Assignment 2

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	1.0	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

Assignment 2

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	0.4000	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	0.1290	0.0968	0.0968	0.1290
Recall@20	0.0968	0.1935	0.1935	0.1935	0.1613
Recall@100	0.1290	0.3226	0.3226	0.3548	0.3226
MAP	0.0213	0.1074	0.1404	0.0851	0.0961
MRR	0.0833	0.5000	1.0	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	0.1060	0.2936	0.3783	0.2375	0.2312
NDCG@100	0.0982	0.2968	0.3576	0.2713	0.2720

Assignment 2

Summary:

In our algorithm, we used classic similarity to decode normalized document length. In this algorithm, we calculated the frequency of query in document using normalized TF and IDF.
We also used ClassicSimilarity for vector space model.Classic similarity 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 - it is not guaranteed that decode(encode(x)) = x. For instance, decode(encode(0.89)) = 0.875.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. 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 differences matter. [1]
BM25 ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document.BM25 algorithm is implemented using k1 = 1.2 and b = 0.75.BM25 algorithm calculates the score by scaling term frequency by k and document length by k and b.
Smoothing technique is used to calculate probabilities of unseen or missing words in language model. Some smoothing techniques are lowering the probability estimates for words that are seen in the document text or assigning that remaining probability to the estimates for the words that are not seen in the text. Language Model with Jelinek Mercer Smoothing uses fixed coefficient to smooth data. In this case, we use 0.7. Language Model with Dirichlet Smoothing uses document length to calculate the coefficient. In case of vector space model, we find similarity on the basis of geometry. Language model is based on probability.

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