Aishwarya Dhage
adhage@iu.edu
Date-10/24/2017

Assignment 2 – Report

Findings-

Short Query-

Evaluation Metric	Your Algorithm	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	0.0900	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	0.5531
NDCG@10	0.2201	0.5801	0.6208	0.4666	0.5704
NDCG@20	0.1793	0.4786	0.4341	0.3704	0.3681
NDCG@100	0.2112	0.3804	0.4036	0.3180	0.3726

Long Query-

Evaluation Metric	Your Algorithm	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	0.3000	0.3000	0.2000	0.3000
P@20	0.1500	0.2500	0.3000	0.2500	0.2500
P@100	0.0400	0.1000	0.1000	0.1000	0.0900
Recall@5	0.0000	0.0323	0.0968	0.0000	0.0323
Recall@10	0.0000	0.0968	0.0968	0.0645	0.0968
Recall@20	0.0968	0.1613	0.1935	0.1613	0.1613
Recall@100	0.1290	0.3226	0.3226	0.3226	0.2903
MAP	0.021	0.0966	0.1333	0.0693	0.0853
MRR	0.0833	0.5000	1.0000	0.1250	0.2000
NDCG@5	0.0000	0.2140	0.6399	0.0000	0.1312
NDCG@10	0.0000	0.2785	0.4153	0.1357	0.2369
NDCG@20	0.106	0.2545	0.3732	0.1924	0.2259
NDCG@100	0.0982	0.2881	0.3519	0.2399	0.2478

Summary of findings-

Task 2-

In our algorithm, we have used the traditional tf*idf to calculate the relevance score of a query in a document.tf*idf will give us the relative importance of a term in a document. In our algorithm we have calculated ”normvalues” which will give importance of query according to length of the document. We have used ClassicSimilarity which is default scoring implementation.

Task 3-

We are using inbuilt function- TopDocs to calculate relevance score of each query, it points to top n search results which contains the queries. Scoredoc[] gives top hits of queries.

First model we have used is the classicsimilarity. It uses tf*idf which is a common vector space model. Default scoring implementation
which encodes norm values as a single byte before being stored. We decode norm values during search when we read from index directory. This norm value decoding and encoding comes with price of precision loss.Compression of norm values to a single byte saves memory at search time, because once a field is referenced at search time, its norms - for all documents - are maintained in memory. [1]

Second model used is BM25 it stands for “Best Match 25”. It is a probabilistic model. We can see in the above table as well that it gives a better precision and recall compared to other algorithms. [3]

Smoothings-MLE may overfit the data that is, it will assign 0 probabilities to words it hasn't seen. Moreover, Maximum likelihood estimate will under estimate when counts of terms is non-zero but still very small. For this solution is some variations in language models such as Smoothing. This improves the efficiency of a model.

In this task we have seen two smoothing models.
1)Jelinek-Mercer method-This involves linear interpolation of MLE with collection model. We have used a coefficient λ of value 0.7. It works better for long queries than Dirichlet Smoothing. [2]

$$p_\lambda(w \mid d) \;=\; (1-\lambda)\, p_{ml}(w \mid d) + \lambda\, p(w \mid \mathcal{C})$$

2)Dirichlet Smoothing- It is a Bayesian smoothing model which works well for short queries than Jelinek-Mercer. It considers document length parameter (μ which can be tuned).[2]

$$p_\mu(w \mid d) \;=\; \frac{c(w;d) + \mu\, p(w \mid \mathcal{C})}{\sum_{w' \in V} c(w';d) + \mu}$$

References-

[1]https://lucene.apache.org/core/5_4_0/core/org/apache/lucene/search/similarities/ClassicSimilarity.html

[2] http://www.stat.uchicago.edu/~lafferty/pdf/smooth-tois.pdf

[3] http://opensourceconnections.com/blog/2015/10/16/bm25-the-next-generation-of-lucene-relevation/