Your Name: Yuping Gong

Your Andrew ID: yupingg

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. No

   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. 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: Yuping Gong

Your Andrew ID: yupingg

Homework 1

1 Structured query set

1.1 Summary of query structuring strategies

1. Use AND as the default operator.

2. When the query only has one term, OR and AND both can be the default operator.

3. Try to add field when the list of terms is short.

4. If there is a long list of terms, use OR.

5. Use NEAR when the word seem to be a phrase or a compound word. Set a small distance when I am quite sure the terms are compound word.

6. If the term seems to be a file format, use field ‘url’ or ‘inlink’

7. If the term is a proper noun, try to add ‘keywords’

1.2 Structured queries

10:#OR(cheap.title internet)

Use the third strategy. For the term ‘cheap’, I add field ‘title’, considering that these kind of information often appear in the title of a document to attract people.

12:#OR(djs.url djs.inlink)

Use the first strategy, the third strategy and the sixth strategy. The term may be a format, so try to add url or inlink as the field. I use OR to combine the result from two field to improve recall.

26:#AND(lower #NEAR/3(heart rate))

Use the first strategy and the fifth strategy. Heart rate may be a phrase, so I add NEAR to combine them and use AND as the default operator.

29:#AND(#NEAR/3(ps 2) games)

Use the first strategy, the third strategy and the fifth strategy. Ps 2 may be a compound word, so I add NEAR to combine them and use AND as the default operator.

33:#NEAR/5(elliptical trainer)

Use the fifth strategy. Elliptical trainer may be a compound word, so I use NEAR to combine them. I set the distance as 5 because I am not sure whether they will be used as a compound word.

52:#OR(avp.url avp.inlink)

Use the first strategy and the third strategy. The term ‘avp’ may be a format, so I add ‘url’ or ‘inlink’ as its field.

71:#AND(living in india.keywords)

Use the first, the third and the seventh strategy. Since india is a proper noun so I try to add keywords as its field.

102:#AND(fickle creek #OR(farm.title farm.body))

Use the first strategy and the third strategy. Because the terms can hardly be a phrase or compound word, just use the default AND as the operator. And the term ‘farm’ may appear in the title so add this field to it.

149:#OR(#NEAR/3(uplift at) #NEAR/3(yellowstone national park))

Use the first and fifth strategy. ‘Yellowstone national park’ is a compound word and ‘uplift at’ is a phrase, so I use NEAR to combine them. Because there are so many terms in this query, I use OR as the operator.

190:#AND(brooks brothers clearance)

Use the first strategy. Since the terms are not related to each other and hard to identify the field, just use the default AND as the operator.

2 Experimental results

2.1 Unranked Boolean

	BOW #OR	BOW #AND	Structured
P@10	0.0100	0.0400	0.0297
P@20	0.0050	0.0200	0.0700
P@30	0.0033	0.0433	0.0950
MAP	0.0010	0.0142	0.0900
Running Time	01:13	00:04	00:06

2.2 Ranked Boolean

	BOW #OR	BOW #AND	Structured
P@10	0.1500	0.2500	0.3800
P@20	0.1800	0.2600	0.3350
P@30	0.1667	0.2767	0.3200
MAP	0.0566	0.0980	0.1212
Running Time	02:51	00:06	00:12

3 Analysis of results

1. The difference between three approaches

The running time:

According to the form above, the running time of BOW #OR is the longest and BOW #AND has shortest running time for both ranked Boolean retrieval and unranked Boolean retrieval.

In the experiment, I only output the top 100 results to do the evaluation. When I use AND to forming the query102, I got even less than 100 results, which means when I use OR to forming queries, the number of all retrieval results is greater than the results number when I use AND to forming queries. For OR operator, there are a lot of documents match the query. The program need to add them in the result and do the sort work, which costs more time than other operators. For NEAR operator, it costs more time than AND because it do all things that AND need do (finding the common document ID) and spend some time to compare the location. For the structured queries, the running time is longer than BOW #AND approach and much shorter than BOW #OR operator. Because I use AND as the default operator, the disadvantage of OR in running time does not hurt the running time a lot. And because I use NEAR and OR in some situation, the running time will become a little bit longer than the BOW #AND approach.

The performance:

The performance of AND is better than OR when I only focus on the top 100 results. Because in the limit amount of result, AND can give more precision result to satisfy the information need. Although OR can also find these documents, they were chopped and can’t be evaluate by the system. So nearly all the MAP and P@N of OR approach is lower than those of AND approach. The structured approach perform better than the other two approaches, which means part of my strategy is reasonable.

2. The difference between two retrieval approaches

Since I only evaluate the top 100 results, I assume that the purpose of query is to find the relevant documents as quickly as possible like our every day's search in Google instead of looking for the whole relevant documents about a particular law in Westlaw. So in this situation, ranked Boolean retrieval performs better than unranked Boolean retrieval. Because the ranked score of a documents make it possible for the program to find more relevant documents in the top 100 result. However, when it comes to the running time, unranked Boolean retrieval is shorter because it has no need to calculate the score which will save some time.

3. The effectiveness, strengths and weaknesses of three operators

The effectiveness of the three operators is different. The OR's effectiveness is the worst because it produces a large amount of irrelevant results. It not only costs more time to run the program but also wastes the user's time to find the information they need. But if I want to get all the retrieval results instead only part of them, the OR operator will be effectiveness to find more relevant documents to satisfy some user's need. The AND operator will be effectiveness when I am quite sure that the relevant documents should contain all the terms. If the query terms are separate words instead of compound word or phrase, I choose to use AND. If I can tell that the terms are compound word or phrase, I choose to use NEAR. The NEAR operator can be effectiveness when the terms appear in most of the documents as a compound word. But if the terms are not used as compound word in most documents, the NEAR operator will perform badly.

The strength of OR is it can find more relevant document to improve the recall. So if people want to find as many relevant document as possible, OR is the operator that need to be choose. The weakness is the running time is long and precision is low, which means we need to wait to get the results and spend time to find the relevant results. The strength of AND is I can find the document contains all the terms, which can improve the precision in most of time. However, if the query include a lot of terms, it becomes harder to find enough results. The strength of NEAR is to help us deal with compound word search. But the weakness is it hard to determine the distance to get better performance. And when I make a mistake to combine words, the results will become worse.

4. The effectiveness, strengths and weaknesses of fields

Sometimes, the precision of results can be improved if I find the right field. For example, if I change the field of query 12 to 'title', the result will perform worse than the default 'body'. But if I change the field to 'url', the performing will become better than 'body'. So the strength of field is it can enhance the precision of query and make the search more efficiency. The weakness of field is if I can't have a good strategy to choose field, the result will be even worse. Since change the field of a term indeed improves the performing of the result, I think it's effectiveness to use field in the query terms.

5. The success and failure in experiment

In the experiment, I met a problem that I found OR's recall and precision are both lower than AND in some query. For unranked Boolean retrieval, the recall = num_rel_ret/num_rel and precision = num_rel_ret/num_ret. Since the num_rel is a fixed number, OR's recall will be lower when its num_rel_ret is low. We only use the top 100 results, which makes this happen. Because OR get more results than AND, more relevant documents that OR found was chopped, which impacts the num_rel_ret. My original strategy is use both OR and AND to get a good mixture of precision and recall. But after I found these problem, I use AND as the default operator. This strategy improves the performance of the results and save a lot of running time.

I also met some problems when I use NEAR. My strategy is use NEAR when the terms seem to be a phrase or compound word. However, how can I know the word is a compound word or not and how can I decide the distance? For query ‘ps 2’, if we treat them as a compound word but they turn out to be not, the performance will be bad. I decide to use a smaller distance when I think the terms are more likely to be a compound word. And when I am not sure about these, I prefer to use larger distance. Take query 26 as an example, I get an improvement in MAP and P@10. If I am in a situation that I want to get the information I need in the first two or three pages of the results, the improvement of P@10 means I can get what I want more quickly. But this strategy to choose distance can’t perform well in every query that uses NEAR. In query 33, the MAP and P@N both become lower. These is a failure of my strategy that I can’t figure out.