Your Name : Aditya Gautam

Your Andrew ID : agautam1@andrew.cmu.edu

Homework 4

Collaboration and Originality

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 : Aditya Gautam

Your Andrew ID : agautam1@andrew.cmu.edu

Homework 4

1 Experiment 1: Baselines

Provide information about the effectiveness of your system in five baseline configurations.

Document the parameter settings that were used to obtain these results.
Comment on the quality and character of the query expansion terms that were included, and the weights that were produced. Do they seem reasonable? Provide information about a few example queries to make your points, for example queries that had the most dramatic change in performance (good or bad) from query expansion, but do not provide information about every query individually. We are primarily interested in your observations about general trends, not quirky queries.
Comment on the effects of query expansion on your system and on the reference system. Are the two systems affected equally by query expansion, or are there important differences?

Ans 1)

	Ranked Boolean AND	Indri			
		BOW		Query Expansion	
		Your System	Reference System	Your System	Reference System
P@10	.1300	.3050	.3200	.3200	.3100
P@20	.1825	.3575	.3675	.3650	.3525
P@30	.2183	.3400	.3417	.3300	.3417
MAP	.0750	.1539	.1591	.1546	.1549
win/loss	N/A	17/3	17/3	18/2	19/1

Parameters used :

- retrievalAlgorithm=Indri
- Indri:mu=1000
- Indri:lambda=0.7
- fbDocs=10
- fbTerms=10
- fbMu=0
- fbOrigWeight=0.5

These parameters are chosen as this is what we are expected to use for experiment 1. Total number of terms and total number of words used in expansion are both set to 10. So, we don't see a very huge difference in the quality of the results. Moreover, the weight set for original and expanded query are both equal i.e. 0.5 which are not optimized parameters, as it gives equal importance to both the original and new terms extracted from top docs. Taking these factor into consideration, we see that the contribution of the expanded query is not significant thus, we are seeing that the MAP value has increase very slightly.

The character that were extracted through query expansion were very significant to the original query. Some of them were synonyms and some where words related to the original intention. To some extend these additional terms is like connecting some missing dots and suggest more appropriate words which can make the original query meaningful. In addition to this, the diversity of result increased a lot due to query expansion and we see that the risk factor of the ambiguous query has reduced a lot. Since the engine doesn't know the exact intention of the ambiguous query like one word. For example, avp query can be related to volleyball association or alien vs predator or antivirus. So, query expansion includes all the diverse term and help in extracting the top docs from all the different meanings. The expansion terms seem to be very reasonable and relevant with reference to original query.

Example query :
Original Query : avp
Expanded Query : 0.5216 avp 0.0501 newsletter 0.0449 rfc3588 0.0429 tour 0.0403 value 0.0402 antiviru 0.0372 beach 0.0369 diameter 0.035 croc 0.0342 alien

The expanded query words signify all the three diverse meaning of the avp. The weights of the original query terms are highest as expected and words next in resemblance like tour,rfc etc. which occurs most frequently in top documents are.

Weights of the new words formed from the original query also seems to be reasonable like the words which is closet to the original words are having the maximum weights after the original query words. So, in a way the weights of the words can be correlated to the similarity or the significance of the original query words/intention.

The most significant change is in the query "vldl levels" which has the original MAP value of 0.0009 in the AND baseline. With Indri model query expansion, the value has changed to .2007, which is one of the drastic increase in the MAP value. Other then this, there are about 10-11 queries for which the MAP value has increased very well. For others, the MAP increase is not that significant and only for one query, we see a decrease in the MAP value.

Two system i.e. my and the reference has more or less same effect on the query expansion. There is not a huge difference in the MAP values since the total words and terms used were only 10, and the weight is 0.5, thus limiting that impact of the query expansion.

2 Experiment 2: The number of feedback documents

Provide information about the effect of the number of feedback documents on query expansion.

	Ranked Boolean AND	Indri BOW, Reference System	Query Expansion, Reference System Initial Results					
			Feedback Documents					
			10	20	30	40	50	100
P@10	.1300	.3200	.3100	.3100	.2950	.2900	.3353	.3050
P@20	.1825	.3675	.3525	.3675	.3575	.3625	.4088	.3750
P@30	.2183	.3417	.3417	.3533	.3500	.3450	.3765	.3500
MAP	.0750	.1591	.1549	.1585	.1582	.1596	.1764	.1678
win/loss	N/A	17/3	19/1	19/1	18/2	18/2	18/2	18/2

Document the values of any parameters that were held constant during this experiment. Comment on the effect of varying the number of feedback documents on the quality and character of the query expansion terms that were included, and the weights that were produced. Were any values consistently better than other values? Does using more documents tend to help the results, or hurt the results? Why? Provide information about a few example queries to make your points, for example queries that had the most dramatic change in performance as the number of documents varied, but do not provide information about every query individually. We are primarily interested in your observations about general trends, not quirky queries. If using more documents improves expansion quality, is the improvement worth the added computational costs?

Ans 2)
Parameters which were held constant during the experiment.

- Indri:mu=1000
- Indri:lambda=0.7
- fbTerms=10
- fbMu=0
- fbOrigWeight=0.5

The only variable parameter was number of docs. With an increase in the number of feedback documents, the quality of retrieval improves i.e. MAP increases. However, there is not much impact on P@10 values, so looks like the top results were more or less same. There is a significant increase in the number of queries which are benefitted as compared to reference Indri System. However, on increasing the number of documents, there is no impact on the number of queries benefits. Also, if we increase the number of documents more than 50, we see that there is a decrease in the MAP value, and thus the quality of result. This is expected as the documents which are lower in rank are not of good quality and can have spam association. The result for 100 terms for query “avp” shows that there are many unwanted terms in the query like spam

medical products. So, this clearly shows that the quality of results increases up to certain extend with the increase in number of document and after that it reduces. This understanding is consistent with the results I got. **The best number of documents seems to be 50.**
So, I can say that increase the number of document towards certain threshold improves the quality of result, but increasing it beyond that decrease the MAP value/quality of retrieval.
With the document number set to 10 and 20, the number of queries which performed better than baseline are 19 out of 20, however it remains constant to 18 on increasing number of documents after that.

Other thing to note is that the weights of important term increase a little bit with the increase in the number of document, which also seems to be consistent of the understanding that the frequency/common words associated with the original query are more likely to appear in the top documents. Like for the example query "avp", the terms weightage increase in this fashion: .19 -> .30 -> .44 -> .48 -> .52, with the document number equals to 10,20,30,40,50 and 100 respectively. This implies that term which are most important will be given higher weightage. However, moving from 50 to 100 doesn't lead to a huge increase in the weight as the document score was less, which is a multiplying factor in the
Example query : avp

Document size = 20
#Wand(0.308 avp 0.0493 newsletter 0.0449 rfc3588 0.0402 antiviru 0.0349 value 0.0266 rfc4006 0.025 rfc4740 0.0245 code 0.0207 rfc4004 0.0199 beach)

Document size = 40
#Wand (0.4809 avp 0.0498 newsletter 0.0449 rfc3588 0.0403 value 0.0402 antiviru 0.0382 tour 0.0369 diameter 0.0297 beach 0.0289 code 0.0266 rfc4006) -> No Alien word

Document size = 50
#Wand (0.5216 avp 0.0501 newsletter 0.0449 rfc3588 0.0429 tour 0.0403 value 0.0402 antiviru 0.0372 beach 0.0369 diameter 0.035 croc 0.0342 alien) -> Alien word appear

With the increase in the number of documents, the execution time also increases as expected. Addition of more document surely increase the quality but the execution time also increase proportionally and it make sense to find the optimal value (50 in this case). In real life, I would assume that it would run in distributed cluster in parallel, so increasing the MAP value with a small increase in computational cost is worth it. There needs to be a balance between the quality and the computation, if both are important resources.

3 Experiment 3: The number of feedback terms

Provide information about the effect of the number of feedback terms on query expansion.

	Ranked Boolean AND	Indri BOW, Reference System	Query Expansion, Reference System Initial Results					
			Feedback Terms					
			5	10	20	30	40	50
P@10	.1300	.3200	.3000	.3100	.3150	.3050	.3100	.3050
P@20	.1825	.3675	.3475	.3525	.3625	.3650	.3625	.3625
P@30	.2183	.3417	.3370	.3417	.3483	.3500	.3517	.3517
MAP	.0750	.1591	.1520	.1549	.1579	.1611	.1612	.1617
Win/loss	N/A	17/3	19/1	19/1	18/2	18/2	18/2	18/2

Document the values of any parameters that were held constant during this experiment. Comment on the effect of varying the number of feedback terms on the quality and character of the query expansion terms that were included, and the weights that were produced. Were any values consistently better than other values? Does using more terms tend to help the results, or hurt the results? Why? Provide information about a few example queries to make your points, for example queries that had the most dramatic change in performance as the number of documents varied. If using more terms improves expansion quality, is the improvement worth the added computational costs?

Ans 3) Parameter that were kept constants during this experiments are:

- Indri:mu=1000
- Indri:lambda=0.7
- fbDocs=50
- fbMu=0
- fbOrigWeight=0.5

I only changed the values of number of feedback terms i.e. from 5 to 50. On increasing the number of term, it has been observed that the quality increases i.e. MAP value increases. With this, we can also see that P@30 increases but P@10 remains same.

Example query : avp
5 words : 0.5216 avp 0.0501 newsletter 0.0449 rfc3588 0.0429 tour 0.0403 value
10 words : 0.308 avp 0.0493 newsletter 0.0449 rfc3588 0.0402 antiviru 0.0349 value 0.0266 rfc4006 0.025 rfc4740 0.0245 code 0.0207 rfc4004 0.0199 beach
20 words : 0.1884 avp 0.0493 newsletter 0.0449 rfc3588 0.0402 antiviru 0.0266 rfc4006 0.0259 value 0.025 rfc4740 0.0209 code 0.0207 rfc4004 0.0165 unassigned 0.0145 vs 0.0127 alien 0.0121 id 0.012 mip 0.0118 type 0.0118 beach 0.0115 session 0.0114 auth 0.0113 part 0.0111 attribute
30 words : (0.1884 avp 0.0493 newsletter 0.04……

With respect to the new words added, we can see these words are not as important as the one which we got from the previous results (with less number of terms). This is expected also, since the words are ranked based on their score which is a function of term frequency, document score and the idf factor. New words coming up with the additional number of terms are correlated to the original query and would further help in retrieving the better results/documents.

Likewise, we can see an increase in the MAP value but the document at the top are the one which are very highly impacted/scored by the most frequency term i.e. top 5-10 words, there is not much impact in the P@10, however lower ranking document will be more relevant as there will now be more words which helps in correlating the document and getting the better docs.

So, we can conclude that adding more term will help in increasing the MAP value and Precision as explained above. Most drastic improvement has been seen in the query "avlv level" with an increase in MAP value from 0.0009 as baseline to 0.2007 with the number of terms set to 20.

With the increase in the number of term, the number of inverted list to be compared and measured increased thus we would need larger RAM, good caching and the fast I/O operations to perform this operation faster. Likewise, we can say the computational cost will increase heavily with increase in number of terms. The time taken on my system increase from 18 sec for 5 term query to up to 180 sec for 50 term query. In small system with only serial execution, I don't think the computation cost can be justified with the increase in MAP value, which is not very significant. **However, on a distributed cluster, it makes sense to use 50 term for query expansion and then retrieve the results.**

4 Experiment 4: Original query vs. expanded query

Provide information about the effect of varying the weight between the original query and the new expansion query.

	Ranked Boolean AND	Indri BOW, Reference System	Query Expansion, Reference System Initial Results					
			fbOrigWeight					
			0.0	0.2	0.4	0.6	0.8	1.0
P@10	.1300	.3200	.3875	.3500	.3500	.3200	.2600	.2900
P@20	.1825	.3675	.4250	.3875	.4050	.3775	.3550	.3475
P@30	.2183	.3417	.4292	.3817	.3733	.3617	.3450	.3333
MAP	.0750	.1591	.1836	.1868	.1806	.1689	.1586	.1514
Win/loss	N/A	17/3	19/1	19/1	18/2	18/2	17/3	17/3

Document the values of any parameters that were held constant during this experiment. Comment also on the balance between the original query and the expansion query. Is a combination of the two queries worthwhile? Why or why not? How does the stability (win/loss) behavior compare to just using the expanded query alone?

Ans 4) Parameter that were kept constants during this experiments are:

- Indri:mu=1000
- Indri:lambda=0.7
- fbDocs=50
- fbTerms = 50
- fbMu=0

The parameters i.e. fbDocs and fbTerms are the best parameters (both 50) observed from the previous experiments. The parameter played around here is fbOrigWeight which decide what is the weightage that needs to there between the original query and the expanded query.

As per the results, given high weightage to the expanded query results in better results and higher MAP value. The best value noticed is 0.2 i.e. the weightage of original query is 0.2 and the expanded is 0.8 in the WAND operator. This is as per expectation as expansion term already contains the original term with highest weightage (in almost all). Increasing the weightage of original query further results in decreasing the quality of result since the contribution of new terms will go low and the result will be more inclined toward the original query. Expanded query already include the original term with highest weights and increasing this parameter would reduce the significance of other related/added terms figured out in the query.

Another interesting thing to note here is that P@10 of the expanded query is best when zero weight is given to the original query and 1 is given to expanded query. This means that even though MAP might be highest at 0.8, but to get best results in top documents, only expanded query would be the best thing as it has all other relevant results which would diversify the query and get the documents with maximum relevant words.

So, I can say that having this parameter would help in maintaining a balance between the original and expanded query, and thus for getting best top results and overall MAP, playing with this parameter is very much needed as it signifies the weightage of added terms to be added into the final query, which is used for retrieving the results.

Coming to the win/loss ratio, we see that increasing this parameter consistently reduces the win/loss ratio. With fbOrigWeight equals to 0, only one query got results worse then the AND baseline, and other got better results thus win/loss was 19/1 = 19. With 0.2 and 0.4 as the weights on original query, the ratio reduced to 18/2=9. Increasing the weight further reduces the ratio further. So, we can say that giving more weightage to expanded query make more sense to fetch better results and quality.

5 Experiment 5: Effect of the original query quality

Provide information about how the quality of the original query affects query expansion effectiveness.

	Ranked Boolean AND	**Query Expansion, Reference System Initial Results**			
		BOW Original Query		**SDM Original Query**	
		Original	**Expanded**	**Original**	**Expanded**
P@10	.1300	.3200	.3883	.3785	.4678
P@20	.1825	.3675	.3650	.3883	.4500
P@30	.2183	.3417	.3883	.3900	.4267
MAP	.0750	.1591	.1905	.1960	.2241
Win/loss	N/A	17/3	19/1	18/2	19/1

Document the values of the parameters used for this experiment.
Does a difference in the quality of the initial retrieval make any difference in query expansion effectiveness or stability?

Ans 5) Below parameter were chosen to do this experiment:

- Indri:mu=2500
- Indri:lambda=0.25
- fbDocs=50
- fbTerms = 50
- fbMu=0

- fbOrigWeight=0.5

The value chosen for this experiment are the best values derived from the previous experiments and the previous homework.

As compared to original query, query expansion has much better results both in terms of MAP and P@n values. The results were better both for original BOW query as well as SDM query. Best results were for SDM query expanded query followed by BOW original query. This is expected as SDM cover various aspect of a particular query with the NEAR and WINDOW operator apart from the original query. Over to this, new similar words were added through the query expansion. So, overall high MAP value was expected as many possible interpretation of the original query with different weights were covered in the SDM Expanded query.

The result for expansion with the best optimal parameters values are always better than the original query for BOW and SDM model. MAP and P@N value is better for SDM model as compared for BOW model. So, we can say that, to get the best results, SDM model needed to be integrated with the query expansion with the use of optimal parameters.

Results were very consistent i.e. with the change in query from original to expansion, the MAP and P@N increased along with the computation time.

6 Analysis of results

You ran a lot of experiments, and have a lot of experimental results. The sections above discuss each experiment individually. In this section, we want you to think about general trends that you observed across the 5 experiments that have not been discussed in earlier sections.
How did query expansion affect the "high Precision" portion of a document ranking (the top-ranked documents) and the "high Recall" portion of the document ranking (farther down the ranking)? Where does query expansion have the greatest impact?
Was query expansion stable in your experiments (as indicated by the win/loss ratio)? Were any experimental conditions more or less stable? Was there a correlation between accuracy metrics and stability?
Is the increased computational complexity worth the increased accuracy (if any)? Keep in mind that a "production" implementation of pseudo relevance feedback would be much more optimized and faster than your implementation.
Feel free to include other comments about what you observed. You did a lot of experiments. This is your opportunity to let us know what you learned in this assignment.

Ans 6)

From the above experiments, I can say that query expansion is an important part and if optimized with current parameter, can significantly improve the overall search performance in terms of MAP and P@.

It has been observed the “high precision” doesn’t have a high impact with the change in number of words or number of documents. P@10 value remains more or less same with the change in these two variable. However, change the weightage given to the original query and the expanded query has a big impact on the P@10 value, it increases drastically. So, we can modify high precision by controlling these parameters and the query expansion. Recall has also been improved drastically for most of the queries with query expansion. Increasing the number of terms and number of document both resulted in the increase in the Recall. However, increasing the number of document before certain value reduces MAP as low quality documents showed up. Query expansion has the greatest impact in the MAP value as increased by around 300% with number of term and docs as 50, and fbOrigWeight (original weight) as 0. This implies that query expansion is very much needed if one wants to significantly improves the top results, so this must be a very relevant technique for search engines giants.

It has been observed from the experiment that most of the query i.e. around 90-95% queries resulted in better MAP values as compared to baseline. So, we can say that query expansion has resulted in better performance in improving the overall performance. Improvement in the queries overpower the bad performance in remaining 5-10% of queries, so expansion benefit overall performance. With the increase in number of terms, documents and/or expanded query, win/loss ratio decreases.

Queries were stable in the experiment; we see consistent decrease in win/loss ratio. Experimental conditions were stable in the experiment; we see consistent increase in the computation time and the memory usage with the increase in number of terms and docs. Accuracy metric and stability are inversely correlated. Increase in accuracy (MAP value) with increase in number of term/docs results in increase in MAP value, however win/loss ratio reduces. Thus, we can say that there are some query which suffers with the increase term/doc/weightage of expansion of query. So, we might have to compromise with one or the other parameters like MAP vs win/loss ratio. But since most of the queries are getting enhanced results, so focusing on increasing MAP value would be more valuable even with decrease in quality in one or two more query.

Computation vs improvement in the result depends on the computation cost, the resources available and the aim of the search engines. Increase in the results accuracy (high MAP/P@n), usually cost higher computation but it is worth it and with new distributed system/clusters, in which the computation can be done in parallel. Applying these technique is totally worth it for better user experience.