Your Name: Songze Chen

Your Andrew ID: songzec

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.

   No.

   .

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.

   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.

   Yes

   .

4. Are you the author of every word of your report (Yes or No)?

   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.

   Yes.

Your Name: Songze Chen

Your Andrew ID: songzec

Homework 1

1 Structured query set

1.1 Summary of query structuring strategies

Briefly describe your strategies for creating structured queries. These should be general strategies, i.e., not specific to any particular query.

First I use Google to search the #OR queries, and I assume the first page is all relevant. Then I analyze the term I use to doing searching. For example, if a term is in the title, then I know I should use title as its field.

1.2 Structured queries

List your structured queries. For each query, provide a brief (1-2 sentences) discussion of:

1. which strategy (from Question 2.1) was used for that query,
2. any important deviations from your default strategies, and
3. your intent, i.e., why you thought that particular structure was a good choice.

69:#AND(sewing.keywords instructions)
79:voyager
84:#NEAR/2(continental plates)
89:ocd.keywords
108:#NEAR(ralph owen brewster))
141:#AND(va dmv registration)
146:#NEAR(sherwood regional library)
153:pocono.body
171:#NEAR(ron howard)
197:#OR(idaho #NEAR(state flower))

69:#AND(sewing.keywords instructions)
The information need is to search sewing instructions, not other instructions, and not sewing some specific things, so I use #AND instead of #OR, and I guess sewing might be the keywords so I also add it to “sewing”.

79:voyager
I didn’t change this query because it is a single word without particular meaning as far as I know.

84:#NEAR/2(continental plates)

I set the operator to be #NEAR because it is a combination word, but it does not have to be always literally together, so I set the distance as 2.

89:ocd.keywords
Ocd is usually an abbreviation of obsessive–compulsive disorder, and usually is used as this abbreviation instead of the whole name. So I think it could be a keyword since it might be in many medical documents, whose keyword naming system is full of abbreviations.

108:#NEAR(ralph owen brewster))
Google tells me Ralph Owen Brewster was a name of a person, and this person is different than Ralph Owen (another person), so I use #NEAR/1 operator

141:#AND(va dmv registration)
Since every state may have different policy for DMV registration, and also, we cannot drop any of the words because any two combination may has different meaning, #AND must be used.

146:#NEAR(sherwood regional library)
This is a library whose name is sherwood regional library, so people may want to see this library first when searching this term. Like Ralph Owen Brewster, it is a particular word so I use #NEAR/1.

153:pocono.body
I cannot recognize this term's particular meaning even by Google. Since it has multiple meanings on Google, so I just set its field to body.

171:#NEAR(ron howard)
The whole first page of "ron howard" is about the American director, so most people who search this word might want to get the information about this person. And his first name and last name are both common, so I use #NEAR/1.

197:#AND(idaho #NEAR(state flower))
I assume people who search this combination want to know the state flower of Idaho. So "Idaho" and "state flower" must both occur, but not necessarily near to each other.

2 Experimental results

Present the complete set of experimental results. Include the precision and running time results described above. Present these in a tabular form (see below) so that it is easy to compare the results for each algorithm.

2.1 Unranked Boolean

	BOW #OR	BOW #AND	Structured
P@10	0.0000	0.1100	0.3300

P@20	0.0150	0.1350	0.3300
P@30	0.0200	0.1533	0.2900
MAP	0.0020	0.0665	0.1600
Running Time	7874.223 ms	1206.516 ms	1403.015 ms

2.2 Ranked Boolean

	BOW #OR	**BOW #AND**	**Structured**
P@10	0.1700	0.3700	0.4500
P@20	0.2800	0.4450	0.5600
P@30	0.3367	0.4633	0.5433
MAP	0.1071	0.1882	0.2520
Running Time	8494.946 ms	1225.474 ms	1338.982 ms

3 Analysis of results: Queries and ranking algorithms

Discuss your observations about the differences between the three different approaches to forming queries, and the two different approaches to retrieving documents (i.e., retrieval models) in terms of their retrieval performance and running time.

Hint: Do not just summarize the results from the previous sections; we can see those results above. You are expected to provide your interpretation of the results based on what you learned in the lectures and readings. This is your chance to show what you learned from this homework assignment - take this section very seriously

Hint: Probably this section doesn't need to be longer than ¾ of a page (not counting these instructions).

Basically, #OR queries are appropriately takes 6 times more than #AND and my own structured.
Retrieval performance: my own structured queries > #AND queries > #OR queries.

Why #OR queries takes more time? Because #OR operation has more operations to search the minimum docID. While #AND operation just return when it finds the result. And my own structured queries has both of them, but basically combination of #AND and #NEAR, so my running time is more like #AND.

Why #OR queries have the least useful results? Because usually information needs are complex, it cannot be represented by one single word. So #OR queries generate many irrelevant document, which although cause high recall, but may and always lead to low precision. #AND queries on the contrary, has more relevant results because for each match, it takes term that has more information, which better represent the information need. My own structured queries has even better results because I analyze every query, and according the meaning in reality, I set different query operators and fields, whose details are shown in 1.2.

Why ranking algorithms have better retrieval performance? Ranking algorithms I implemented calculate how many times the specific term occurs. And naively speaking, the more times a required term occurs in a document, the document should get a higher score. (Although the reasonable weight is not linearly increasing against occurred time, at current we just count them as linearly increased). However, In my "linear" implementation, if we have an (#OR termA termB) query, if termA has occurred 2 times, and termB has occurred 0 times, it will have the same score as the situation where termA and termB both have occurred one time. But actually the second situation should has higher score.

4 Analysis of results: Query operators and fields

Discuss the effectiveness, strengths, and weaknesses of the query operators and fields, and your success and failure at using them in queries. Did they satisfy your expectations?

Hint: Same hints as above.

#OR operator has higher recall and lower precision while #AND has higher precision and lower recall. And #OR usually costs more time than #NEAR and #AND (but sometimes we need high recall for our database so we must use #OR). #NEAR may have even more higher precision and lower recall but it varies according the nature of the information needs and the term itself. For example, if i want to search the American director Ralph Owen Brewster, I believe the near operator is the best operator for high recall and precision. But if we don't use #NEAR properly, for example if we want to search "dell laptop" by #NEAR operator, we may lose both recall and precision.

The default field is body, usually using it will cost more time than other fields. So we may want to use title, keywords and other fields instead if we can. But we also obviously know if the title or other meta data that has very short content does not include the information we need, we may miss a very important result. So it is important that we take a good guess by analyzing people's habit about making document titles and other fields. And also we need to retrieve the most representative terms for our titles when we establish our own documents.

My #NEAR operator is a little tricky and naughty because it uses greedy algorithm and might gives non-optimal results and score. When the documents' score is low, which may occur if we use #NEAR, one single miss may change the whole result such as P@n.