Your Name: Silun Wang

Your Andrew ID: silunw

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: Silun Wang

Your Andrew ID: silunw

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	0.3250	0.2500	0.3100	0.2250	0.2650
P@20	0.3675	0.3125	0.3525	0.2875	0.3125
P@30	0.3700	0.3300	0.3476	0.3033	0.3185
MAP	0.1881	0.1754	0.1883	0.1664	0.1746
win/loss	N/A	11/9	15/5	12/8	12/8

```
Indri:mu=1000
Indri:lambda=0.7
```

Overall, the quality, characters and weights of the query expansion seem reasonable. Take "10: cheap internet" for example, the expanded query is:

```
10: #WAND (0.0136 package 0.0146 provider 0.0149 service 0.0152 fast 0.0171
dial 0.0298 domain 0.0541 web 0.0817 host 0.0902 internet 0.1535 cheap)
```

"Cheap" and "internet" remains high weights, and other closely related terms are included in the expanded query. Also notice that some unexpected terms may show up in the expanded query, take "ps 2 games" for example:

```
#WAND(0.0733 ps 0.0719 ps2 0.0709 games 0.0572 2 0.0515 ago 0.0486 ps2dvd
0.0318 pal 0.0315 torrent 0.0289 month 0.0235 com )
```

They are slightly different. Overall, the reference system behaves a little better than mine in terms of P@N and MAP. They behave the same at win/loss rate. Query expansion negatively affects the accuracy of both the two systems. We can see from the table that P@N and MAP drop as an effect of 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, Your System	Query Expansion, Your Initial Results					
			Feedback Documents					
			10	20	30	40	50	100
P@10	0.3250	0.2500	0.2250	0.2350	0.2250	0.2300	0.2400	0.2550
P@20	0.3675	0.3125	0.2875	0.2925	0.2975	0.2950	0.2925	0.2925
P@30	0.3700	0.3300	0.3033	0.3033	0.3167	0.3100	0.3167	0.3033
MAP	0.1881	0.1754	0.1664	0.1670	0.1659	0.1678	0.1672	0.1627
win/loss	N/A	11/9	12/8	12/8	12/8	11/9	11/9	10/10

	Ranked Boolean AND	Indri BOW, Reference System	Query Expansion, Reference System Initial Results					
			Feedback Documents					
			10	20	30	40	50	100
P@10	0.3250	0.3100	0.2650	0.2650	0.2300	0.2450	0.2600	0.2650
P@20	0.3675	0.3525	0.3125	0.3125	0.2950	0.3050	0.3075	0.3075
P@30	0.3700	0.3476	0.3185	0.3183	0.3183	0.3167	0.3217	0.3117
MAP	0.1881	0.1883	0.1746	0.1746	0.1683	0.1722	0.1735	0.1732
win/loss	N/A	15/5	12/8	12/8	11/9	11/9	11/9	11/9

```
Indri:mu=1000
Indri:lambda=0.7
fb=true
fbTerms=10
fbMu=0
fbOrigWeight=0.5
```

No value is consistently better than other values. We can also see from the table that using more documents does not necessarily help the results. My system experiences a slight increase at P@10 and MAP as we use more documents, it then experiences a slight drop at MAP and win/loss rate. For the reference system, using more feedback documents makes MAP slightly drops, but overall it's rather stable.

Reason for this might be that top-ranked (especially top 20) documents has a better quality, hence the terms extracted from these documents are more stable and likely to improve results than low-ranked documents.

For fbDocs > 20, the improvement does not worth the added computational cost.

Effects of query expansion on your system and on the reference system are only slightly different. Like in experiment 2, reference system still behaves a little better than mine in terms of P@N and MAP, nearly the same in win/loss rate. As fbDocs reaches 20, increasing the document number can negatively affect the accuracy of both two systems. We can see from the table that P@N and MAP drop as an effect of using more than 20 retrieved documents.

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	0.3250	0.2500	0.2250	0.2250	0.2250	0.2200	0.2150	0.2150
P@20	0.3675	0.3125	0.2825	0.2875	0.2825	0.2800	0.2900	0.2925
P@30	0.3700	0.3300	0.3033	0.3033	0.3017	0.3083	0.3117	0.3100
MAP	0.1881	0.1754	0.1616	0.1664	0.1677	0.1694	0.1692	0.1694
Win/loss	N/A	11/9	10/10	11/8	11/9	13/7	13/7	13/7

	Ranked Boolean AND	Indri BOW, Reference System	Query Expansion, Reference System Initial Results					
			Feedback Terms					
			5	10	20	30	40	50
P@10	0.3250	0.3100	0.2350	0.2400	0.2400	0.2350	0.2350	0.2350
P@20	0.3675	0.3525	0.2825	0.3000	0.3075	0.3050	0.3050	0.3050
P@30	0.3700	0.3476	0.3050	0.3233	0.3283	0.3333	0.3317	0.3333
MAP	0.1881	0.1883	0.1647	0.1722	0.1741	0.1760	0.1761	0.1768
Win/loss	N/A	15/5	11/9	11/9	12/8	13/7	13/7	13/7

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

```
Indri:mu=1000
Indri:lambda=0.7
fb=true
fbDocs=10
fbMu=0
fbOrigWeight=0.5
```

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?

No value is consistently better than other values. And using more and more terms does not always help the result consistently. Because as we increase the number of feedback terms, top ranked terms are generally of good quality, but the quality of the low-ranked terms are not as good as top ones. So more terms can result in poor retrieval results.

The following is 30 terms expanded from “cheap internet”. They end up with some unexpected and poor-qualified terms like “satellite” and “uk”.

```
#WAND(0.4683 cheap 0.4432 internet 0.0864 provider 0.0785 service 0.0783 host
0.0698 dial 0.0632 access 0.0595 fast 0.045 server 0.044 domain 0.0438 up
0.0436 web 0.0404 dsl 0.0399 expry 0.0354 name 0.0342 speed 0.0261 high
0.0254 date 0.0252 save 0.0252 map 0.023 free 0.0226 cable 0.0222 broadband
0.0217 phone 0.0195 world 0.0187 card 0.0186 uk 0.0183 satellite 0.0172 isp
0.0167 site )
```

Computation cost dramatically increases as we use more terms. From the table, we can see that P@N, MAP and win/loss rate only slightly improves after using more terms. So the improvement does not worth the added computational costs.

Effects of using more terms on your system and on the reference system are only slightly different. Reference system behaves a little better than mine in terms of P@N and MAP, nearly the same in win/loss rate. As fbTerms reaches 30, increasing the number does not improve the accuracy of both two systems.

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	0.3250	0.2500	0.2150	0.2250	0.2300	0.2200	0.2250	0.2500
P@20	0.3675	0.3125	0.2800	0.2925	0.3025	0.3075	0.3050	0.3125
P@30	0.3700	0.3300	0.3217	0.3150	0.3100	0.3317	0.3317	0.3300
MAP	0.1881	0.1754	0.1600	0.1655	0.1686	0.1699	0.1729	0.1754
Win/loss	N/A	11/9	10/10	11/8	12/8	13/7	13/7	11/9

	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	0.3250	0.3100	0.2800	0.2900	0.2800	0.2500	0.2450	0.2500
P@20	0.3675	0.3525	0.3025	0.3100	0.3225	0.3250	0.3100	0.3125
P@30	0.3700	0.3476	0.3283	0.3167	0.3183	0.3383	0.3400	0.3300
MAP	0.1881	0.1883	0.1711	0.1772	0.1791	0.1758	0.1763	0.1754
Win/loss	N/A	11/9	10/9	11/9	13/7	13/7	13/7	11/9

```
Indri:mu=1000
Indri:lambda=0.7
fb=true
fbDocs=20
fbTerms=30
fbMu=0
```

Regarding my system, the table shows MAP is the highest when fbOrigWeight equals 1.0, which means that a combination of the two queries is not worthwhile. As for the reference system, MAP is the highest when fbOrigWeight is around 0.4, which means a combination is worthwhile. Because my initial retrieval results are not as good as the reference one. This implies that the initial ranking file is important for query expansion. If we are able to choose from a clearer, better-qualified, more representative source (e.g. Wikipedia), the query expansion is more likely to positively affect the accuracy. Also, under such circumstances, a combination also makes the stability better. (See the second table, win/loss rate increases for weight 1.0 → 0.4)

Not equally affected. In this experiment, a combination of the two queries negatively affect my system, but it can positively affect the reference system because of the good initial ranking file. Actually, setting weight = 0.4 gets the highest MAP for the reference system.

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, Your Initial Results			
		BOW Original Query		SDM Original Query	
		Original	Expanded	Original	Expanded
P@10	0.3250	0.3200	0.2650	0.4150	0.4550
P@20	0.3675	0.3600	0.3400	0.4525	0.4650
P@30	0.3700	0.3467	0.3617	0.4550	0.4883
MAP	0.1881	0.1865	0.1975	0.2464	0.2826
Win/loss	N/A	15/5	16/4	14/6	16/4

	Ranked Boolean AND	Query Expansion, Reference System Initial Results			
		BOW Original Query		SDM Original Query	
		Original	Expanded	Original	Expanded
P@10	0.3250	0.3100	0.3150	0.2650	0.3500
P@20	0.3675	0.3525	0.3550	0.3125	0.4175
P@30	0.3700	0.3476	0.3467	0.3185	0.4083
MAP	0.1881	0.1883	0.1974	0.1746	0.2237
Win/loss	N/A	15/5	15/5	12/8	15/5

```
Indri:mu=500
Indri:lambda=0.1
SDM: 0.25 AND 0.50 NEAR 0.25 WINDOW
fbDocs=20
fbTerms=30
fbMu=0
fbOrigWeight=0.4
```

Yes. A better-qualified initial retrieval can positively affect P@N, MAP and win/loss rate. It also helps improve stability. Usually, query expansion negatively affects my system. But in this case, to my surprise, query expansion dramatically improves my SDM original query.

6 Analysis of results

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?

Query expansion can negatively affect the "high Precision" while positively affect the "high Recall" portion. It has the greatest impact for MAP. (Only if the initial ranking file is of good quality)

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?

Not stable. Using a better-qualified initial ranking file helps increase the stability. Usually, better stability implies better accuracy metrics.

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.

For the reference system, yes. For my system, no. If the initial ranking file's quality is not guaranteed, it is better to avoid query expansion in BOW.