ILS-Z534 - Assignment 3: Feedback and Query Expansion

Prahasan Gadugu

Task 1: Rocchio algorithm for relevance feedback

Table 1: Rocchio algorithm parameter value comparison (we fix α = 1, and please use F1 score):

	β = 0.2	0.4	0.6	0.8	1
γ = 0	0.2111	0.2111	0.2121	0.2202	0.2196
0.2	0.1751	0.1859	0.1864	0.2300	0.1706
0.4	0.2020	0.2299	0.2078	0.2474	0.1894
0.6	0.2074	0.2370	0.2330	0.2684	0.2030
0.8	0.2306	0.2524	0.2481	0.2848	0.2074
1	0.2518	0.2774	0.2496	0.2842	0.2342

The best performed value setting is with the one that has more F1 score, as it means that the Precision and Recall are more balanced. So as per the table mentioned for my values the F1 score is high for the setting alpha = 1, beta = 0.8 and gamma = 0.8. The reason for this might be the corpus I have been given has lot of judged documents. Generally, the number of relevant documents and the non-relevant documents maintain the balance and as we are fixing alpha, we are calculating the distance from the original query and training the algorithm on what is relevant and what's not relevant.

Table 2: Compared Rocchio algorithm performance (with the best parameter setting) with vector space model w.r.t. precision, recall, F1, MAP, and NDCG. Please explain the reason why.

Evaluation metric	Vector Space Model	Rochhio Algorithm (best parameter setting)	Rochhio Algorithm (best F1 value setting)
P@5	0.2959	0.3080	0.621
P@10	0.3019	0.2580	0.4360
P@20	0.2600	0.1680	0.2940
P@100	0.1648	0.0552	0.1128
Recall@5	0.0539	0.1113	0.2023
Recall@10	0.0960	0.1634	0.2491
Recall@20	0.1416	0.1900	0.2768
Recall@100	0.3578	0.2517	0.3426
MAP	0.1975	0.1302	0.2213
NDCG@5	0.3116	0.3406	0.6309
NDCG@10	0.3183	0.3192	0.5306
NDCG@20	0.3043	0.2710	0.4530
NDCG@100	0.3213	0.2434	0.3738

- A closer look at the table shows that the recall increases gradually from @5 to @100. Recall should 1 when all documents are retrieved without taking relevant score into consideration.
- On the contrary, the precision values drop gradually.
- We know that Precision = True Positives/(True Positives + False Positives)
- Therefore, as we retrieve more documents, precision value should drop since it will penalize misclassified documents with each result.
- Mean Average Precision (MAP) calculates the performances of relevancy tests over several queries. In this case, MAP has been calculated for 20 queries. Precision with just one query is equivalent to MAP for just 1 query.
- NDCG calculates the ratio of Discounted Cumulative Gain to the Ideal Discounted Cumulative Gain.
- In this case, NDCG is also calculated for 20 queries. Therefore, if the ideal ranks match with our model ranks, the NDCG will be 1.
- My model shows that NDCG is between range 0.1-0.2 meaning that our model is at an offset of 0.2 from the ideal scenario.
- The same trends we can find for the Vector space best parameter setting and for the best F1 setting as well.

Task 2 implementation: Feedback terms filtering

So, for effectively choosing the feed back terms to following tasks have been performed,

- Initially, in Task 1 I have used a Simple Analyzer as it breaks text into terms whenever it encounters a character which is not a letter. All terms are lower cased. Hence it takes into consideration even the noisy terms into account.
- But in the Task 2, I have used the Standard Analyzer to tokenize the terms and included a filter of Stop Analyzer with the following Lucene functionality "StopAnalyzer.***ENGLISH_STOP_WORDS_SET***"

- In addition to that, as I have observed some prepositions are not in the English word stop set mentioned above, I have taken the IDF values for which the following mathematical formula is used,

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$$

- Here N is the total number of documents in the collection and the denominator in the logarithmic function df refers to the document frequency of the term, that is the number of times the term t appears in the document.

- The IDF of a rare term is high, whereas the IDF of a frequent term is likely to be low, hence this approach is followed.

- After calculating the IDF, the terms like “have”, “the” are eliminated by excluding the low IDF valued terms.

- So here the formula I propose is IDF + “English Stop words Lucene list” +Standard Analyzer consideration to better filter the feedback terms.

  The Task 2 model table of F1 scores is as follows,

	β = 0.2	0.4	0.6	0.8	1
γ = 0	0.2244	0.2244	0.2244	0.2244	0.3434
0.2	0.3540	0.1902	0.1932	0.1856	0.1718
0.4	0.2510	0.3486	0.2158	0.2076	0.1962
0.6	0.2650	0.2380	02314	0.2844	0.2116
0.8	0.2830	0.2546	0.2482	0.2320	0.2094
1	0.2130	0.2680	0.2472	0.2491	0.2330

Compared to the values in Task 1 model, the Task 2 model have significantly better F1 score values as I have filtered the noisy terms from the feedback. Here the values for which the F1 score is better is for beta = 0.8 and gamma = 0.6.