Saniya Ambavanekar 2000275390 sambavan

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	0.144	0.292	0.304	0.348	0.284
P@10	0.148	0.302	0.3	0.33	0.282
P@20	0.133	0.264	0.27	0.289	0.247
P@100	0.0982	0.1642	0.1674	0.169	0.16
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	0.146	0.1337
Recall@100	0.2244	0.3556	0.3557	0.3452	0.33
MAP	0.1049	0.199	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	0.352	0.3015
NDCG@10	0.1629	0.3194	0.3196	0.3448	0.301
NDCG@20	0.1603	0.3081	0.3115	0.3329	0.291
NDCG@100	0.193	0.3215	0.3251	0.3309	0.3108

Long Query					
Evaluation metric	Your algorithm	Vector Space	BM25	Language Model DM	Language Model JM
P@5	0.124	0.256	0.284	0.256	0.232
P@10	0.118	0.244	0.244	0.242	0.214
P@20	0.114	0.221	0.234	0.234	0.212
P@100	0.0722	0.1406	0.1488	0.146	0.1378
Recall@5	0.0185	0.0349	0.0402	0.0403	0.0406
Recall@10	0.0361	0.0621	0.0703	0.0708	0.0658
Recall@20	0.0595	0.1064	0.1159	0.127	0.1136
Recall@100	0.1602	0.2929	0.3167	0.334	0.2901
MAP	0.0648	0.1529	0.1676	0.1586	0.1514
MRR	0.2475	0.4528	0.4597	0.3475	0.364
NDCG@5	0.134	0.3107	0.3029	0.2499	0.2348
NDCG@10	0.1286	0.3194	0.2729	0.2473	0.2294
NDCG@20	0.1298	0.3081	0.2718	0.259	0.241
NDCG@100	0.1409	0.3215	0.2872	0.2753	0.2609

Findings:

1) **My Algorithm:**
    - In easySearch we have implemented the tf-idf scoring technique to find relevant documents and get the ranking of them.
    - Term Frequency (tf) indicates how often does a term occur in the document.
    - Inverse document frequency(idf) indicates how many documents a term appears in.
    - Hence the relevance score is tf *idf.
    - We have introduced the length of the document as the normalization technique which gives significant bias towards matching the shorter documents as compared to the longer.
    - This means suppose a term occurs twice in longer document and same term appears twice in shorter document then the shorter document is given more importance than the longer.

2) **Classic Similarity:**
    - It is the default similarity lucene provides. It gets the top k relevant documents by using vector space model.
    - Each document is represented as real valued vectors of tf idf weights.
    - A cosine similarity is used to calculate similarity between the query vector and document vector.
    - As we can infer from the above reading, lucene's classic similarity is giving more accurate precision as compared to my algorithm
    - Lucene's vector space model uses the same tf-idf technique to calculate the relevance score.

3) **BM25 Similarity:**
    - BM25 stands for Best Match 25 Similarity.
    - It is the probabilistic information retrieval model.
    - The idf calculation of BM25 is similar to the IDF calculation of classic similarity.
    - The difference is in the tf calculation. BM25similarity either takes a "k" value which tunes the tf and helps to retrieve more relevant documents.
    -

4) **Language model using Dirichlet Smoothing:**

- This is a “generative model” which considers probability distribution of strings over text.
- Using these probabilities, it uses likelihood technique to calculate the similarity.
- Smoothing is a technique which is used to estimate probabilities of missing or unseen words.
- The smoothing factor here depends on the document length.

5) Language model using Jelinek Mercer Smoothing:

- This is also a “generative model” which considers probability distribution of strings in a given text.
- Using these probabilities, it uses likelihood technique to calculate the similarity.
- The smoothing factor here is provided in terms of lambda.