Name: Akshay Naik
Email ID: aunaik@iu.edu

ILS Z534 (Search) – Assignment 2

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	0.0631	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	0.1804	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	0.0210	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	0.1060	0.2545	0.3732	0.1924	0.2259
NDCG@100	0.0982	0.2881	0.3519	0.2399	0.2478

Name: Akshay Naik
Email ID: aunaik@iu.edu

Summary of Findings:

In searchTRECtopic.java I have used Classic Similarity and I am not using the inbuilt function like topDocs in implementation. We are calculating the score of each document for a given query using:

$$F(q, doc) = \sum_{t \in q} \frac{c(t,doc)}{length(doc)} * \log\left(1 + \frac{N}{k(t)}\right)$$

As expected it doesn't give good performance as compared to other models provided by Lucene.

DefaultSimilarity(ClassicSimilarity) is based on highly optimized vector space model. The default scoring encodes norm values as a single byte before being stored. This compression saves memory at search time because once a field is referred at search time, its norms are maintained in memory. The rationale supporting such lossy compression of norm values is that given the difficulty and inaccuracy of users to express their true information need by a query, only big difference matter. As it performs vector operations by considering each document as a document vector and query as query vector, it take huge amount of time to execute as compared to other models.

BM25 is a probabilistic model. In BM25, k is useful for modifying the impact of TF and b allows us to finely tune how much the document length influences the scoring.

$$\text{Score: } IDF * ((k + 1) * tf) / (k * (1.0 - b + b * (|d|/avgDl)) + tf)$$

Default values for k and b are 1.2 and 0.75 respectively. BM25 gives one of the best performances.

In language model, smoothing is used to assign probability value to words not present in a document. Thus, smoothing prevents zero probabilities. They also try to improve the accuracy of the model as a whole. In Jelinek-Mercer smoothing, a parameter λ is used to tune the smoothing factor which is generally kept as 0.6 or 0.7. In Dirichlet smoothing μ is used as smoothing factor. Smoothing depends on the sample size.It is quite clear from the table above that Dirichlet smoothing gives better result for short queries whereas Jelinek-Mercer gives better results for long queries.