Your Name: Sz-Rung Shiang

Your Andrew ID: sshiang

Homework 3

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.

   No, for the part of sorting Hashmap, which is from the website StackOverFlow. (http://stackoverflow.com/questions/8855849/sorting-a-hashmap-while-keeping-duplicates)

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

   Yes.

Your Name: Sz-Rung Shiang

Your Andrew ID: sshiang

Homework 3

Instructions

1 Experiment 1: Baselines

	Ranked Boolean AND	Indri			
		BOW		Query Expansion	
		Your System	Reference System	Your System	Reference System
P@10	0.2500	0.3400	0.3500	0.3400	0.3650
P@20	0.3125	0.3975	0.4075	0.3725	0.3750
P@30	0.3433	0.4017	0.4033	0.3867	0.3833
MAP	0.1146	0.1994	0.2029	0.1933	0.2063
win/loss	N/A	14/6	15/5	14/6	14/6

1.1 Parameters

The following is the parameters I used for experiment 1:

- Retrieval Algorithm: Indri
- Mu:2500
- Lambda: 0.4
- Fb:mu: 0.0
- FbDocs: 10
- FbTerms: 10
- FbOrigWeight: 0.5

For the win/loss calculation, I used MAP as criterion for counting the number of queries which win/lose compared to the AND system.

1.2 Discussion

In table 1, there are two sets of systems: my system and reference system. In the columns for first-pass results, we can see that the reference system performs better than my system does. Based on different systems, the results of query expansion using Indri show that the one based on reference system outperforms the one based on my system. In addition, the one of relevance feedback using reference system gets better results than both reference system and my system in P@10 and MAP, while my system with relevance feedback performs even worse than my system without relevance feedback and the reference system. The intuition behind is that the results of relevance feedback highly depend on the

quality of the first-pass results. If the first-pass results are not good enough, it's possible that the results will be even worse. Another point is that the relevance feedback doesn't guarantee to improve precision, but it might be helpful for MAP, which is also related to recall.

The win/loss evaluation metric shows how many queries achieve better performance after relevance feedback compared to the #AND system. As we can see that it's not always the case adding relevance feedback can get improvement.

The relevance feedback might not be always effective. For query with ambiguous information need, the performance can be improved; for example, the query "pvc" is related to plastic material but what's the exact intention for making this query is not clear. It may be related to material or recycling. Using relevance feedback makes the new query includes terms like: "plastic", "pip" or "sheet", which may bring more information (but not necessary). On the other hand, relevance feedback of query "neil young" is less effective because the expansion terms are like "music" and "trail" which is more ambiguous.

2 Experiment 2: The number of feedback documents

	Ranked Boolean AND	Indri BOW, Reference System	Query Expansion, Reference System Initial Results					
			Feedback Documents					
			10	20	30	40	50	100
P@10	0.2600	0.3400	0.3650	0.3650	0.3850	0.4300	0.3850	0.3700
P@20	0.3125	0.3975	0.3750	0.3975	0.4050	0.4525	0.4100	0.4050
P@30	0.3433	0.4017	0.3833	0.3867	0.4000	0.4383	0.3900	0.4017
MAP	0.1146	0.1994	0.2063	0.2185	0.2163	0.2244	0.2146	0.2121
win/loss	N/A	14/6	14/6	11/9	13/7	14/6	13/7	13/7

2.1 Parameters

The following is the fixed parameters I used for experiment 2:

- Retrieval Algorithm: Indri
- Mu:2500
- Lambda: 0.4
- Fb:mu: 0.0
- FbTerms: 10
- FbOrigWeight: 0.5

For the win/loss calculation, I used MAP as criterion for counting the number of queries which win/lose compared to the AND system for Indri BOW and compared to the reference system for other columns.

2.2 Discussion

In this section, experiment of different number of documents for relevance feedback is conducted. The best performance is achieved when the number of document is 40, and it related to how many possible information needs or synonyms. The number also depends on how many relevant documents there are in

the database, and how the system performs. The most important point for choosing this parameter is to choose a threshold that can have larger relevant document ratio. Recall that the precision@20 and precision @30 usually higher than precision@10, which means that a lot of relevant documents might be in between the rank from 10 to 30; therefore, using more than 10 documents as relevance feedback source is reasonable.

For the query "Korean language" and "trombone for sale", the result with relevance feedback on 40 documents outperforms the results with only 10 documents. On the other hand, for the query "neil young" which is a specific person, the number of relevant document is only a few and therefore using less documents is better. In addition, P@5 and P@10 metrics are 1.0 and therefore using relevance feedback on these documents is a strong evidence to retrieve more relevant documents.

3 Experiment 3: The number of feedback terms

	Ranked Boolean AND	Indri BOW, Reference System	Query Expansion, Reference System Initial Results					
			Feedback Terms					
			5	10	20	30	40	50
P@10	0.2600	0.3400	0.3550	0.3650	0.4150	0.4350	0.4450	0.4500
P@20	0.3125	0.3975	0.4025	0.3750	0.4700	0.4750	0.4700	0.5333
P@30	0.3433	0.4017	0.4217	0.3833	0.4383	0.4433	0.4400	0.4517
MAP	0.1146	0.1994	0.2020	0.2063	0.2330	0.2376	0.2399	0.1794
Win/loss	N/A	14/6	13/7	14/6	14/6	16/4	15/5	5/15

3.1 Parameters

The following is the fixed parameters I used for experiment 3:

- Retrieval Algorithm: Indri
- Mu:2500
- Lambda: 0.4
- Fb:mu: 0.0
- FbDocs: 40
- FbOrigWeight: 0.5

For the win/loss calculation, I used MAP as criterion for counting the number of queries which win/lose compared to the AND system for Indri BOW and compared to the reference system for other columns.

3.2 Discussion

In this section, different number of relevance feedback terms is shown. The result using 40 relevance feedback terms gets the best performance. The number of terms needed depends on the type of the query: If the query is with ambiguous or there might be a lot of synonyms for the query terms, then relevance feedback is necessary to modify the query.

For example, for the query with less ambiguity "sf bart", the system with 5 terms (1.0 in P@10) outperforms the system with 50 terms (0.9 in P@10). The need of query expansion is non-necessary because the information need is very clear and adding other terms may make the query ambiguous.

Another query "becoming a paralegal" is an example showing the trade-off of precision and recall. The information need is clear, however, there might be some synonyms for the query to represents the same terms. For P@5 performance the system with 5 expansion terms performs best, while for P@30 the system with 30 expansion terms performs best. It shows that the strict query may help to get results with high precision but low recall; on the other hand, adding additional terms may lead to change in the information need of the query but it may be helpful to increase the recall.

For ambiguous query "Korean language", it shows that the system with 40 terms (1.0 in P@10) performs much better than the system with 5 terms (0.5 in P@10). For this kind of query with ambiguous information need, relevance feedback can help to improve the system.

In addition to the type of query, the number of relevant documents in the system might be another factor for judging the usefulness of relevance feedback. If the number of relevant document that can be retrieved by matching the original query is high enough, then there is no much need for query expansion.

4 Experiment 4: Original query vs. expanded 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	0.2600	0.3400	0.4300	0.4450	0.4550	0.4300	0.4050	0.3400
P@20	0.3125	0.3975	0.4475	0.4550	0.4775	0.4550	0.4275	0.3975
P@30	0.3433	0.4017	0.4433	0.4350	0.4350	0.4367	0.4200	0.4017
MAP	0.1146	0.1994	0.2491	0.2459	0.2432	0.2340	0.2164	0.1994
Win/loss	N/A	14/6	13/7	14/6	17/3	16/4	14/6	10/10

4.1 Parameters

- Retrieval Algorithm: Indri
- Mu:2500
- Lambda: 0.4
- Fb:mu: 0.0
- FbDocs: 40
- FbTerms: 40

For the win/loss calculation, I used MAP as criterion for counting the number of queries which win/lose compared to the AND system for Indri BOW and compared to the reference system for other columns.

4.2 Discussion

In this section, results of different fbOrigWeight parameters are shown. The results with parameter as 0 is the one using completely expanded query, and the one with parameters as 1 is the one using completely original query. The guideline of choosing the parameter is to balance the two terms: the larger the parameter, the more the system relies on the expanded query. If the query terms are not suitable for query expansion, such as proper nouns, then higher value may be harmful, while if the query terms are general terms, a higher value should be good. In our experiment results, the query "penguins", which is a general term with ambiguous information need, with parameter value as 0.8 (0.2 in P@10) performs better than parameter value as 0.2 (0 in P@10). For the query "neil young", which is a person name, with parameter value as 0.2 (1.0 in P@10) performs better than parameter value as 0.8 (0.8 in P@10). Based on this finding, a possible improvement for the relevance feedback system is to use variant parameter for this parameter.

5 Experiment 5: Smoothing on longer queries

	Indri BOW, Reference System	**Query Expansion, fbTerms = 10**					
		μ					
		1500	**2000**	**2500**	**3000**	**4000**	**5000**
P@10	0.3400	0.4150	0.4250	0.4300	0.4150	0.3800	0.3600
P@20	0.3975	0.4475	0.4575	0.4525	0.4400	0.4450	0.4375
P@30	0.4017	0.4400	0.4367	0.4383	0.4317	0.4467	0.4467
MAP	0.1994	0.2285	0.2316	0.2244	0.2250	0.2200	0.2173
Win/loss	14/6	13/7	12/8	16/4	15/5	15/5	16/4

	Indri BOW, Reference System	**Query Expansion, fbTerms = 20**					
		μ					
		1500	**2000**	**2500**	**3000**	**4000**	**5000**
P@10	0.3400	0.4500	0.4450	0.4350	0.4300	0.4200	0.3950
P@20	0.3975	0.4500	0.4625	0.4625	0.4675	0.4475	0.4525
P@30	0.4017	0.4350	0.4383	0.4333	0.4350	0.4450	0.4483
MAP	0.1994	0.2334	0.2352	0.2349	0.2362	0.2312	0.2273
Win/loss	14/6	14/6	14/6	15/5	16/4	16/4	14/6

	Indri BOW, Reference System	**Query Expansion, fbTerms = 30**					
		μ					
		1500	**2000**	**2500**	**3000**	**4000**	**5000**
P@10	0.3400	0.4600	0.4500	0.4350	0.4450	0.4500	0.4400
P@20	0.3975	0.4550	0.4675	0.4725	0.4725	0.4575	0.4650
P@30	0.4017	0.4567	0.4433	0.4433	0.4433	0.4500	0.4533
MAP	0.1994	0.2379	0.2403	0.2396	0.2398	0.2362	0.2360
Win/loss	14/6	15/5	16/4	15/5	14/6	15/5	16/4

5.1 Parameters

- Retrieval Algorithm: Indri
- Lambda: 0.4
- Fb:mu: 0.0
- FbDocs: 40
- FbOrigWeight: 0.4

5.2 Discussion

The parameter *mu* a smoothing factor especially for document length. As the value gets higher, the longer documents can get more benefits to compete with shorter documents. Different from the results of previous homework, the system achieves best performance when mu is large (mu is 8000 in the previous experiment.); however, we achieve opposite results from that. The reason might be that if the documents are very long, then the matching terms might not be representative of the documents. For example, if there are some documents about "pvc" with similar count of matching tokens, and there is a one containing 100 terms and another containing 10000 terms in the documents, we can infer that "pvc" might be more likely to be a keyword or representative in the short documents, and other terms in the documents might be more relevant to our original query term. For long documents, "pvc" might be only the sub-topic in one paragraph and thus other terms are not necessary to be related to the original query terms. As a result, choosing lower mu can help the system to select the shorter documents to be relevance feedback source.

In our experiment part, as the number of relevance feedback terms is low (fbTerms=10), the large value mu (mu=5000) is the worst, while as the number of relevance feedback terms is high (fbTerms=30), the value of mu doesn't affect that much even though lower mu still achieve better results. It means that if we can only choose relevant terms from small set of documents, then the system has to be more carefully judging which documents are relevant and selected. As the number of fbTerms increases, the system becomes more robust.