Your Name: Han Zhang

Your Andrew ID: hanzhan2

Homework 1

Collaboration and Originality

Your report must include answers to the following questions:

1. Did you receive help of any kind from anyone in developing your software for this assignment (Yes or No)? It is not necessary to describe discussions with the instructor or TAs.

   If you answered Yes, provide the name(s) of anyone who provided help, and describe the type of help that you received.

   Yifan Pan. I discussed the logic of some part of codes provided by instructor.

2. Did you give help of any kind to anyone in developing their software for this assignment (Yes or No)?

   If you answered Yes, provide the name(s) of anyone that you helped, and describe the type of help that you provided.

   No.

3. Are you the author of every line of source code submitted for this assignment (Yes or No)? It is not necessary to mention software provided by the instructor.
   Yes.
   If you answered No:
   a. identify the software that you did not write,
   b. explain where it came from, and
   c. explain why you used it.

4. Are you the author of every word of your report (Yes or No)?
   Yes.
   If you answered No:
   a. identify the text that you did not write,
   b. explain where it came from, and
   c. explain why you used it.

Your Name: Han Zhang

Your Andrew ID: hanzhan2

Homework 1

1 Structured query set

1.1 Summary of query structuring strategies

First, use #NEAR/n to retrieval for special term combination. For the majority of the special multiple-word terms, #NEAR/n could be applied to combine them into one in the special order.

Second, highlight keys with changing field. The key words that matter the most in a query could be reinforced by focusing on their occurrences in title field. Some words could be really important in a query or be too special to indicate the industry or field. Focusing documents with titles containing these words helps to improve the performance. Meanwhile, for some multiple-word phrases, if one or several words in the phrase have alternatives, they will not be highlighted. Otherwise the amount of return may be restricted a lot.

Third, decrease the important levels of frequent words. Some query may include some very frequent words that have much less important level. These world's importance could be decreased by the combination of #OR and keywords field. Since people seldom put frequent and general words into keywords, seeking for their scores as keywords will return a small size documents with low scores. Then combining this part with the rest by #OR, the effect of the general words will be alleviated.

Last, for most of queries, #AND is generally better than #OR. This is because that this time we focus on the top K(100) results, and we prefer the precision more. Based on the natural characters of #AND and #OR, #AND performs better. Meanwhile, for many small sets of words, #AND is good enough.

1.2 Structured queries

718:#AND(Controlling acid.title rain)

The second highlighting key word strategy and the last #AND strategy are applied. Acid is the key word here while rain has some alternatives.

719:#AND(Cruise ship damage sea life)

The last general strategy is applied. Since there are no special terms and many words have alternatives.

724:#AND(Iran Contra)

The last strategy is applied. These two words are equally important.

725:#AND(Low #NEAR/3(white blood cell) count)

The first special term strategy and the last general strategy are applied. White blood cell is a very special phrase here.

733:#AND(Airline overbooking)

The last strategy is applied. These two words are equally important.

734:#AND(Recycling.title successes)

The second highlighting key word strategy and the last #AND strategy are applied. Recycling is the key word.

735:#OR(#NEAR/10(Afghan women) condition.keywords)

The third frequent word alleviation is applied here. Condition is not very important and has many alternatives.

741:#NEAR/10(Artificial.title Intelligence.title)

The first special phrase and the second highlighting keys strategies are applied here. Artificial Intelligence is a very special phrase.

744:#OR(#AND(Counterfeit ID) punishments.keywords)

The third frequent word alleviation is applied here. Punishment is a frequent word and has many alternatives.

746:#AND(Outsource job India)

The last strategy is applied. These two words are equally important.

2 Experimental results

2.1 Unranked Boolean

	BOW #OR	**BOW #AND**	**Structured**
P@10	0.0000	0.2000	0.3600
P@20	0.0000	0.2250	0.3300
P@30	0.0033	0.2367	0.3467
MAP	0.0002	0.0489	0.0720
Running Time	00:09	00:02	00:01

2.2 Ranked Boolean

	BOW #OR	**BOW #AND**	**Structured**
P@10	0.0400	0.3800	0.5600
P@20	0.0800	0.3650	0.5100
P@30	0.0867	0.3300	0.4667
MAP	0.0079	0.0871	0.1197
Running Time	00:09	00:02	00:01

3 Analysis of results: Query operators and fields

#OR operator returns a big score list generated from all arguments. It improves recall but reduces the precision. It works well to combine similar words and take the higher score for the same document. That is why it helps to realize the third strategy, decreasing the important levels of frequent words and focusing on the important part. I try to decrease the scores of frequent words with specifying the field and take the scores of higher and more relevant query terms. However, #OR is not good for multiple-word phrases. Multiple-word phrases usually make much more sense as the whole. #OR separates each word during evaluation. What's more, #OR doesn't satisfy my expectations here in this experiment since the recall of #OR is not high. I think this is because #OR changes the distribution of all relevant documents and could not make sure the relevant document amount in the top N. That is also part of reason why my last #AND strategy would work.

#AND operator returns the overlapped part of score lists and take the lower score for the same document. It improves precision but reduces recall. Its advantage is to consider all terms. This is where the last general #AND strategy comes from. For the top N retrieval, precision is more significant. However, its disadvantage is treating all arguments equally while a query generally has its own focus.

#NEAR/n operator returns the score list that reflects the ordered terms within a given distance. Its advantage is to retrieval the terms combination. This is where the first strategy comes from. A special phrase should not be treated separately as terms. Searching for a phrase will improve the precision. While #NEAR/n's disadvantage is that the distance is not easy to choose, and all arguments must follow the given order. These are strict restrictions and sometimes these filters some relevant documents out.

Usage of fields may require the knowledge of the target field and the articles' structures, which is not applicable in the daily use of retrieval, but great for professional users. Title, URL and keywords could be used for stressing the key words. They are strict restrictions. Misusage of field could get few results. Meanwhile, the effectiveness of fields also depends on the quality of dataset. It is hard to know whether URL are well structured, and whether all articles have clear keywords or not. Body is the default and the safe one to use. But it is also not a strict restriction. The advantage of using field is that it helps a lot if being well used, while their disadvantage is that it is hard to use them well.

4 Analysis of results: Queries and ranking algorithms

Generally, #AND performs better than #OR while Structured query performs better than #AND on P@k, MAP and running time. Ranked Boolean performs better than Unranked Boolean.

First, #AND performs better than #OR on precision and running time. This consists with the discussion above. #AND is a stricter operator, so it could return more relevant results and score them with their lowest relevance to all terms. Meanwhile, since this is a top N retrieval, #OR doesn't show its advantage on recall. #AND performances better. As for the running time, For the same terms, #OR generates a much longer score list than #AND does. It is costlier to maintain and manipulate a longer array. So even for the same operation of the same time complexity, such as O(logN), a big N requires more time that a small N.

Second, structured queries perform better than #AND on precision, recall and running time. Structured queries have more restrictions and will retrieve less documents, most of which are irrelevant documents. Structured queries will reduce the false positive rate. As a result, both precision and recall will perform better. And since it retrieves less documents, the running time will perform better too.

Third, Ranked Boolean performs better on precision and recall than Unranked Boolean. The running time is similar since the operations before getscore method are the same. If we check the whole result lists, these two result lists should be the same. Since these two score systems score the document in different ways, the sorted results are different. Consequentially, truncating the whole score list at 100 will lead to totally different results. Ranked Boolean scores more relevant results higher scores, so it is more possible to have more true relevant documents in top 100. That is why Ranked Boolean performs better.