SHORT QUERY

| Evaluation | EasySearch | Vector Space Model | BM25 | Language Model with Dirichlet Smoothing | Language Model with Jelinek Mercer Smoothing |
|---|---|---|---|---|---|
| P@5 | 0.2000 | 0.4000 | 0.6000 | 0.6000 | 0.4000 |
| P@10 | 0.1000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 |
| P@20 | 0.1000 | 0.4000 | 0.3000 | 0.3500 | 0.2500 |
| P@100 | 0.0600 | 0.0190 | 0.1000 | 0.0900 | 0.1000 |
| Recall@5 | 0.0323 | 0.0645 | 0.0968 | 0.0968 | 0.0645 |
| Recall@10 | 0.0323 | 0.1613 | 0.1613 | 0.1613 | 0.1613 |
| Recall@20 | 0.0645 | 0.2581 | 0.1935 | 0.2258 | 0.1613 |
| Recall@100 | 0.1935 | 0.2903 | 0.3226 | 0.2903 | 0.3226 |
| MAP | 0.0634 | 0.1833 | 0.1894 | 0.1404 | 0.1462 |
| MRR | 1.0000 | 1.0000 | 1.0000 | 0.5000 | 1.0000 |
| NDCG@5 | 0.3392 | 0.5531 | 0.7227 | 0.4913 | 0.0645 |
| NDCG@10 | 0.2201 | 0.5801 | 0.6208 | 0.4666 | 0.1613 |
| NDCG@20 | 0.1793 | 0.4786 | 0.4341 | 0.3704 | 0.1613 |
| NDCG@100 | 0.2112 | 0.3804 | 0.4036 | 0.3180 | 0.3226 |

LONG QUERY

| Evaluation | EasySearch | Vector Space Model | BM25 | Language Model with Dirichlet Smoothing | Language Model with Jelinek Mercer Smoothing |
|---|---|---|---|---|---|
| P@5 | 0.0000 | 0.2000 | 0.6000 | 0.0000 | 0.2000 |
| P@10 | 0.0000 | 0.4000 | 0.3000 | 0.3000 | 0.4000 |
| P@20 | 0.1500 | 0.3000 | 0.3000 | 0.3000 | 0.2500 |
| P@100 | 0.0400 | 0.1000 | 0.1000 | 0.1100 | 0.1000 |
| Recall@5 | 0.0000 | 0.0323 | 0.0968 | 0.0000 | 0.0323 |
| Recall@10 | 0.0000 | 0.1900 | 0.0968 | 0.0968 | 0.1290 |
| Recall@20 | 0.0323 | 0.1935 | 0.1935 | 0.1935 | 0.1613 |
| Recall@100 | 0.1290 | 0.3226 | 0.3226 | 0.3548 | 0.3226 |
| MAP | 0.0214 | 0.1074 | 0.1409 | 0.0851 | 0.0961 |
| MRR | 0.0833 | 0.5000 | 1.0000 | 0.1429 | 0.2000 |
| NDCG@5 | 0.0000 | 0.2140 | 0.6399 | 0.0000 | 0.1312 |
| NDCG@10 | 0.0000 | 0.3453 | 0.4153 | 0.2064 | 0.3032 |
| NDCG@20 | 0.1066 | 0.2936 | 0.3783 | 0.2027 | 0.2312 |
| NDCG@100 | 0.0982 | 0.2968 | 0.3576 | 0.2713 | 0.2720 |

Summary of Findings:

In EasySearch implementation, I extracted short and long queries from the TREC Topic document and ranked all the documents retrieved from the index based on the traditional TF-IDF formula. But as seen in the evaluation chart, traditional TF-IDF formula is not that efficient as the Precision and Recall across all the top K (5, 10, 20, 100) is lower than the precision recall of rest of the algorithms across all the top K documents. This algorithm will perform terribly in the IR system.

It is also observed that the length of the document is compressed while indexing by normalizing it and decompressed while calculating the TF-IDF. Normalizing the documents length will result it change of the document length while converting it back to actual length but the variation is in decimals so the damage will be negligible. The main intention behind normalizing is decreasing the size of the index.

BM25 outperforms all the algorithms across almost all the evaluation parameters and hence it is better. Here, we have used the free parameters with value $k_1 = 1.2$ and $b = 0.75$ [1]

Language Model with Dirichlet Smoothing (LMD) and Language Model with Jelinek Mercer Smoothing (LMJ) adds a smoothing factor in order to avoid assignment of 0 value to probability of terms it has not seen. As it is evident from the recall and precision of LMD and LMJ, values are higher for LMD and LMJ than the precision and recall for EasySearch. Thus, the overall efficiency improves due to smoothing. We have used lambda = 0.7 for LMJ.

Reference:

1. https://lucene.apache.org/core/6_5_0/core/org/apache/lucene/search/similarities/BM25Similarity.html