Your Name: ADITYA GAUTAM

Your Andrew ID: agautam1@andrew.cmu.edu

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.
   Ans : 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)?
   Ans : 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.
   Ans : 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)?
   Ans : 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: Aditya Gautam

Your Andrew ID: agautam1@andrew.cmu.edu

Homework 1

1 Structured query set

1.1 Summary of query structuring strategies :

Ans : For creating queries, I tried to think it from the perspective of the user, who is curious about something and using few keywords to get that information. After looking at the query, first I tried to segment if it is based on some specific name/place or something in particular like some famous movie, some actress, a city or a sportsperson. In this case, the query need to be more or less intact and one or multiple words needs to be clubbed together. So, in these type of queries, I basically used a near operator with a distance of 1 or more, depending upon if there is a possibility of word in between like Sachin Tendulkar needs to be done with near/2 operator as middle name is mentioned in many webpage. With this, I also search for words in the title.url along with the body.

For other queries, I first figured out the alternative words or synonyms, which equally represent user intentions as that of original query and can be replaced with the word without changing the original meaning of the sentences/query. Like a query “sewing instructions” is as good as “sewing guidelines” or “sewing tutorial”, all implies the same i.e. the user is looking for the information about how to sew. In addition to this, I have think of the way in which the query can be formed and if the words needs to be present in the same sentence and paragraph. If rearrangement of words still gives us the same logical meaning i.e. the query, then I used OR operator on both or more NEAR operator query. Likewise, possible combination were retrieved.

Another important thing to note is the representation or words i.e. url, title, body, inlink etc. If someone is looking for something specific like a place or a hotel, then putting the words in URL and title would most likely fetch better results. Also, some abbreviation can be ambiguous, so I have queried for full form with NEAR operator with distance 1 as words needs to be next to each other. If a query is unique like OCD,I queried for obsessive compulsive disorder as some sites might not have mentioned OCD. So, to conclude, I tried to think myself as a user and made changes in the query and operator as per the need and structure.

1.2 Structured queries

Queries and the description:

69:#OR(#NEAR/3(sewing instructions)) #OR(sewing.title) #OR(sewing.url)
Here the user is trying to look for the instruction of sewing. So, both the words must be present in the query, hear the use of near operator. I have also added sewing in the title and the URL to fetch more relevant results. Since it is instruction for sewing, so it is very likely that words needs to be present next to each other with very less space i.e. few stop words etc. Otherwise, a paragraph with these two words used in different context can be fetched. So, this explains the use of synonyms, NEAR operator and title/url addition.

79:#OR(voyager.url voyager.title voyager.keywords voyager.body)
It is a one word query, which could be ambiguous. Looks like person is just looking for 'voyager', so any document which has this words, present either in the body or title or url should be counted into the result based on the frequency of this term. So, I used all possible aspect of this words in HTML/PDF or any other type of document.

84:**#OR(#NEAR/3(continental plates) #NEAR/4(plates continental)) #OR(continental.url) #AND(continental.url plates.url)**
With "Continental plates", the person is most likely looking information about tectonics plates, which is a widely used word. So, along with continental, it is important to add tectonics words also as an additional query. Most of the query must be next to each other like continental plates, should ideally be next to each other but taking into consideration some exception and reverse possibility, I kept the NEAR operator distance to 3 for queries as it is expected the words to be very near to each other else false query may come up.

89:#OR(#NEAR/1(Obsessive-compulsive disorder)) #OR(ocd.url ocd.title)
The query 'OCD' is just a one word abbreviated query, which might have a different full form. So, it is difficult to know the real intention of the user's. Since OCD stands for Obsessive-compulsive disorder, I queries for this rather than OCD, as many sites might be having full form. Since it is a full form,it has to be next to each other, thus the distance of NEAR is kept at 1,to avoid any false alarm. With this, I have also added the original 'OCD' query in URL, title or just the body.

108:#NEAR/1(ralph owen brewster) #AND(#OR(brewster.url) #OR(owen.url))
Here, the person is looking for the information about Ralph Owen Brewster. So, keeping the name intact (NEAR/1) with a possibility of name present/Surname present in the URL would most likely fetch good results. All the words has to be present else, it might give the result for person with same name or same surname. I tried removing the middle word and saw a dip in MAP, so looks like person is called by the full name and not just ralph Brewster or owen Brewster or some other combination.

141:#AND(va #NEAR/5(dmv registration)) #AND(va #NEAR/5(registration dmv)) #AND(virginia #NEAR/6(dmv registration)) #NEAR/6(virginia motor registration)

Person is looking for information about motor vehicle registration in VA i.e. Virginia. Taking any two words out of the original query will give ambiguous and unwanted results, so we would want to have all the words in any document. Some webpages might be using VA and some Virginia. So, result from both the queries are clubbed together with distance of 5/6, as anything more than this distance could give us ambiguous results. It experimented with different distance and came to this conclusion.

146:#OR(#NEAR/1(sherwood regional library)) #OR(#NEAR/2(regional library sherwood))

The query is about a particular library, so omission of any word would result in ambiguity and generalized results. So, it is mandatory that we need to include all the words for the retrieval. In most of the sentences/title, the words need to appear together, so distance of NEAR is kept at 1. However, there is also a possibility that sentence could be like this "reginal library of sherwood" or similar query using some stop words. So, I have included this case too.

153:#OR(pocono.url pocono.title pocono.body)

Single word query by user of a particular place or famous thing. So, any document with this keyword should be relevant. Likewise, if a webpage contains this word in the body or title or url, it should be fetched. Thus I have included all the relevant fields like URL,title and body.

171:#OR(#NEAR/1(ron howard) #NEAR/1(ron.title howard.title)) #OR(RonHoward.url #AND(Ron.url Howard.url))

Person is looking for information about some famous person/celebrity/sportsmen with given name and surname or some latest news about him. So, all documents which contains both the words together should be relevant thus NEAR/2 is used. With this, if the full name is present in the heading/title of any page, it should also be included as it would be relevant like heading of a latest news. Addition of url wouldn't make much of a difference as full name should be present in the body somewhere, but good to have it.

197:#OR(#NEAR/40(idaho state flower))

Idaho state flower query is corresponding to any alternative famous name. All the words are essentials and needs to be there in the web page else we might get generalized web pages.So, I have used the operator NEAR with the distance 40. Distance of 40 is used since the words like 'Idaho' and 'State flower' could be at a distance within a sentence or a paragraph and would be a revelant result. Experimented with various distance 40 and this seems optimized.

2 Experimental results

2.1.1 Unranked Boolean

	BOW #OR	BOW #AND	Structured
P@10	0.0000	0.1100	0.3000
P@20	0.0150	0.1350	0.3500
P@30	0.0200	0.1533	0.3067
MAP	0.0020	0.0665	0.1771
Running Time	00:07.2	00:01.4	00:02.2

2.1 Ranked Boolean

	BOW #OR	BOW #AND	Structured
P@10	0.1700	0.3700	0.5500
P@20	0.2800	0.4450	0.6500
P@30	0.3367	0.4633	0.6333
MAP	0.1071	0.1882	0.3032
Running Time	00:07.6	00:01.6	00:02.1

Please note that the time taking is more or less same in both ranked and unranked, as most of the time is taken in retrieving the document and sorting is being done in the unranked model based on the external docID.

3 Analysis of results: Queries and ranking algorithms

AND and OR are not so intelligent queries as they take union and intersection irrespective of the user's intention of filtering/retrieving useful information through query. **OR query** retrieve all the documents which has any of the word present, as the result we can expect lots of document, with most of them not so relevant as it may have only one relevant query word. Like "sherwood regional library" will give the result of all the doc with any word present, however we might need only the docs with all the words present in close vicinity. Likewise, MAP and P@n for OR is lowest among all.

AND query on the other hand would retrieve comparatively more relevant results as it requires a **document to have presence of all the words.** Like for above query, it will retrieve all docs with all three words present, which is comparatively better but not very accurate since we haven't considered the position, distance and the order of the word. **Due to this randomness and position dependency, AND will better results(performance) than OR, but worst than structured query (takes position into consideration).** Likewise, MAP and P@n for AND is much better than OR but not better than structured.

Structured query can be customized by analyzing the understanding of the user and best guess at what the user is looking for. Likewise, we can categorize query words into different buckets and apply OR and AND operator along with NEAR, which **takes into consideration the distance and order of the words, designed as per user intention and words format in pages**. Likewise, the query is more precise and the document retrieved would also be better since it specifically caters the closest user intention as presented by various operators applied with various distance. **So, MAP and P@n for structured query is highest i.e. 3032(Ranked),.**

Out of retrieved result, the scoring is done based on the number of terms present in the doc and the query operator algorithm. Likewise, **the likelihood of relevance of the page should be proportional to the number of occurrence of the terms, in most cases**. So, ranking model is likely to give much better results, as compared to unranked, which is why the MAP and P@ of ranked is higher than unranked model in all cases mentioned above. Ideally, the time for ranked should be higher but it is more or less same in this, as we are sorting in unranked model based on the external doc ID. Also, good proportion of the time is consumed in retrieving the relevant documents from the indexing and thus contribution of sorting relevant documents wouldn't play that important role in the timing.

4 Analysis of results: Query operators and fields

Every operator has its own strength and weakness and it depends on the query and user's intention. Like for a single word query you need to just look for the word. Thus, OR operator will do the task. However, when you know that the words are dependent and needs to be clubbed together, we need to use NEAR operator in combination with other similar or complementary query.

While signifying the importance of the distance and order, NEAR operator was successful to a very good extend like searching for OCD was quite easy. But in other cases, determining the best suitable distance was a bit tough and challenging. Also, the position i.e. body, url, title etc. played a significant role i.e. considering them the score (i.e. rank) of the page can go up. However, I think that different sections should be given different weightage like presence of a word in URL or title should be much higher than body, thereby improving the rank of the page.

I would say that using OR, AND and NEAR operator, I was succeeded in answering some queries fully and other partially. Like, a query OCD could mean multiple abbreviation and thus the we can't say for sure which would cater the user's information need. In this and some other cases, none of the operator mentioned was very helpful thus doesn't remove the ambiguity completely.

Even though, I am able to get pretty decent result (0.3032 MAP-highest as per leaderboard) for structured query but there are lot of ways in which further optimization can be done and more accurate results can be retrieved. Like, **I felt the use of window operator** in some queries (where order doesn't matter) and also an algorithm to retrieve similar words(synonym). With this, there has be more precise way to categorize ambiguous words like query 'Apple' could be fruit or Apple Inc. **So, I can say that these queries operator, fields and scoring methodology partially satisfied my expectation.** Intelligence can be induced using machine learning, recommendation, users past query etc.