Your Name: Leonardo Neves

Your Andrew ID: lneves

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: Leonardo Neves

Your Andrew ID: lneves

Homework 4

1 Experiment 1: Baselines

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

	Ranked Boolean AND	Indri			
		BOW		Query Expansion	
		Your System	Reference System	Your System	Reference System
P@10	0.1300	0.3100	0.3200	0.3400	0.3100
P@20	0.1825	0.3625	0.3675	0.3625	0.3300
P@30	0.2183	0.3417	0.3417	0.3417	0.3517
MAP	0.0750	0.1588	0.1591	0.1644	0.1597
win/loss	N/A	5.67	5.67	2.8	0.81

Document the parameter settings that were used to obtain these results.

fb=true, fbDocs=10 ,fbTerms=10, fbMu=0 , fbOrigWeight=0.8 , Indri:mu=1000, Indri:lambda=0.4

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.

When using the reference system, since we are using the same top documents for each query, we get the same set of weights and top terms for each query, what makes the results worse in general, as expected. We compute the win/loss ratio of the indri bow compared to the ranked Boolean and the query expansion compared to the indri bow.

In general, the query terms for the queries do sound reasonable, like brooks brothers clearance that got terms like coupon and off. Despite of that, we did not One of the queries with most improvements was query #89,ocd. Adding terms like disorder to the query, like I did on experiments for previous homeworks, showed significant improvements in performance. On the other hand, some queries were hurt like query #79: voyager. Terms like Chrysler, star and trek are to general and ended up getting more non-relevant documents than the baseline. Query expansion, for this experiment, was not as good as expected since most terms, although related to the topic, were too generic and would hurt more than help on finding relevant documents.

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?

Since my system uses different documents per query, it is expected and true that my results would show better improvements than the reference system. By first ranking documents based on the original query, I increase the probability of having relevant terms for that particular query on the top documents.

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	0.1300	0.3200	0.3100	0.3150	0.2800	0.2550	0.2350	0.0800
P@20	0.1825	0.3675	0.3300	0.3250	0.3075	0.2925	0.2575	0.1225
P@30	0.2183	0.3417	0.3517	0.3450	0.3067	0.2883	0.2617	0.1283
MAP	0.0750	0.1591	0.1597	0.1565	0.1379	0.1257	0.1108	0.0612
win/loss	N/A	5.67	0.81	0.72	0.35	0.33	0.33	0.11

Document the values of any parameters that were held constant during this experiment.

fb=true ,fbTerms=10, fbMu=0 , fbOrigWeight=0.5 , Indri:mu=1000, Indri:lambda=0.4

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?

Including more documents was only responsible for hurting the queries. I imagine, since we are using the same set of top documents for expanding the queries, that the top documents will not be a good match for most queries and, if we expand the number of top documents, less relevant terms will be selected for the query expansion. This probably wouldn't be a problem if, instead of using the given initial results, we had used my system to get the initial results per query. Increasing the number of documents made more queries worse, like queries #102 and #105 that got better than the baseline when we had only 10 documents but worse with 40. An interesting fact is that, for the queries that got better than the baseline with 40 documents, all map values were better than the same map values for when we had 10 documents, showing that, for those specific queries, the generic initialization produced relevant expansion terms. Despite of that, since most queries are hurt and the overall results are much worse than the baseline, the added computational costs are not worth it.

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	0.1300	0.3200	0.3100	0.3100	0.3000	0.3100	0.3100	0.3050
P@20	0.1825	0.3675	0.3450	0.3300	0.3225	0.3200	0.3250	0.3175
P@30	0.2183	0.3417	0.3567	0.3517	0.3433	0.3517	0.3417	0.3417
MAP	0.0750	0.1591	0.1591	0.1597	0.1583	0.1577	0.1575	0.1578
Win/loss	N/A	5.67	0.67	0.81	0.81	0.67	0.81	1.0

Document the values of any parameters that were held constant during this experiment.

fb=true, fbDocs=10, fbMu=0 , fbOrigWeight=0.5 , Indri:mu=1000, Indri:lambda=0.4

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?

Increasing the number of terms decreased the MAP value, but the win/loss rate and even precision at higher ranks was not stable. If we consider only MAP values, increasing terms should only hurt the results. This happen because, like experiment 2, we are using general documents as the initial result. Getting more terms from the top documents will, for most queries, find more non-relevant and misleading terms than using a smaller number of terms. In fact, the result is so unstable because of the generic initial results, that using a small number of terms hurt more queries than using the value of 10 that was being used and, although getting lower MAP and P@n values, getting 50 terms, the maximum for our experiment, is the one to improve the largest number of queries, meaning that, for a system that prioritizes win/loss ratio, this would be the better. Probably, the set of 50 had a better distribution of relevant terms per query, hurting less queries than the previous experiments. Anyway, the difference between the results and the computational cost and the associated time required for it might not be worth for most scenarios. One example of a query that got worse from increasing the number of terms was the query #141 about the dmv va registration. When increasing the number of terms from 10 to 30, the map value for this query got much worse, probably because most of the 20 new terms were not related to the query. This was the only query that had significant changes until 50 terms.

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	0.1300	0.3200	0.0000	0.0250	0.2400	0.3150	0.2950	0.3100
P@20	0.1825	0.3675	0.0000	0.0550	0.2850	0.3375	0.3525	0.3625
P@30	0.2183	0.3417	0.0050	0.0733	0.3117	0.3483	0.3467	0.3417
MAP	0.0750	0.1591	0.0038	0.0310	0.1392	0.1598	0.1579	0.1588
Win/loss	N/A	5.67	0.00	0.00	0.35	0.81	0.67	0.53

Document the values of any parameters that were held constant during this experiment.

fb=true , fbDocs =10,fbTerms=50, fbMu=0 , Indri:mu=1000, Indri:lambda=0.4

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?

A combination of the query and expanded terms shows improvement on the overall MAP values if given slightly more weight to the initial query, although it hurts more queries than improves, as shown by the below 1.0 win/loss ratio. I believe that this result is misleading because of the initial results we are using. Probably, having a different initialization per query would get better overall results. Despite of that, giving a higher weight to the initial query seems more reasonable and presented better results, but having the expanded terms should be important. I have tested both the values of 50 terms and 10 terms, as the first had shown better win/loss ratio and the latter better map values for the previous experiment. The 50 terms experiment got me the best result, but the extra time necessary might not justify this improvement. Also, because of the generic initial values, the results for using the expanded terms alone is terrible, which should not be true for a specific initialization. From this, we could take that having more weight to the original terms seems important because those were selected by the user that knows the information need, but giving weight to the expanded terms is also important and this parameter should be tuned. In addition, additional terms show improvements but the additional computation cost might not make it worth using.

5 Experiment 5: Effect of the original query quality

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

	Ranked	Query Expansion, Reference System Initial Results

	Boolean AND	BOW Original Query		SDM Original Query	
		Original	Expanded	Original	Expanded
P@10	0.1300	0.3200	0.3050	0.3850	0.4050
P@20	0.1825	0.3675	0.3500	0.4375	0.4425
P@30	0.2183	0.3417	0.3550	0.4300	0.4300
MAP	0.0750	0.1591	0.1589	0.1999	0.2010
Win/loss	N/A	5.67	0.81	5.67	0.81

Document the values of the parameters used for this experiment.

fb=true , fbDocs =10,fbTerms=10, fbMu=0 , Indri:mu=1000, Indri:lambda=0.4, fbOrigWeight=0.6

I have used 10 terms instead of 50 as in the last experiment because of the time took by the 50 terms experiment.

Does a difference in the quality of the initial retrieval make any difference in query expansion effectiveness or stability?

Yes, having a better initial result clearly improves the result, although the win/loss ratio is the same. This happens because, for the improved queries, the improvement is larger and improves all the metrics we are analyzing.

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?

From the experiments, we could see that query expansion is capable of changing high precision and high recall, but it seems that high precision ones are the most affected, having a largest difference to the baseline on the overall values than the high recall. From the last experiment we can see that this would also hold for a better initialization, where we got a 2-point improvement on p@10 and p@30 had no difference. It reasonable to assume this would hold for other initialization set ups.

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?

No, it was not stable. The most stable condition was when expanding the number of feedback terms, experiment #3, where only two queries would go from getting worse to getting better or the other way around. I could not observe an explicit correlation between accuracy and stability, since sometimes we would have an experiment getting a lower win/loss ratio, but improving so much on the improved queries

that the overall precision and MAP would be better, even if the individual maps would not have all improved.

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.

I believe that, in a production environment, it would be worth it, but not on our scenario. On a production environment we would have specific query initialization and not a generic one like we did for the experiments and, by running a couple examples where I used my system as the initial result, this would already improve performance and justify using more terms and documents. Also, because of the scale, we could have distributed computation and cache that stored the expansion terms for popular queries, making it more useful and faster.

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.

From this assignment I have learned that query expansion is a good way of improving our results by re-ranking documents. Using a good system and producing a reasonable first rank of the documents improves our chances of selecting more meaningful and relevant expansion terms, which could improve significantly our results on top ranks and even, on a smaller scale, on lower ranks. In addition, since the user is the only one, if any, that actually knows the information need, we should give his/her terms a higher weight, although not too high. This weight is a parameter to be tuned by system, since some systems might have trained users that would produce better queries.