Your Name: Raghav Sharma

Your Andrew ID: rvsharma

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.

Answer: 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.

Answer: 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.

Answer: 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.

Answer: Yes

Your Name: Raghav Sharma

Your Andrew ID: rvsharma

Homework 1

1 Structured query set

1.1 Summary of query structuring strategies

#NEAR #AND #OR operators

For creating the structured queries, we can use the combination of #NEAR/n, #OR, #AND operators. The strategy for developing structured queries requires a knowledge of precision and recall parameters to a query result. The parameter P, precision is given as what fraction of the returned results are relevant to the information need. While the recall R, for a query is defined as the fraction of the relevant document returned by the system.

The #AND operator increases the precision but has low recall, #OR operator increases recall but has low precision. The NEAR operator is an extension of the AND operator but with proximity considerations. Therefore, if there is a term that is required to be in the query then we can use the AND operator as a filtering criterion. On the other hand, if the we have options within the query arguments to be included then we can use OR to signify an option between those arguments.

Given a set of bag-of-words OR queries, we can use the above model to create a generic strategy for creating the structured queries. If the argument terms are important then we can use AND to include them all in the structured query and use #OR to indicate the alternate choices between them. Lastly the #NEAR operator is an extension of the #AND operator in the Boolean Retrieval models. It enforces the criterion that the argument terms must occur in the same document and within the proximity of each other. The n in the #NEAR/n indicates the allowance that the arguments are given for the proximity. The closer the proximity the better the precision, lower recall and farther the proximity the lower the precision and higher recall. Thus as there is a tradeoff between precision and recall not only between the #AND and #OR operator but also within the #NEAR/n operator, an estimate based on the best guess must be made while constructing the query.

1.2 Structured queries

Original Query 1 > 718: #OR (controlling acid rain)
Structured query 1> 718: #AND(controlling #OR(#NEAR/3(acid rain) #NEAR/3(rain acid)))
Answer 1: Here I used the #AND bag of words model to combine with the high recall of the #OR operator.
Answer 2: no important deviations from default strategies
Answer 3: The word controlling is inherently important to the query and is used as is as an argument to the #AND operator. The word acid rain should appear in proximity to each other and can occur in either format. That is, acid followed by rain or rain followed by acid. Thus, these two combinations must be given as an #OR argument which contains the #NEAR Inverted list operator as arguments.

Original query 2: 719: #OR(cruise ship damage sea life)

Structured Query 2: 719: #AND(#OR(cruise ship) #NEAR/5(damage sea life))

Answer 1: this query can be divided into two parts cruise ship words are synonyms and can harm the sea life. Both of these divisions are important for the query results and thus given, as an argument to the #AND operator.

Answer 2: yes, the word cruise and ship both are synonymous to each other and therefore included as arguments to the #OR operator.

Answer 3: The root operator #AND combines two necessary segments, the #OR operator combines the synonymous terms and we use the #NEAR operator for signifyingthe proximity that may exist between the terms 'damage sea life'

Original query: 724: #OR (iran contra)

Structured query: 724: #OR(#NEAR/3(iran contra) #NEAR/3(contra iran))

Answer 1: Here I used the #OR bag of words model to improve on the recall of the results.

Answer 2: no.

Answer 3: The word iran and contra can appear in any format and these combination are given as an argument to the #OR operator.

Original query : #OR(low white blood cell count)

Structured query: 725:#OR(#NEAR/3(low.body #NEAR/1(white.body blood.body cell.body) count.body) #NEAR/3(low.body count.body #NEAR/1(white.body blood.body cell.body)))

Answer 1: Here #OR bag of words model was used along with #NEAR operators

Answer 2: No

Answer 3: the phrase 'white blood cell' is important to the information need and is thus bunched with parameter 1 for the #NEAR operator. The words low and count can occur within distance of 3 to each other and the phrase 'white blood cells'. The operator OR is used to indicate that both the combination is possible.

Original query: #OR(airline overbooking)

Structured query: 733:#OR(#NEAR/3(airline overbooking) #NEAR/3(overbooking airlines))

Anwer 1: The #OR bag of words model is used along with proximity operators for the noun airlines and verb overbooking.

Answer 2: No

Answer 3: The phrase airline overbooking can occur within 3 terms of each other and their places can be exchanged. Thus these are given as an #OR argument for including in the results.

Original Query : #OR(recycling successes)

734:#AND(recycling.title #OR(#NEAR/4(recycling successes) #NEAR/4(successes recycling) successes))

Answer 1: This uses a mixture of #AND and #OR bags of words strategy to improve recall and precision.

Answer 2: Yes, the word recycling is assumed as an emphasis on the recall document. It is followed by the title field indicating that we nly need documents which have recycling as the title and might containg the successes.

Answer 3: the query is combined with #OR operator since the location of the word 'recycling' and 'successes' can often be interchanged.

Original Query: #OR(afghan women condition)

735:#OR(#NEAR/5(afghan women condition) #NEAR/5(condition afghan women) #NEAR/5(afghan condition women) #NEAR/5(condition women afghan) #NEAR/5(women condition afghan))

Answer 1: We use the #OR bag of words model along with proximity operators to improve the precision and lower the recall.

Answer 2: No

Answer 3: the phrase 'afghan women condition' can occur in any of the orders so we use the permutation to include in the query arguments and use an OR operator to specify that these can occur in any order.

Original query: #OR(artificial intelligence)

741: #OR(#NEAR/3(artificial.title intelligence.title) #NEAR/3(artificial.body intelligence.body) #NEAR/3(artificial.url intelligence.url))

Answer 1: here we use #OR bag of words model along with field to specify that these words can either occur in a url, title or body of a document.

Answer 2: Yes, we make specific use of the fields to specify the possible occurences of these query terms

Answer 3: the word 'artifical' and 'intelligence' can occur within 3 distance of each other and still be a relevant document. The phrase 'intelligence artificial' does not make any sense so, we can combine the probability of these arguments with #OR operator in each of the possible fields of title, body and URL.

Original Query: #OR(counterfeit ID punishments)

744:#OR(#NEAR/5(punishments.title counterfeit.title ID.title) #OR(#NEAR/5(counterfeit ID punishments) #NEAR/5(punishments counterfeit ID) #NEAR/5(ID counterfeit punishments)))

Answer 1: Here we use the #OR BOW strategy and also specify that the occurence of these words in the title is also relevant.

Answer 2: No

Answer 3: Here again we make permutation of the words ‘punishments’, ‘ID’ and ‘counterfeit’ and the satisfying parameter for proximity operator is kept at 5. These permutations are then combined with the #OR operator.

Original query: #OR(outsource job india)

746: #OR(#AND(india.title #NEAR/5(outsource.body job.body)) #NEAR/5(outsource job india) #NEAR/5(job india outsource) #NEAR/5(india job outsource) #NEAR/5(job outsource india) #NEAR/5(outsource india job) #NEAR/5(india outsource job))

Answer 1: The information need presented here can be thought of as ‘The jobs that are being outsourced to India’. We used #OR BOW strategy to develop the query.

Answer 2: No

Answer 3: since India can occur in a title and outsource and job in body we can make use of the field operators to specify their location in the document. The permutations of the word ‘outsource’, ‘job’ and ‘india’ are specified with the proximity operator and given as an argument for the #OR operator.

---

2 Experimental results

2.1 Unranked Boolean

	BOW #OR	**BOW #AND**	**Structured**
P@10	0.0000	0.2000	0.3900
P@20	0.0000	0.2250	0.3600
P@30	0.0033	0.2367	0.3100
MAP	0.0002	0.0489	0.0505
Running Time	00:08	00:02	00:02

2.2 Ranked Boolean

	BOW #OR	**BOW #AND**	**Structured**
P@10	0.0400	0.3800	0.4800
P@20	0.0800	0.3650	0.3800
P@30	0.0867	0.3300	0.3500
MAP	0.0079	0.0871	0.0547
Running Time	00:09	00:02	00:03

3 Analysis of results: Query operators and fields

#OR- the #OR operator combines the arguments and looks for them in the documents in the corpus and returns the maximum score of the document which satisfies the collection of the arguments in the document. Due to the unrestricted constraint of the operator, it has the property of giving low precision results. However, the strength of this operator lies in giving a high recall. It is specially, useful in cases where the user requires high recall of the documents but isn't too specific about the precision of the terms. Thus, as we can see from the tables above the #OR BOW model doesn't perform too well on the precision criteria P@n and MAP. It also takes a long time to go through the collection since it has to calculate the maximum of the term frequency occurring for a term in the document. Thus, the running time of the #OR is significantly more than the #AND or structured queries.

#AND – the #AND operator combines the arguments and looks for occurrence of them in the document that satisfies the occurrences of them in all the document. The #AND operator is more stringent in the constraints of the arguments and therefore has a high precision and low recall of the search results. The #AND operator is used in cases where the user cares more about the usefulness of the first few documents rather than the trailing ranks of the results. Therefore, it is more specifically can be used in web searches where user typically focuses on the first few model than the last ranks. Due to the high precision nature of the results of the operator they are more useful in the Ranked retrieval models than the unranked models. As we can infer from the tables the precision P@n and MAP for the #OR operator is high and has less runtime that it.

#NEAR/n – The proximity operator is an extension of the AND operator and uses the property of the postings list which record the positions to interpret the distance of each of its arguments. Therefore it has the same properties as the AND operator and offers more flexibility in terms of the possibility of the occurrences of its arguments.

Body, title, and URL fields – these fields are an important part of the query formation process. Using this we can specify in which part of a structured document we are looking for the arguments. However, they also increase the running time of the query since the specific fields have to be looked up when doing a search in the document.

All these operators and fields were an integral part in building the structured queries in the experiment. It was expected to increase the precision in the structured query approach and as we can infer from the results, there was a significant improvement in the precision from the #OR BOW and #AND BOW approach.

4 Analysis of results: Queries and ranking algorithms

Differences between the three different approaches (#OR BOW, #AND NOW, Structured query)

The characterstics of the #OR BOW approach is that it has high recall and low precision. As we can infer from the tables above, it is indicated by the low precision P@n for the #OR approach. Since each argument has to be matched to all the document the running time for this model is also higher(~5 times) than #AND and structured query.

While the #AND operator has a high precision and low recall which is evident in the MAP, and P@n parameters. They improved significantly (~10 times) over the baseline #OR model. The run time also has significantly improved over base #OR model. This can be inferred from the implementation of the AND operator in which the loop breaks as soon as one of the arguments is not present in the document.

The structured query approach is designed to utilize the #NEAR operator and the fields. With the proximity operator the precision can be improved over the #AND operator. This is due to the property that the NEAR/n operator allows us to specify the approximate location of each of these terms in the document. Since the #NEAR is an extension of the #AND operator we do not expect the run time to vary much from the #AND implementation. However, the run time does increase from the #AND operator due to the length of the query and the depth of the query tree that has to be evaluated.

Difference between Ranked Boolean and Unranked Boolean retrieval models:

In Unranked Boolean model each document which matches is given a score of one and is thus unordered representation of the result. However, due to the disorderly nature of the results the precision is bad and is even 0 for the #OR approach for P@10, and P@20. Although since we do not have to worry about ordering the document and calculating the scores for each individual document the run time is better than the Ranked model.

In Ranked Boolean model each document score is calculated depending on the matches it has for the arguments of the query. The precision increases for this model in both P@n and MAP parameters. But the calculation of the document score takes time and the run time is more than the unranked model, although not significantly more.