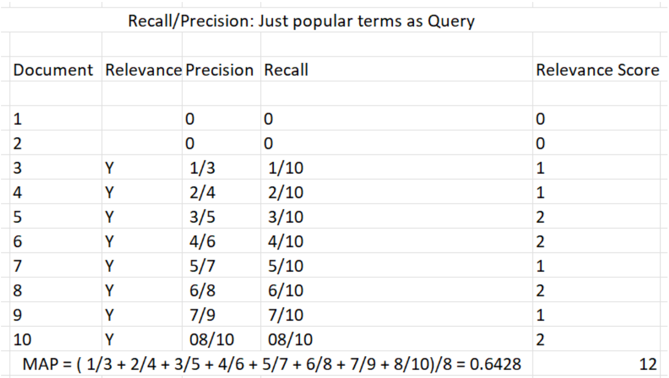
## Appendix (Calculations)

Query 1



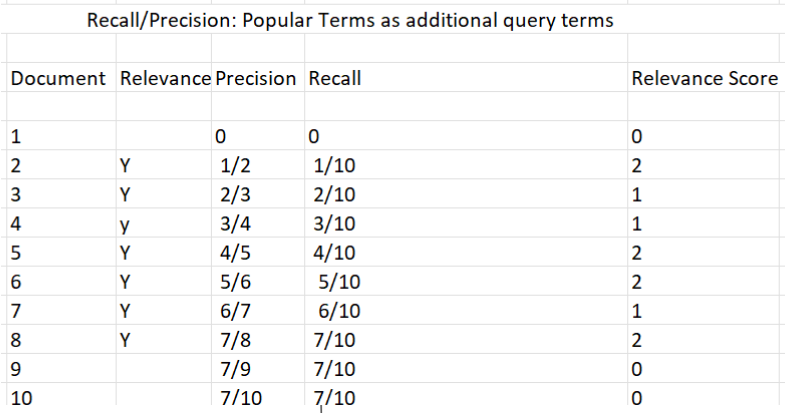
Recall/Precision: Before

| Document | Relevance | Precision | Recall | Relevance Score |
|---|---|---|---|---|
| 1 | Y | 1 | 1/10 | 2 |
| 2 | Y | 1 | 2/10 | 1 |
| 3 | Y | 1 | 3/10 | 2 |
| 4 | Y | 1 | 4/10 | 1 |
| 5 | Y | 1 | 5/10 | 2 |
| 6 | Y | 1 | 6/10 | 2 |
| 7 | Y | 1 | 7/10 | 1 |
| 8 | Y | 1 | 8/10 | 1 |
| 9 |  | 8/9 | 8/10 | 0 |
| 10 | Y | 09/10 | 9/10 | 1 |
| MAP = (1+1+1+1+1+1+1+1+9/10)/9 = 9.7889 |  |  |  | 13 |



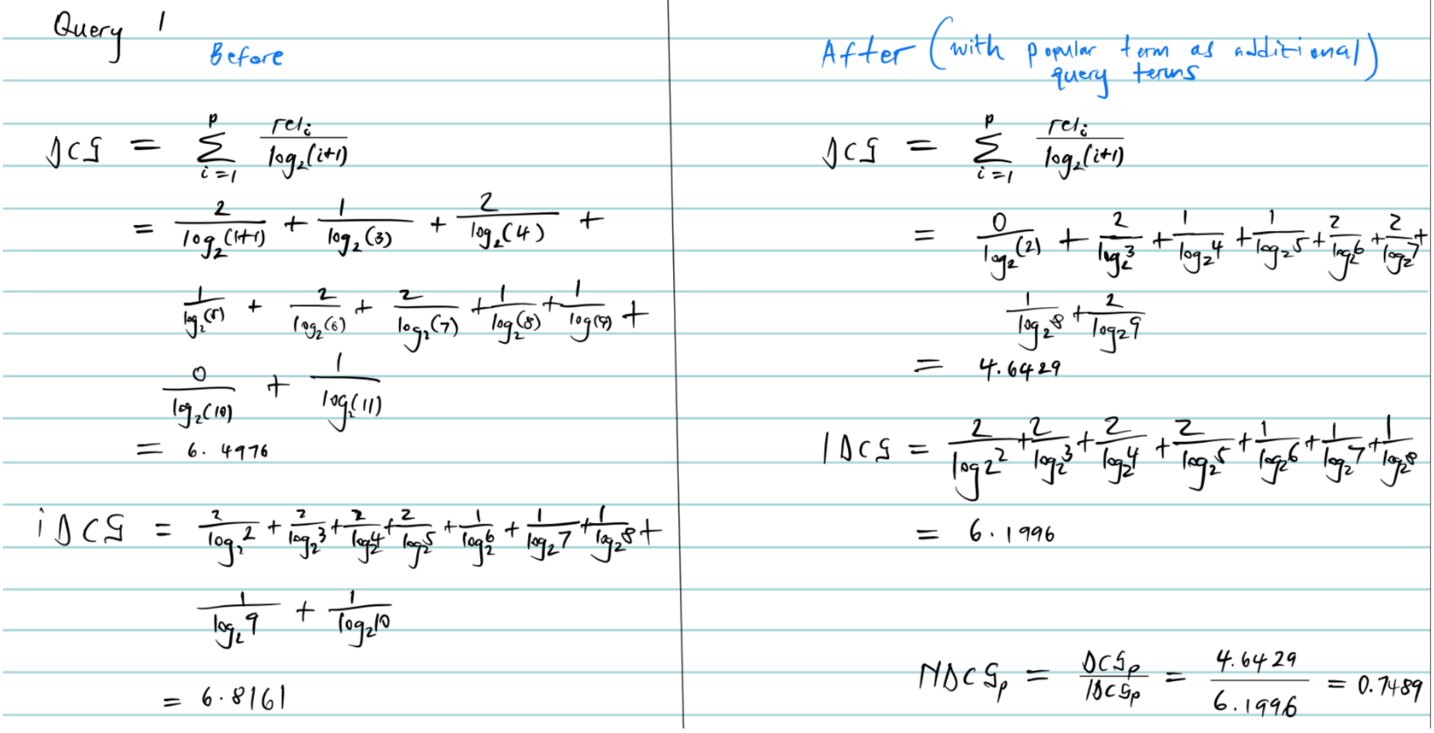
Recall/Precision: Just popular terms as Query

| Document | Relevance | Precision | Recall | Relevance Score |
|---|---|---|---|---|
| 1 |  | 0 | 0 | 0 |
| 2 |  | 0 | 0 | 0 |
| 3 | Y | 1/3 | 1/10 | 1 |
| 4 | Y | 2/4 | 2/10 | 1 |
| 5 | Y | 3/5 | 3/10 | 2 |
| 6 | Y | 4/6 | 4/10 | 2 |
| 7 | Y | 5/7 | 5/10 | 1 |
| 8 | Y | 6/8 | 6/10 | 2 |
| 9 | Y | 7/9 | 7/10 | 1 |
| 10 | Y | 08/10 | 08/10 | 2 |
| MAP = ( 1/3 + 2/4 + 3/5 + 4/6 + 5/7 + 6/8 + 7/9 + 8/10)/8 = 0.6428 |  |  |  | 12 |



Recall/Precision: Popular Terms as additional query terms

| Document | Relevance | Precision | Recall | Relevance Score |
|---|---|---|---|---|
| 1 |  | 0 | 0 | 0 |
| 2 | Y | 1/2 | 1/10 | 2 |
| 3 | Y | 2/3 | 2/10 | 1 |
| 4 | y | 3/4 | 3/10 | 1 |
| 5 | Y | 4/5 | 4/10 | 2 |
| 6 | Y | 5/6 | 5/10 | 2 |
| 7 | Y | 6/7 | 6/10 | 1 |
| 8 | Y | 7/8 | 7/10 | 2 |
| 9 |  | 7/9 | 7/10 | 0 |
| 10 |  | 7/10 | 7/10 | 0 |



Query 1

Before

$$DCG = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

$$= \frac{2}{\log_2(1+1)} + \frac{1}{\log_2(3)} + \frac{2}{\log_2(4)} + \frac{1}{\log_2(5)} + \frac{2}{\log_2(6)} + \frac{2}{\log_2(7)} + \frac{1}{\log_2(8)} + \frac{1}{\log(9)} + \frac{0}{\log_2(10)} + \frac{1}{\log_2(11)}$$

$$= 6.4976$$

$$iDCG = \frac{2}{\log_2 2} + \frac{2}{\log_2 3} + \frac{2}{\log_2 4} + \frac{2}{\log_2 5} + \frac{1}{\log_2 6} + \frac{1}{\log_2 7} + \frac{1}{\log_2 8} + \frac{1}{\log_2 9} + \frac{1}{\log_2 10}$$

$$= 6.8161$$

After (with popular term as additional query terms)

$$DCG = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

$$= \frac{0}{\log_2(2)} + \frac{2}{\log_2 3} + \frac{1}{\log_2 4} + \frac{1}{\log_2 5} + \frac{2}{\log_2 6} + \frac{2}{\log_2 7} + \frac{1}{\log_2 8} + \frac{2}{\log_2 9}$$

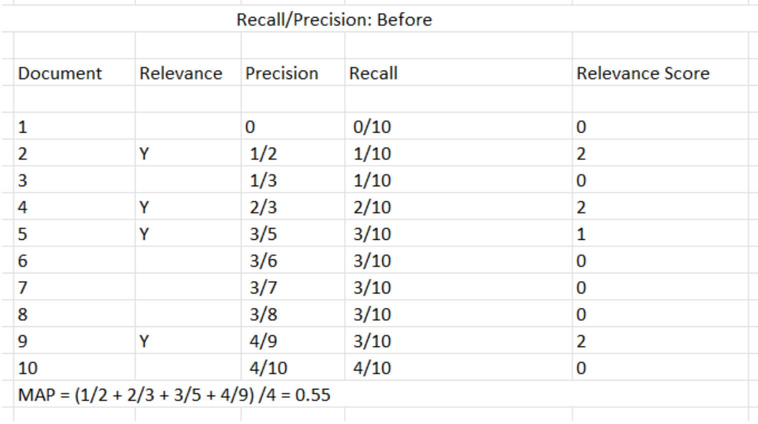
$$= 4.6429$$

$$IDCG = \frac{2}{\log_2 2} + \frac{2}{\log_2 3} + \frac{2}{\log_2 4} + \frac{2}{\log_2 5} + \frac{1}{\log_2 6} + \frac{1}{\log_2 7} + \frac{1}{\log_2 8}$$

$$= 6.1996$$

$$NDCG_p = \frac{DCG_p}{IDCG_p} = \frac{4.6429}{6.1996} = 0.7489$$

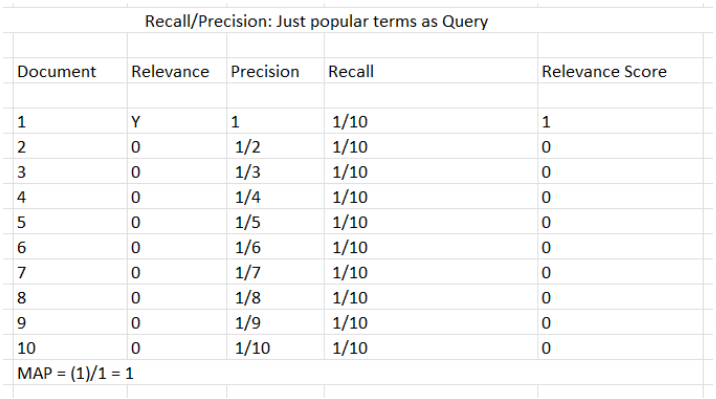
## Query 2



Recall/Precision: Before

| Document | Relevance | Precision | Recall | Relevance Score |
|---|---|---|---|---|
| 1 |  | 0 | 0/10 | 0 |
| 2 | Y | 1/2 | 1/10 | 2 |
| 3 |  | 1/3 | 1/10 | 0 |
| 4 | Y | 2/3 | 2/10 | 2 |
| 5 | Y | 3/5 | 3/10 | 1 |
| 6 |  | 3/6 | 3/10 | 0 |
| 7 |  | 3/7 | 3/10 | 0 |
| 8 |  | 3/8 | 3/10 | 0 |
| 9 | Y | 4/9 | 3/10 | 2 |
| 10 |  | 4/10 | 4/10 | 0 |

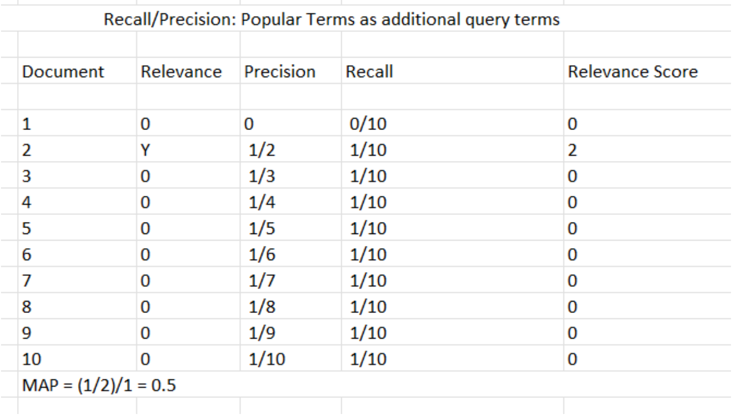
MAP = (1/2 + 2/3 + 3/5 + 4/9) /4 = 0.55



Recall/Precision: Just popular terms as Query

| Document | Relevance | Precision | Recall | Relevance Score |
|---|---|---|---|---|
| 1 | Y | 1 | 1/10 | 1 |
| 2 | 0 | 1/2 | 1/10 | 0 |
| 3 | 0 | 1/3 | 1/10 | 0 |
| 4 | 0 | 1/4 | 1/10 | 0 |
| 5 | 0 | 1/5 | 1/10 | 0 |
| 6 | 0 | 1/6 | 1/10 | 0 |
| 7 | 0 | 1/7 | 1/10 | 0 |
| 8 | 0 | 1/8 | 1/10 | 0 |
| 9 | 0 | 1/9 | 1/10 | 0 |
| 10 | 0 | 1/10 | 1/10 | 0 |

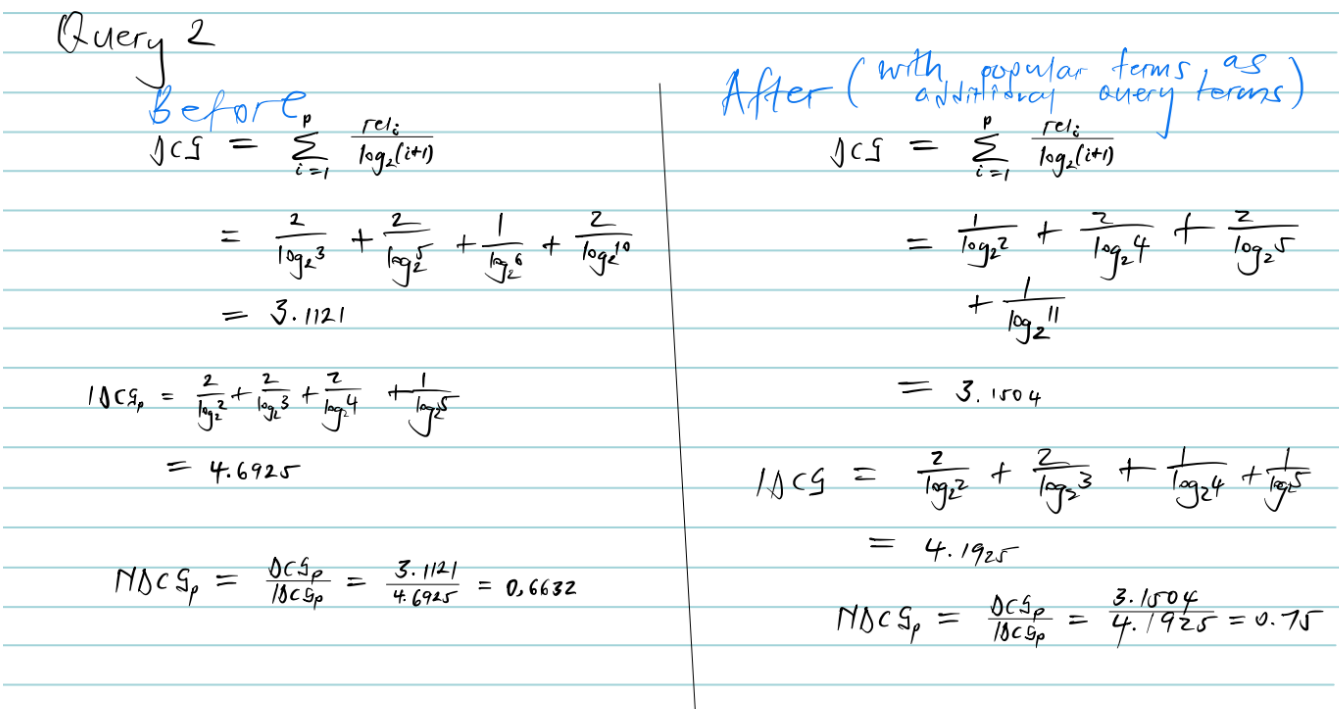
MAP = (1)/1 = 1



Recall/Precision: Popular Terms as additional query terms

| Document | Relevance | Precision | Recall | Relevance Score |
|---|---|---|---|---|
| 1 | 0 | 0 | 0/10 | 0 |
| 2 | Y | 1/2 | 1/10 | 2 |
| 3 | 0 | 1/3 | 1/10 | 0 |
| 4 | 0 | 1/4 | 1/10 | 0 |
| 5 | 0 | 1/5 | 1/10 | 0 |
| 6 | 0 | 1/6 | 1/10 | 0 |
| 7 | 0 | 1/7 | 1/10 | 0 |
| 8 | 0 | 1/8 | 1/10 | 0 |
| 9 | 0 | 1/9 | 1/10 | 0 |
| 10 | 0 | 1/10 | 1/10 | 0 |

MAP = (1/2)/1 = 0.5



Query 2

Before

$$DCG = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

$$= \frac{2}{\log_2 3} + \frac{2}{\log_2 5} + \frac{1}{\log_2 6} + \frac{2}{\log_2 10}$$

$$= 3.1121$$

$$IDCG_p = \frac{2}{\log_2 2} + \frac{2}{\log_2 3} + \frac{2}{\log_2 4} + \frac{1}{\log_2 5}$$

$$= 4.6925$$

$$NDCG_p = \frac{DCG_p}{IDCG_p} = \frac{3.1121}{4.6925} = 0,6632$$

After (with popular terms, as additional query terms)

$$DCG = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

$$= \frac{1}{\log_2 2} + \frac{2}{\log_2 4} + \frac{2}{\log_2 5} + \frac{1}{\log_2 11}$$

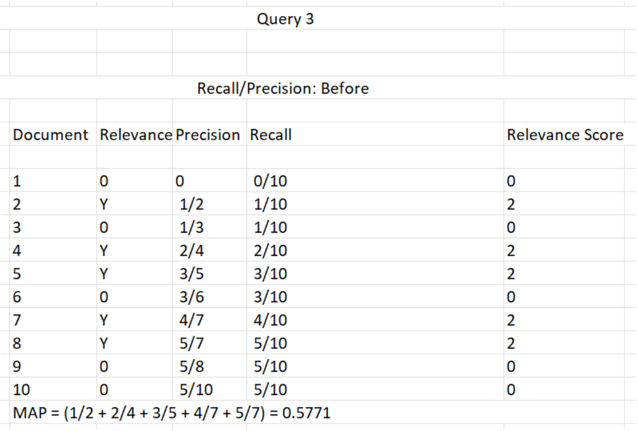
$$= 3.1504$$

$$IDCG = \frac{2}{\log_2 2} + \frac{2}{\log_2 3} + \frac{1}{\log_2 4} + \frac{1}{\log_2 5}$$

$$= 4.1925$$

$$NDCG_p = \frac{DCG_p}{IDCG_p} = \frac{3.1504}{4.1925} = 0.75$$

## Query 3

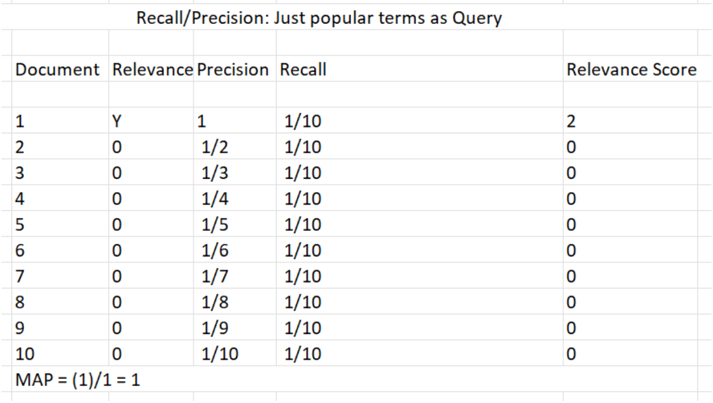


Query 3

Recall/Precision: Before

| Document | Relevance | Precision | Recall | Relevance Score |
|---|---|---|---|---|
| 1 | 0 | 0 | 0/10 | 0 |
| 2 | Y | 1/2 | 1/10 | 2 |
| 3 | 0 | 1/3 | 1/10 | 0 |
| 4 | Y | 2/4 | 2/10 | 2 |
| 5 | Y | 3/5 | 3/10 | 2 |
| 6 | 0 | 3/6 | 3/10 | 0 |
| 7 | Y | 4/7 | 4/10 | 2 |
| 8 | Y | 5/7 | 5/10 | 2 |
| 9 | 0 | 5/8 | 5/10 | 0 |
| 10 | 0 | 5/10 | 5/10 | 0 |

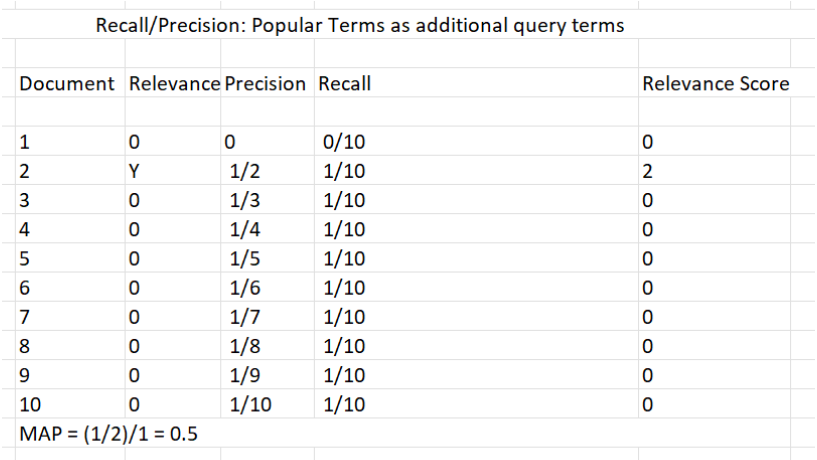
MAP = (1/2 + 2/4 + 3/5 + 4/7 + 5/7) = 0.5771



Recall/Precision: Just popular terms as Query

| Document | Relevance | Precision | Recall | Relevance Score |
|---|---|---|---|---|
| 1 | Y | 1 | 1/10 | 2 |
| 2 | 0 | 1/2 | 1/10 | 0 |
| 3 | 0 | 1/3 | 1/10 | 0 |
| 4 | 0 | 1/4 | 1/10 | 0 |
| 5 | 0 | 1/5 | 1/10 | 0 |
| 6 | 0 | 1/6 | 1/10 | 0 |
| 7 | 0 | 1/7 | 1/10 | 0 |
| 8 | 0 | 1/8 | 1/10 | 0 |
| 9 | 0 | 1/9 | 1/10 | 0 |
| 10 | 0 | 1/10 | 1/10 | 0 |

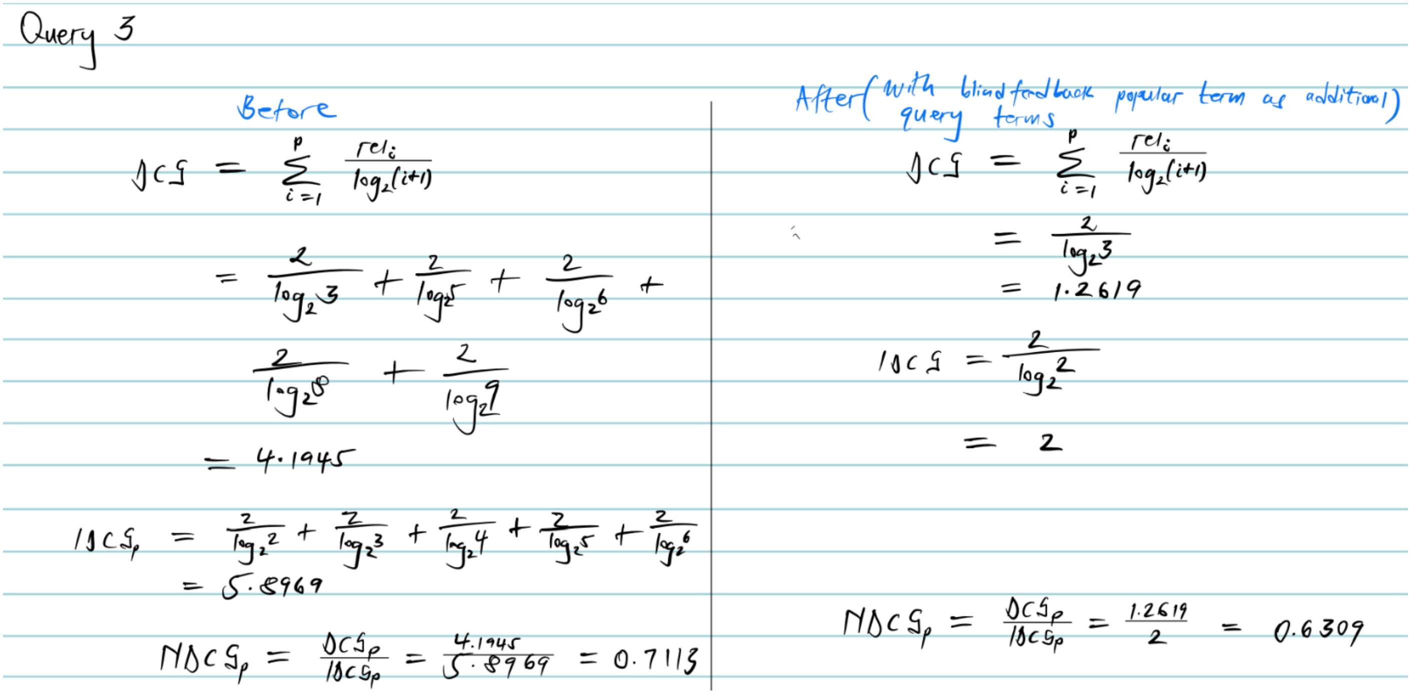
MAP = (1)/1 = 1



Recall/Precision: Popular Terms as additional query terms

| Document | Relevance | Precision | Recall | Relevance Score |
|---|---|---|---|---|
| 1 | 0 | 0 | 0/10 | 0 |
| 2 | Y | 1/2 | 1/10 | 2 |
| 3 | 0 | 1/3 | 1/10 | 0 |
| 4 | 0 | 1/4 | 1/10 | 0 |
| 5 | 0 | 1/5 | 1/10 | 0 |
| 6 | 0 | 1/6 | 1/10 | 0 |
| 7 | 0 | 1/7 | 1/10 | 0 |
| 8 | 0 | 1/8 | 1/10 | 0 |
| 9 | 0 | 1/9 | 1/10 | 0 |
| 10 | 0 | 1/10 | 1/10 | 0 |

MAP = (1/2)/1 = 0.5



Query 3

Before

$$DCG = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

$$= \frac{2}{\log_2 3} + \frac{2}{\log_2 5} + \frac{2}{\log_2 6} + \frac{2}{\log_2 8} + \frac{2}{\log_2 9}$$

$$= 4.1945$$

$$IDCG_p = \frac{2}{\log_2 2} + \frac{2}{\log_2 3} + \frac{2}{\log_2 4} + \frac{2}{\log_2 5} + \frac{2}{\log_2 6}$$

$$= 5.8969$$

$$NDCG_p = \frac{DCG_p}{IDCG_p} = \frac{4.1945}{5.8969} = 0.7113$$

After (with blind feedback popular term as additional query terms)

$$DCG = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

$$= \frac{2}{\log_2 3}$$

$$= 1.2619$$

$$IDCG = \frac{2}{\log_2 2}$$

$$= 2$$

$$NDCG_p = \frac{DCG_p}{IDCG_p} = \frac{1.2619}{2} = 0.6309$$