Your Name Mengyu Yang

Your Andrew ID mengyuy

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.

2. Did you give help of any kind to anyone in developing their software for this assignment (Yes or No)?

   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.

   Yes.

4. Are you the author of every word of your report (Yes or No)?

   Yes.

Your Name Mengyu Yang

Your Andrew ID mengyuy

Homework 4

1 Experiment 1: Baselines

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

		Indri			
	Ranked Boolean AND	**BOW**		**Query Expansion**	
		Your System	**Reference System**	**Your System**	**Reference System**
P@10	0.1300	0.2500	0.3100	0.2250	0.3100
P@20	0.1800	0.3125	0.3525	0.2875	0.4200
P@30	0.1683	0.3300	0.3467	0.3033	0.4400
MAP	0.0502	0.1754	0.1883	0.1664	0.2231
win/loss	N/A	19/1	19/1	18/2	19/1

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

Indri:mu = 1000

Indri:lambda = 0.7

fbDocs = 10

fbTerms = 10

fbMu = 0

fbOrigWeight = 0.5

1. Comment on the quality and character of the query expansion terms that were included, and the weights that were produced. Do they seem reasonable?

The quality of the query expansion terms varies from query to query and even varies from term to term in the same query. Some terms are pretty reasonable and helpful to boost the performance, while others are meaningless. Most expanded terms are (1) synonyms, (2) various morphological forms of original query terms, (3) important aspect of a concept in original query, (4) concrete example of a concept in original query, etc. Usually the expanded query contains the original terms and the original terms have the largest weight, but there are also cases in which the original terms are dropped. I will explain my observation by two examples.

2. 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.

Query 11: gmat prep classes

my system generated the expansion query: 11: #WAND (0.0036 i 0.0037 test 0.0040 score 0.0044 you 0.0048 training 0.0049 class 0.0056 mlic 0.0096 course 0.0106 prep 0.0303 gmat).

This is a good example of query expansion. MAP was improved from 0.0585 to 0.4273.

The quality of these expansion terms is quite good. Most expanded terms are synonyms or morphological forms of the original terms or reflect important concept in original query. E.g. “class” is the single form of “classes”, “course” is a synonym of “classes”, “training” has the similar meaning with “prep”, “test” is a general word for “gmat”, “score” is the most important aspect about “gmat”, “mlic” is the most famous courses for gmat prep, etc. These expanded terms help improve the search performance. By contrast, expanded terms like “I” and “you” provide little meaning, they got high scores just because they are words that commonly used in articles. The original words “gmat” and “prep” got the largest weight, following by “course” (synonym of classes). By contrast, useless words such as “you” and “I” got relatively low score, the reason behind which might be the idf effect implemented in query expansion algorithm.

There is also bad example such as query 1:obama family tree.

My system generated the expanded query:

1: #WAND (0.0032 4 0.0032 3 0.0033 5 0.0037 6 0.0365 history 0.0432 tree 0.0490 genealogy 0.0501 surname 0.0501 crest 0.1472 family)

This expanded query decrease MAP from 0.0180 to 0.0002.

Most expanded terms in this query are meaningless such as “4”, “3”, “5”, “6”. Documents contains those numbers are very likely to be non-relevant. The expanded query dropped the word “obama”, which lead to the loss of an important concept – users who search this query are concerns about Obama’s family tree instead of a random person’s family tree, so dropping “Obama” is detrimental to the search result. The original word “family” has the largest weight, followed by some reasonable terms such as “surname”, “genealogy” (synonym of family tree), “tree” (original term), “history” (similar concept with family tree). We can see here, even in the same query, the quality of the expanded terms vary.

(3) 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?

The two systems are NOT affected equally by query expansion. If we compare MAP between Ranked Boolean and Indri Query Expansion, both my system (from 0.0502 to 0.1666) and the reference system (from 0.0502 to 0.2231) improved a lot by query expansion. The improvement is larger for reference system than my system. However, if we compare MAP between Indri BOW and Indri Query Expansion, my system (from 0.1754 to 0.1664) actually loses some performance by doing query expansion while the reference system (from 0.1883 to 0.2231) benefit from expansion. The differences in performance might be due to the different parameter setting in query expansion, such as different original query weight and different smoothing fbMu.

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, Your System**	**Query Expansion, Your Initial Results**					
			Feedback Documents					
			10	**20**	**30**	**40**	**50**	**100**
P@10	0.1300	0.2500	0.2300	0.2200	0.2200	0.2450	0.2350	0.2400
P@20	0.1800	0.3125	0.2875	0.2950	0.2950	0.2900	0.2850	0.2825
P@30	0.1683	0.3300	0.3133	0.3033	0.3083	0.3050	0.3133	0.3000
MAP	0.0502	0.1754	0.1666	0.1618	0.1647	0.1652	0.1660	0.1610
win/loss	N/A	19/1	18/2	18/2	17/3	17/3	17/3	17/3

	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	0.3100	0.2500	0.2350	0.2200	0.2550	0.2550	0.2400
P@20	0.1800	0.3525	0.2975	0.3025	0.2975	0.3025	0.3050	0.2925
P@30	0.1683	0.3467	0.3167	0.3150	0.3183	0.3133	0.3167	0.3133
MAP	0.0502	0.1883	0.1737	0.1684	0.1680	0.1715	0.1700	0.1663
win/loss	N/A	19/1	18/2	18/2	17/3	17/3	17/3	17/3

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

Indri:mu = 1000

Indri:lambda = 0.7

fbTerms = 10

fbMu = 0

fbOrigWeight = 0.5

(1) 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?

As can be seen from the chart, fbDocs = 10 gives the best MAP and P@10/20/30. Increasing fbDocs to 20, and 30 leads to worse result, but continue increasing the fbDocs to 40 and 50 gives a good result similar to fbDocs. The worst performance is at fbDocs = 100. Correspondingly, the expanded terms at fbDocs = 10 /50 has the best quality and are most relevant. The expanded terms generated by fbDocs = 100 is least relevance and worst quality and the reason behind that might be the noise introduced by low-ranking documents. The expanded query in most cases contains the original terms and the original terms counts the largest weight, though there are certain cases that the some original terms are dropped and other expanded terms got more weight. For both my system and the reference system, set fbDocs = 10 gives better results than other values.

(2) Does using more documents tend to help the results, or hurt the results? Why?

It depends. Increasing document number from 20 to 40 helps the results but continue increasing to 100 hurts the result. Using more documents, the information we can use becomes larger, but at the same time, the noise given by low-ranking document is also increasing. There is a tradeoff between more information and noise. When the number of document is increased from 20 to 40, the benefit of getting more information is larger than the loss due to noise. After that, the effect of noise become larger than the effect of introducing more information.

(3) 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.

Take 5: 5:mitchell college as an example. The following are the results at fbDocs = 10, 50 and 100 respectively.

5: #WAND (0.0015 biology 0.0015 charles 0.0015 chri 0.0016 beauty 0.0017 community 0.0019 foul 0.0022 university 0.0025 school 0.0071 college 0.0374 mitchell)

5: #WAND (0.0028 science 0.0030 2 0.0039 state 0.0041 1 0.0057 school 0.0060 community 0.0068 0 0.0120 university 0.0385 college 0.0470 mitchell)

5: #WAND (0.0044 science 0.0045 from 0.0061 1 0.0074 0 0.0078 school 0.0087 state 0.0119 community 0.0257 university 0.0529 mitchell 0.0642 college)

From fbDocs =10 to fbDocs = 50, there are many useful words such as “school”, “university” added to the query. However, continue increasing fbDocs to 100 introduces some meaningless words such as “from”, which does not reflect users’ information needs.

(4) If using more documents improves expansion quality, is the improvement worth the added computational costs?

It depands. Using more documents will slow the speed of search engine but may improve the result to some extent. For scenario that result accuracy is most important is slow processing time is acceptable, the improvement worth the added computational costs. On the other hand, for scenario that very time-sensitive and little accuracy loss is acceptable, a relatively small fbDocs may be enough.

(5) 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?

The trend of the performance with different value setting is the same for my system and the reference system. Both the two systems give the best result at fbDocs = 10 and worst results at fbDocs = 100. Both the two system improved by using query expansion compared with Ranked Boolean. But compared with Indri BOW, both the two system perform worse after using query expansion.

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, Your System**	**Query Expansion, Your Initial Results**					
			Feedback Terms					
			5	**10**	**20**	**30**	**40**	**50**
P@10	0.1300	0.2500	0.2250	0.2300	0.2250	0.2300	0.2400	0.2350
P@20	0.1800	0.3125	0.2825	0.2875	0.2875	0.2825	0.3000	0.3000
P@30	0.1683	0.3300	0.3033	0.3133	0.3033	0.3050	0.3050	0.3033
MAP	0.0502	0.1754	0.1616	0.1666	0.1664	0.1647	0.1711	0.1707
Win/loss	N/A	19/1	18/2	18/2	18/2	18/2	18/2	18/2

	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	0.3100	0.2350	0.2400	0.2550	0.2500	0.2400	0.2350
P@20	0.1800	0.3525	0.2825	0.3000	0.2975	0.3000	0.3000	0.3025
P@30	0.1683	0.3467	0.3050	0.3233	0.3300	0.3317	0.3333	0.3283
MAP	0.0502	0.1883	0.1647	0.1722	0.1744	0.1746	0.1748	0.1747
Win/loss	N/A	19/1	18/2	18/2	18/2	18/2	18/2	18/2

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

Indri:mu = 1000

Indri:lambda = 0.7

fbDocs = 10 [Best parameter from Exp2]

fbMu = 0

fbOrigWeight = 0.5

(1) 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?

As can be seen from the chart above, the more the fbTerms, the better the MAP and P@10/20/30, which means that the quality of the query expansion terms is better for large fbTerms. For large fbTerms, more relevant terms are added into query. Since the number of terms become large, the weight of each term becomes smaller. When fbTerms = 40, both precision and MAP are better than other values.

(2) Does using more terms tend to help the results, or hurt the results? Why?

When the number of terms is very small, increasing fbTerms do help the results, but when the number of terms is already reasonable, adding more terms does not help the result. Too small set of query terms gives limited information and may generate ambiguity in information need. Thus adding some expanded terms usually help to improve the result. However, too many terms will introduce some meaningless terms which become noise to the expanded query. There is a tradeoff. Neither too small or too large fbTerms is good.

(3) 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.

Take #26: lower heart rate as an example. This query result improved quite a lot from fbTerms = 5 to fbTerms = 30 but does not change much from fbTerms = 30 to fbTerms = 50 . The following are the expanded queries when fbTerms = 5, 30 and 50.

26: #WAND (0.0060 your 0.0115 monitor 0.0193 lower 0.0308 heart 0.0422 rate)

26: #WAND (0.0015 i 0.0015 170 0.0015 190 0.0016 slow 0.0016 fitness 0.0016 120 0.0017 my 0.0017 watch 0.0017 130 0.0017 age 0.0018 low 0.0018 googlescout 0.0018 blood 0.0018 death 0.0019 150 0.0019 reebok 0.0021 cholesterol 0.0021 from 0.0021 mortgage 0.0022 exercise 0.0024 atenolol 0.0028 kit 0.0028 pulse 0.0031 you 0.0032 disease 0.0039 your 0.0053 monitor 0.0136 lower 0.0171 heart 0.0296 rate)

26: #WAND (0.0010 how 0.0010 use 0.0011 do 0.0011 price 0.0011 cache 0.0011 spring 0.0012 more 0.0012 may 0.0012 pressure 0.0012 tax 0.0013 rest 0.0013 exchange 0.0013 home 0.0013 have 0.0013 can 0.0014 beat 0.0014 about 0.0014 cardiosport 0.0014 interest 0.0014 140 0.0015 i 0.0015 170 0.0015 190 0.0016 slow 0.0016 fitness 0.0016 120 0.0017 my 0.0017 watch 0.0017 130 0.0017 age 0.0018 low 0.0018 googlescout 0.0018 blood 0.0018 death 0.0019 150 0.0019 reebok 0.0021 cholesterol 0.0021 from 0.0021 mortgage 0.0022 exercise 0.0024 atenolol 0.0028 kit 0.0028 pulse 0.0031 you 0.0032 disease 0.0039 your 0.0053 monitor 0.0136 lower 0.0171 heart 0.0296 rate)

When fbTerms is very small, the number of terms limit many useful terms. For example, when fbTerms = 30, there are many good expanded terms such as “blood”, “pulse”, “cholesterol” that are not included when fbTerms = 5. There terms help to express a specific information need. However, too many terms may also introduce some useless and non-relevant terms such as “do” “about” “have” which do not have much meaning.

(4) If using more terms improves expansion quality, is the improvement worth the added computational costs?

It depandes. From my observation, when the number of terms become larger, the time consumed is largely increased. However, from fbTerms = 10 to fbTerms = 50, the MAP increase is minor (0.1722 to 0.1747 for reference system). For scenario that accuracy result is most important and long-time process is acceptable, then the improvement worth the added computational costs. On the other hand, for scenario that time is more important and little accuracy loss is acceptable, then a small fbTerms may be preferred.

(5) 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?

The two systems affected equally by query expansion. The trend of the performance with different value setting is the same for my system and the reference system. Both the two systems give the best result at fbTerms = 40 and worst results at fbTerms = 5. Both the two system improved by using query expansion compared with Ranked Boolean. But compared with Indri BOW, both the two system perform worse after using query expansion.

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, Your System**	**Query Expansion, Your Initial Results**					
			fbOrigWeight					
			0.0	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**
P@10	0.1300	0.2500	0.2050	0.2100	0.2200	0.2250	0.2300	0.2500
P@20	0.1800	0.3125	0.2775	0.2750	0.2875	0.2850	0.2925	0.3125
P@30	0.1683	0.3300	0.2917	0.2883	0.3050	0.3133	0.3267	0.3300
MAP	0.0502	0.1754	0.1536	0.1630	0.1651	0.1673	0.1711	0.1754
Win/loss	N/A	19/1	17/3	17/3	18/2	18/2	19/1	19/1

	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	0.3100	0.2550	0.2450	0.2300	0.2300	0.2500	0.2500
P@20	0.1800	0.3525	0.2850	0.2925	0.3100	0.3150	0.3050	0.3125
P@30	0.1683	0.3467	0.3217	0.3267	0.3183	0.3383	0.3367	0.3300
MAP	0.0502	0.1883	0.1674	0.1726	0.1763	0.1744	0.1754	0.1754
Win/loss	N/A	19/1	18/2	18/2	18/2	19/1	19/1	19/1

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

Indri:mu = 1000

Indri:lambda = 0.7

fbDocs = 10 [Best parameter from Exp2]

fbTerms = 40 [Best parameter from Exp3]

fbMu = 0

(1) 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?

From my experiments, the more weight the original query, the better the result. When original query weigh 1.0, the result is the best. Thus the combination of the two queries is not worthwhile at least for this dataset. The original query not only has better result but also save a lot of running time. The more weight the original query count, the more stable the behavior is.

(2) Comment on the effects of the combined query on your system and on the reference system. Are the two systems affected equally, or are there important differences?

My system and the refenrence system are affected equally. They have the same trend – the more the original weight count, the better the performance is, and the more stability the system has.

5 Experiment 5: Effect of the original query quality

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

	Ranked Boolean	Query Expansion, Your Initial Results	
		BOW Original Query	**SDM Original Query**

	AND	Original	Expanded	Original	Expanded
P@10	0.1300	0.3000	0.2900	0.3900	0.3800
P@20	0.1800	0.3325	0.3250	0.4200	0.3900
P@30	0.1683	0.3467	0.3267	0.4200	0.4100
MAP	0.0502	0.1813	0.1788	0.2124	0.1989
Win/loss	N/A	19/1	18/2	19/1	18/2

	Ranked Boolean AND	**Query Expansion, Reference System Initial Results**			
		BOW Original Query		**SDM Original Query**	
		Original	**Expanded**	**Original**	**Expanded**
P@10	0.1300	0.3100	0.2900	0.4000	0.4000
P@20	0.1800	0.3525	0.3150	0.4200	0.4300
P@30	0.1683	0.3467	0.3267	0.4400	0.4400
MAP	0.0502	0.1883	0.1745	0.2231	0.2235
Win/loss	N/A	19/1	18/2	19/1	19/1

Document the values of the parameters used for this experiment.

Indri:mu = 1000

Indri:lambda = 0.7

fbDocs = 10 [Best parameter from Exp2]

fbTerms = 40 [Best parameter from Exp3]

fbOrigWeight = 0.8

fbMu = 0

SDM: 0.3, 0.4, 0.3.

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

The difference in query expansion effectiveness and stability between my system and the reference system is very minor. The reference system is a little bit more stable than my system for SDM experiment, but the difference is only one query.

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.

(1) 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 above, query expansion is detrimental to top-ranked documents. As expansion queries include some non-relevant terms and decrease the weight of the original terms, the precision of the query at top ranking is decreased. The negative effect is extremely apparent when the original query well describes the users information needs. However, for ambiguous query such as “djs” and “avp”, the expanded query gives better performance.

On the other hand, query expansion usually increases the recall and has a positive effect on the bottom-ranked documents. The reason behind this is that query expansion add synonyms, morphological terms of the original terms. Those terms help to cover a larger range of users information needs and thus help improve recall. However, when terms are added to original query, some of them are meaningless such as “I”, “you” or number 1, 2, 3 etc in previous examples.

From the experiment, query expansion has greatest impact at P@20. The first 10 documents are not affected much as the original terms are also included in expanded query and has large weight in most cases. After 10th document, the ranking is affected much by the expanded query as the expanded words reduce the weight of original term.

(2) 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?

The win/loss ratio is from 17/3 to 19/1, so the system is quite stable in general. Since the worst win/loss ratio is still 17/3, which is not very bad, I can’t see there is any special conditions in terms of stability. Usually, the higher MAP and P@10/20/30, the better stability is.

(3) 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.

Query expansion help improve “recall” but slow the process time. For scenario that needs explore many documents and time-insensitive, query expansion may be worthwhile. But for scenario that time is the main concerns and only the top 3 or 5 documents are needed, query expansion is not worthwhile.

(4) 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.

I have a guess that query expansion helps in diversification. (Well, only a guess). I examined the detail report of the experiment and found that for query 15:espn sports, both precision and MAP improved dramatically (MAP from 0.1555 to 0.4116) after using query expansion. The reason behind might be that “espn sports” is a query that contains info need from different aspect, and the expanded terms complete the meaning of this phrase, such as adding terms “soccor” “basketball” “standing” “university” etc.