Your Name: Aihua Peng

Your Andrew ID: aihuap

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: Aihua

Your Andrew ID: aihuap

Homework 1

1 Structured query set

1.1 Summary of query structuring strategies

The overall concern about how to structure the query is to get close to produce what people really want. Higher precisions for certain queries as well as high mean average precision are the first goal. Besides that, time complexity is another concern.

a. Use fields to specify the scope for each term. For example, some word may be meaningful in link only and some others may be better considered in title. This will also reduce time consumption.

b. Use NEAR/n when it is needed to combine/concatenate/reorder words in NEAR/n where adjacent (or reordered) words turn to be more meaningful. Treat the result as a single word for other strategies.

c. If not b, use AND when words are relatively meaningful but do not need to appear nearly. This produces more precision.

d. If not c, use OR when words do not seem to be related to each other (or could replace each other). In this scenario, for example in academic search, word 'fish' and word 'Simon' would either be important but they do not need to show up at the same time.

1.2 Structured queries

10:#AND(cheap internet)

It follows strategy c. This query may search for 'cheap' things on the 'internet' or a 'cheap internet' service. As we do not want lose possible results, I suggest use a AND as the operator and do not include in any fields.

12: #OR(djs.url)

It follows strategy a. This query may search for 'DJs' and it is an abbreviation. Such an information need would be related to an url directly and I specify that in url field.

26:#AND(lower NEAR/1(heart rate))

It follows strategy b and c. This query considers 'heart rate' as a phrase and requires it to be located in the document body. Such a structure improves precision a lot since 'heart rate' is much more meaningful than 'heart xxxxx…xxx rate'. Also, the appearance of 'lower' in this document increases the probability to show useful information.

29: #OR(#NEAR/0(ps.keyword 2.keyword) games)

It follows strategy a, b and c. In the keyword field, ‘ps’ and ‘2’ are considered to be a phrase ‘ps2’ that shows a ‘games’ oriented meaning. ‘Games’ is required to be shown at keywords such that ‘ps2’ really means Sony PS2, that giving us more precision. Also, the field keyword will save time but will not lose too much information about game.

33: #NEAR/4(elliptical trainer)

It follows strategy b. ‘elliptical trainer’ would be a phrase but we also accept ‘elliptical x x trainer’ because they both provide similar information. NEAR/4 (n=4) here do not harm precision but still keeps recall in a good level.

52: #OR(avp.inlink)

It follows strategy a. There is no clear idea about what the ‘avp’ really is since it is an abbreviation. It may stands for Alien Vs. Predator or Association of Volleyball. It is not bad to put avp in inlink field because those films or organizations are easy to be found by links.

71: #AND(#NEAR/2(living in) india.title)

It follows strategy a, b and c. The query looks for ‘india’ in the title, which I believe reasonable because the title gives us a brief idea about the document. However, I choose not to include ‘living in’ in title which may lose some really useful documents. ‘living’ and ‘in’ are combined together within a distance of 2.

102: #NEAR/1(fickle creek farm)

It follows strategy b. There makes no sense when we separate ‘fickle’, ‘creek’ and ‘farm’. That means either AND or OR will produce many unusual information. I choose NEAR/1 here because ‘fickle creek farm’ as a phrase is really meaningful.

149: #AND(uplift.title at yellowstone #NEAR/1(national park))

It follows strategy a, b, c and d. First of all, ‘national park’ is considered as a phrase. Secondly, ‘yellowstone’ needs to be shown up in the document body that provides more information about what kind of national park would be talked. Finally, uplift is constrained at title that reduces the time consumption and provides more strict information.

190: #OR(#NEAR/1(brooks brothers) clearance.keyword)

It follows strategy a, b and c. I specify ‘brooks brothers’ as a phrase which points to that clothing brand. The user may accept other clearance so OR operator is used to combine ‘brooks brothers’ and ‘clearance’. Putting ‘clearance’ in keyword saves time and also provides higher precision.

2 Experimental results

2.1 Unranked Boolean

	BOW #OR	BOW #AND	Structured
P@10	0.0100	0.0400	0.1500
P@20	0.0050	0.0200	0.1850
P@30	0.0033	0.0433	0.1767
MAP	0.0010	0.0142	0.0505
Running Time	00:15	00:04	00:04

2.2 Ranked Boolean

	BOW #OR	BOW #AND	Structured
P@10	0.1500	0.2500	0.3600
P@20	0.1800	0.2600	0.2950
P@30	0.1667	0.2767	0.2733
MAP	0.0566	0.0980	0.1055
Running Time	00:15	00:04	00:04

3 Analysis of results

3.1 Three experiments

♦ **BOW #OR –** No matter what model is, the pure OR operators on all queries, as expect, gives bad precision, long running time and good recall. It seems not useful but sometimes when the document set is small and user wants to search as much relevant information as possible, it would be significantly good to get as many as possible. It is also suitable for those scenarios where two terms are the same important, each being able to supplement the other. OR is good at 'providing more information but less accurate'.

♦ **BOW #AND –** No matter what model is, the pure AND operators on all queries, as expect, gives good precision, short running time and bad recall. In comparison with operator OR, AND cares all terms in a query, which is very meaningful for most cases in web search. When users have more need on accurate information, AND is always a good choice. For example, 'Brooks Brothers Clearance'.

♦ **Mixed –** The combinational use of OR, AND and NEAR follows the strategies in section 1. It improves precision a lot while keeping the short running time. This is because NEAR operator gives more precise and more meaningful documents in response to that query, thus getting closer to information need. Specifying fields is always helpful in maintaining good running time and sometimes provides awesome precision once the field is set correctly. That flow of strategies is also helpful in improving precision since it handles different scenario than pure OR or Pure AND.

3.2 Unranked Model & Ranked Model

- **Precision –** The chart tells that no matter what the operator is, Ranked Model provides better precision. The reason is obvious: Ranked Model ranks the document by term frequency. It is believed that the more often the terms show up, the higher confidence that this document is related to information need. Unranked Model on the other hand, also ranks the document, but by external id that does not care term frequency, meaning that all documents have the same score: 1.0. It is hard to find the real relevant documents from the result.

- **Recall –** The chart does not tell about the recall rate, but as I tested the same query for different models I would like to guess that Ranked Model also provides better recall (sometimes do not). This is not always true, but generally ranked model tries to include more related documents whereas Unranked Model may miss those documents.

- **Time –** The time for both models are nearly the same. The algorithm is totally the same for each operator. Also, they both calculate scores even though score for Unranked Model is always set to 1.0. The sorting time might be different, but just slightly.

3.3 OR, AND & NEAR/n

- **Precision -** #OR provides worst precision since it tries to include as many as possible once there is a match. It does not care about the locations, neither the combinations of terms. Many irrelative documents will be counted. #AND provides much higher precision because a result document needs to include all of the terms. This is often consistent with most information need: each word in the query is part of what people really want from the search engine. #NEAR/n gives best precision among these three. It restricts the location distance as well as the order of terms. Therefore, results given by #NEAR/n would be more meaningful and more close to what people really think.

- **Recall -** #OR provides highest recall whereas the other two are bad at this feature. The reason is obvious: OR would prefer to include as many documents as possible when there exists at least one matched terms. Thus, it is hard to miss a relevant document. #AND is not that bad if the number of terms is not huge. #NEAR/n plays the role of filter and accept only those documents, which contains a specific order of words. It may miss a lot of possible relevant ones.

- **Time -** #OR is the worst at running time. The software needs to scan all the score lists of each term. Imagine that a term ‘food’ in a food service article set would likely to show many more times that ‘fish’. Calculating ‘food or fish’ is high at time consuming since we cannot optimize anything to prevent scanning all documents to happen. #AND, on the other hand, could easily be optimized especially when ‘fish’ only maps to a small set of documents. As software scans the set, the current list is becoming smaller and smaller. The running time of #NEAR/n is similar to #AND but it is also related to what the n would be. A very small n and a very big n will both cost less time than an ‘appropriate’ n. Indeed, how much time does #NEAR/n cost really depends on the inverted list itself. If the terms’ inverted list is very long and the locations are close, it would cost more time.

3.4 Fields

From my experience about fields, I would like to talk about both its advantage and disadvantages.

- **Strengths –** In the first place, it reduces time consumption because it significantly reduces the set of documents to be scanned. It raises up the precision when the term is really needed to be in that field. For example, ‘food’ is a good field for ‘fish’ when user wants to search for something like fried fish.

- **Weakness –** It is easy to feel that if we put a term in a wrong field (not related to information need), it will reduce both recall and precision. It separates aside the possible correct documents. For example, ‘food’ is not a good for ‘fish’ when user wants to search for something like fishing rod.

3.5 Success & failure

Whether a query really represents the information needs significantly determine whether such a search process is a success or a failure. I give two examples here indicating how a query could be a success and which kind of factors would lead it to fail.

- **A success –**149:#AND(uplift at yellowstone.keywords #NEAR/1(national park)).
  MAP = 0.18, P5 = 0.8, P10 = 0.6, P20 = 0.55, P30 = 0.53, Recall = 0.26.
  The reason why it approaches a very good precision and also not bad recall is because of the combinational use of fields, NEAR/n and AND. First of all, ‘national park’ would really be the information need and NEAR/1 constraint the two terms as one phrase. Secondly, considering ‘yellowstone’ as a keyword speeds up the processing and also maintain a good recall because documents related to Yellowstone would likely to put that term into keywords field. Finally, ‘uplift’ is not easy to determine its location, and we do not need documents which contains only ‘uplift’. Thus AND is the best choice for connecting those components.

- **A failure -** 190:#OR(#NEAR/1(brooks brothers) clearance.title).
  MAP = 0.02, P5 = 0, P10 = 0.1, P20 = 0.1, P30 = 0.07, Recall = 0.1
  The reason why it lowers down both precision and recall is because of the misuse of title and or. Firstly, the real relevant documents talking about ‘clearance’ might not appear on the title. The field filters out a lot of possible correct documents. For another, OR operator counts documents with only ‘brooks brothers’ and documents with only ‘clearance’. The user may only care about Brooks Brothers clearance but not other unless information.

3.6 Feelings & Summary

Although there are good algorithms, together with good models, the search engine is not able to successfully search out all the relevant documents. The gap between information need and query is not easy to get through. However, for a certain type of use, e.g. web search, we can develop good searching strategies to tell search engine to provide better results. We can also implement feedback system to let search engine learn what might be the best results.

In summary, precision and recall is a tradeoff. It is very hard to get middle around because information need is hard to guess. AND and NEAR operator provide better precision whereas OR operator performs better in recall. Ranked model is generally better. Strategies are really important to obtain the ‘best’ result in a certain scenario. However, ‘best’ is never easy to get.