Assignment 2: Retrieval Algorithm and Evaluation

Z534: Search, Fall 2017

Please compare the different search algorithms (files generated in task2 and task 3) and finish the following table:

Short query

Evaluation metric	Your algorithm	Vector Space Model	BM25	Language Model with Dirichlet Smoothing	Language Model with Jelinek Mercer Smoothing
P@5	0.1440	0.2920	0.3040	0.3480	0.2840
P@10	0.1480	0.3020	0.3000	0.3300	0.2820
P@20	0.1330	0.2640	0.2700	0.2890	0.2470
P@100	0.0982	0.1642	0.1674	0.1690	0.1600
Recall@5	0.0233	0.0529	0.0478	0.0621	0.0524
Recall@10	0.0551	0.0961	0.0879	0.1018	0.0913
Recall@20	0.0828	0.1451	0.1377	0.1460	0.1337
Recall@100	0.2244	0.3556	0.3557	0.3452	0.3300
MAP	0.1049	0.1990	0.2011	0.2072	0.1942
MRR	0.2855	0.4798	0.4778	0.4786	0.4542
NDCG@5	0.1598	0.3107	0.3212	0.3520	0.3015
NDCG@10	0.1629	0.3194	0.3196	0.3448	0.3010
NDCG@20	0.1603	0.3081	0.3115	0.3329	0.2910
NDCG@100	0.1930	0.3215	0.3251	0.3309	0.3108

Long query

Evaluation metric	Your algorithm	Vector Space Model	BM25	Language Model with Dirichlet Smoothing	Language Model with Jelinek Mercer Smoothing
P@5	0.1240	0.2560	0.2840	0.2560	0.2320

P@10	0.1200	0.2440	0.2440	0.2420	0.2140
P@20	0.1140	0.2210	0.2340	0.2340	0.2120
P@100	0.0728	0.1406	0.1488	0.1460	0.1378
Recall@5	0.0185	0.0349	0.0402	0.0403	0.0406
Recall@10	0.0363	0.0621	0.0703	0.0708	0.0658
Recall@20	0.0595	0.1064	0.1159	0.1270	0.1136
Recall@100	0.1631	0.2929	0.3167	0.3340	0.2901
MAP	0.0652	0.1529	0.1676	0.1586	0.1514
MRR	0.2477	0.4528	0.4597	0.3475	0.3640
NDCG@5	0.1340	0.2819	0.3029	0.2499	0.2348
NDCG@10	0.1299	0.2682	0.2729	0.2473	0.2294
NDCG@20	0.1299	0.2586	0.2718	0.2590	0.2410
NDCG@100	0.1422	0.2694	0.2872	0.2753	0.2609

Please summarize your findings of this task:

Task 1
Implemented a search algorithm using TF-IDF ranking function.
Query is parsed using a standard analyzer into query terms. N represents the total number of documents in corpus AP89. Term frequency in documents c(t, doc) is computed for each query term and stored in a hash map. Document frequency (k) is computed i.e count of documents containing the query term. IDF is computed as log(1 + (N/k)).

ClassicSimilarity is the default scoring implementation which encodes norm values as a single byte before being stored. At search time, the norm byte value is read from the index directory and decoded back to a float norm value. This encoding/decoding, while reducing index size, comes with the price of precision loss.

Normalized length for each document is computed as (1/sqrt(number of tokens)) and document length as 1 / (normalizedDocLeng * normalizedDocLeng). We get the term frequency from its postings and calculate the relevance score for each query term and relevance score for the document given the query.

Task 2
Involved testing the performance of search function from Task 1 with TREC topics (topics.51-100). Two fields in the TREC topics <title>, <desc> were used as short, long queries respectively. Top 1000 search results were written to a result file.

Task 3
Involved using Lucene API to test the ranking algorithms: Vector space model, BM25, Language model with Dirichlet smoothing and Language model with JM smoothing. The algorithms performance is compared using long (<desc> field), short queries (<title> field) from TREC topics.

Task 4
Used trec_eval to compare the performance of the retrieval algorithms based on the evaluation metrics: precision, recall and MAP.

Evaluation metrics
Precision and recall are set-based measures. They evaluate the quality of an unordered set of retrieved documents. Precision is the fraction of retrieved documents that are relevant. Recall is the fraction of relevant documents that are retrieved. Typically, as recall increases, precision drops.

For web surfers, high precision is more important than recall as they would be interested in getting most relevant results in the first page but might not be interested in looking at every relevant result. For analysts, recall is more important than precision.

MAP is the mean of the precision scores for a single query after each relevant document is retrieved. Normalized Discounted Cumulative Gain (NDCG) considers the position of the document in the result set (graded relevance) to measure gain or usefulness.

Observations
If we compare the results based on average precision, it is observed that BM25, LM show similar increase in performance for short, long queries. It is observed that Language model always does better than the tf-idf model but significant gains are observed at higher levels of recall.

For short query, based on Precision at 5, 10, 20, 100 and MAP, NDCG evaluation metrics performance is observed to be better with Language Model with Dirichlet Smoothing followed by BM25 compared to other retrieval algorithms.

For long query, based on precision, MAP, NDCG evaluation metrics BM25 and LM performance is observed to be better compared to other retrieval algorithms.

BM25 is a bag of words retrieval function, it scales based on term frequency and document length.

For the Language model, Smoothing involves adjusting the maximum likelihood estimator to compensate for data sparseness. Retrieval performance is generally sensitive to the smoothing parameters, but also the sensitivity pattern is dependent on the query type, with performance being more sensitive to smoothing for long queries than for short queries. Long queries generally require more aggressive smoothing to achieve optimal performance.

With Jelinek-Mercer smoothing, documents that match more query terms will be ranked higher than those that match fewer terms. Smoothing helps avoid zero probabilities and also helps to improve search accuracy as observed.

References

https://lucene.apache.org/core/5_4_0/core/org/apache/lucene/search/similarities/ClassicSimilarity.html
https://galton.uchicago.edu/~lafferty/pdf/smooth-tois.pdf