Homework 1

1 Introduction

1.1 Collaboration and Originality

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.
   Yes, from TA Yizhen Ma, I asked her to confirm my understanding of some concepts.
   If you answered Yes, provide the name(s) of anyone who provided help, and describe the type of help that you received.

2. Did you give help of any kind to anyone in developing their software for this assignment (Yes or No)?
   No.
   If you answered Yes, provide the name(s) of anyone that you helped, and describe the type of help that you provided.

3. Did you examine anyone else's software for this assignment (Yes or No)? It is not necessary to mention software provided by the instructor.
   No.
4. 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.

5. 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.

2 Structured queries

2.1 Query structuring strategies

I first search the original query terms in google, and try to understand the meaning of the query terms. Extract information like whether the two terms are synonyms, which terms are often used together, or which term is often included in url or title. After I get to know the meaning, I then try to see which result is more likely to be a fit if I were the user who typed in those terms, and analyze the characteristic of the results that are good fit to what I believe the user is looking for. Based on the characteristics observed, I then can decide how to link different terms and which field to put them in.

1. For example, use NEAR for terms that are usually show up together for the topic

2. Use SYN/OR to link NEAR when the order of the adjacent terms might be reversed

3. Use AND when the relative distance between two terms are not predictable, but both are useful information

4. Use TITLE field if it's predictable that the results with searching terms on title will have high relevance

5. Use AND when adding any other operator seems redundant and result in lower scores

How to optimize queries with low MAP score based on result:

6. If num_relevant is high but num_ret_rel is low, try to relax the operator i.e. use more SYN/OR than NEAR/AND or use BODY field than other field

7. If num_ret is high but num_ret_rel is low, try to make constraint stricter so that the ones returned are more relevant, basically the reverse action from described above.

We can also look at the distribution of P_5, P_10, etc

8. If top numbers have low values, it means probably our query is not strict enough.

9. If P_200 or such large scale have low values, it means our query rule out some cases for the results that are also relevant but not subject to the general feature we observed, need to relax it a little.

10. remove bad terms

11. use OR to connect different fields of the same terms

2.2 Queries

For Unranked Boolean

711: #AND(#NEAR/1(Train station) #AND(security measures)) Use of strategy: 1, 3

730: #AND(#NEAR/1(Gastric bypass) complications) Use of strategy: 1, 3

733: #OR(#NEAR/10(overbooking Airline) #NEAR/10(Airline overbooking)) Use of strategy: 1,2, 6

751: #NEAR/2(Scrabble Players) Use of strategy: 1, 6

758 : #NEAR/5(Embryonic stem cells) Use of strategy: 1, 7

764 :#AND(Increase mass transit use) Use of strategy: 3, 5

802: #AND(Volcano eruptions global temperature) Use of strategy: 3, 5

809: #AND(wetlands #NEAR/5(wastewater.title treatment.title)) Use of strategy: 1, 4, 8

811:#SYN(#NEAR/4(handwriting recognition) #NEAR/4(recognition handwriting)) Use of strategy:1,2

826:#AND(#SYN(#NEAR/4(Seminole Florida) #NEAR/4(Florida Seminole)) Indians)

Use of strategy: 1,2, 7

For Ranked Boolean (only show queries that are different from above)

711: #AND(#NEAR/10(Train station) #OR(#NEAR/10(security measures) #NEAR/10(security.title measures.title) security.keywords security.url)) Use of strategy: 1,2, 11

730: #AND(#OR(#NEAR/4(Gastric bypass)) #OR(complications complications.url complications.keywords complications.title)) Use of strategy: 1,2, 6, 9, 11

733: #OR(#AND(Airline overbooking) #AND(Airline.keywords overbooking.keywords)) 6,11

751: #OR(#NEAR/10(Scrabble Players) #AND(Scrabble.title Players.title)) Use of strategy: 1,2, 11

764 :#OR(#AND(Increase mass transit use) #NEAR/10(mass transit) #NEAR/10(mass.title transit.title))

802:#AND(eruptions global temperature) Use of strategy: 10

809: #AND(wetlands #OR(#NEAR/10(wastewater treatment) #NEAR/10(wastewater.title treatment.title))) Use of strategy: 1,2,11

3 Experiment: Unranked Boolean

	BOW #OR (Exp-3a)	BOW #AND (Exp-3b)	Structured (Exp-3c)
P@10	0.0100	0.0800	0.2800
P@20	0.0050	0.1200	0.2950
P@30	0.0067	0.1567	0.2933
MAP	0.0015	0.0747	0.1057
Running Time	0:5.3	0:0.7	0:0.6

4 Experiment: Ranked Boolean

	BOW #OR (Exp-4a)	BOW #AND (Exp-4b)	Structured (Exp-4c)
P@10	0.1100	0.4300	0.5800
P@20	0.1200	0.4400	0.5250
P@30	0.1133	0.4000	0.4400
MAP	0.0164	0.1542	0.1845
Running Time	0:5.4	0:0.7	0:0.7

5 Analysis of results: Ranking algorithms

The running time for queries Structured < AND < OR

Accuracy: Structured > AND > OR

Time forming queries: Structures > AND = OR

OR:

Both rankedboolean and unrankedboolean algorithm works bad for the first type of query where we only use OR, the running time is the longest and the performance is very bad for both ranking algorithms, the reason is that, OR operator alone can not work well in filtering out irrelevant information. Instead lots of results that contain only partially the query terms will be retrieved, which tend to be highly irrelevant results. The running time will be long, and accuracy would be low for the same reason, they are negatively corelated in this case due to lack of filtering, longer time leads to lower accuracy.

AND:

Both rankedboolean and unrankedboolean algorithm works fine with the second type of query where we only use AND, the running time is relatively short, and performance is OK. Surprisingly, compare to the performance difference between second type and the structured queries. Rankedboolean is better in taking queries with only AND. The reason is probably due to its success in including relevant results in a large scale.

For the unrankedboolean with AND operator, it clearly fails in identifying the best top results. The reason is that the term frequency is not taken into consideration while calculating the score, a lot of results might include each term only once and will get the same score for those contain a larger number of tf. The ranking then is determined by external ID order, which is not a relevant information to use.

For the rankedboolean with AND, even though it fails in identifying the best top results, the good score in P_30 and even larger scale give it a decent MAP score. The reason is probably that minimum tf among the query terms is a decent feature to use.

The relationship between accuracy and running time is mixed here. But generally, they are also negatively correlated, shorter time gives better accuracy.

STRUCTURED:

Due to the characteristic of unrankedboolean algorithm, which is to give the same score for selected results. It works best for structured queries, which are carefully formed so that only the relevant results would be given a score. The running time might be even lower than AND, since NEAR operator is even more strict then AND, and result in a higher accuracy. It took a long time to form the query since we need to balance the strictness of our query.

For the rankedboolean algorithm, even though we already give different scores based on tf, structured queries would help in peeking the most relevant results to the top so that MAP score is improved

6 Analysis of results: Query operators and fields

AND, NEAR tend to decrease the running time since they are harsh filtering operator, and decrease returned result numbers, SYN, OR, on the other hand have a longer running time with a larger number of returned results

SYN operator is normally good when connecting two NEAR operator which has opposite order, for example SYN (NEAR/2(a b) NEAR/2(b a))

NEAR operator is really good in finding the best match, for example, in the NEAR/2(scrabble player) case, it gives a really small num_ret, however all the returned with top scores are highly relevant, gives me a high score in P_5, P_20, etc, resulting in high MAP

Another successful use is to associate a large N with NEAR, which means the two terms are in the same sentence, rather than very close to each other. This gives me surprisingly good results for (airline overbooking) case

OR is not a very helpful operator, since if you link A, B where one of them, say A, is a more relaxed constraint, the effect of B is almost ignored. But sometimes OR can be used in place of SYN to link two NEAR operators.

For this particular assignment, URL and KEYWORDS fields are not very informative, incorporating either field will result in plunge in number of returned results, hurting our MAP score by a lot. This might due to the nature of Lucene. However, URL/KEYWORDS/TITLE should be considered as unpredictable field comparing to BODY. For example, people can be creative in designing the passage titles.

TITLE keywords sometimes can be helpful when we try to give the most relevant results high score, and generate results with high P_5, P_30, etc. Would probably harm our results for P_100 and P_300, since relevant results might not have key terms in their titles.

It is hard when there are many synonyms used in the context, for example, “global warming” and “global temperature”, when we only search one of them but expect the results of both, it would be difficult to form a good structured query, as in the case of #AND(Volcano eruptions global temperature) and #AND(Increase mass transit use).

Understanding the meaning of the terms is also crucial as in the case of #AND (#SYN(#NEAR/4(Seminole Florida) #NEAR/4(Florida Seminole)) Indians). Originally, I put Seminole and Indian as SYN based on my understanding which gives really bad results. Further investigation makes me to decide to put Florida and Seminole together.

Some terms can be appear in different fields, so it could be good to use OR to connect the different FIELDS of the same term