Your Name: Wei Wang

Your Andrew ID: ww5

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.

   If you answered Yes, provide the name(s) of anyone who provided help, and describe the type of help that you received.
   No

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

   If you answered Yes, provide the name(s) of anyone that you helped, and describe the type of help that you provided.
   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.

   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.
   Yes

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

   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.
   Yes

Your Name: Wei Wang

Your Andrew ID: ww5

Homework 3

1 Experiment 1: Baselines

	Ranked Boolean AND	Indri			
		BOW		Query Expansion	
		Your System	Reference System	Your System	Reference System
P@10	0.4000	0.4850	0.4950	0.4800	0.4600
P@20	0.3675	0.4375	0.4425	0.4550	0.4425
P@30	0.3417	0.4267	0.4350	0.4400	0.4267
MAP	0.1071	0.1335	0.1358	0.1359	0.1324
win/loss	N/A	Win	Win	Win	Loss

1.1 Parameters

I kept Indri parameters constant: mu=2500, lambda=0.4; All the rest of fb parameters are the same as baseline provided in the HW requirement: fb=true, fbDocs=10, fbTerms=10, fbMu=0, fbOrigWeight=0.5

1.2 Discussion

The Indri results are all better than Ranked Boolean AND results. When introducing parameters mu and lambda, the indri model will smooth the query. Also, the indri query model considered document length, collection term frequency etc. In the query expansion category, my own system performs better than reference system as initial rank document. This can cause by different rank result that we produce. In my expanded result, there might be adding up some more relevant terms to topic.

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	0.4000	0.4950	0.4600	0.4700	0.4750	0.4750	0.4750	0.4700
P@20	0.3675	0.4425	0.4425	0.4475	0.4500	0.4575	0.4825	0.4650
P@30	0.3417	0.4350	0.4267	0.4300	0.4350	0.4583	0.4850	0.4633
MAP	0.1071	0.1358	0.1324	0.1345	0.1345	0.1371	0.1391	0.1380
win/loss	N/A	Win	Win	Win	Win	Win	Win	Win

2.1 Parameters

Constant Variable: retrievalAlgorithm=Indri; Indri:mu=2500; Indri:lambda=0.4; fb=true; fbTerms=10; fbMu=0; fbOrigWeight=0.5

2.2 Discussion

As the number of feedback documents increase, all precision numbers are increasing slowly, and they are all have a higher map than in Ranked Boolean AND. The p@10 is not increasing significantly after feedback documents reach 30, which means that when the search engine went through more than 30 documents, the top 10 documents have a stable precision rate. While for the rest of metrics value, they are keep increasing as number of documents increased.

Although the precision is keep rising, we cannot always set a high value for feedback documents. The computation cost is too high for large feedback documents. In my algorithm, I need to run through them for almost 2 hour if set fbdocs to be 100. It is relatively kept it as 30 in the real-world implementation.

Fbdocs=50→704: #wand(0.038800 party 0.014100 candidate 0.011500 politics 0.010200 view 0.009400 total 0.009000 map 0.007900 0 0.005500 san 0.005200 election 0.004800 john)

Fbdocs=30→704: #wand(0.022500 party 0.011500 politics 0.005200 candidate 0.004000 election 0.003100 view 0.003000 democratic 0.002700 republican 0.002700 total 0.002600 government 0.002500 voter)

For query 704, we can find that they are quite different after changing the fbdocs, as the number increase, there are more specific terms exist such as “John” and “san”, these information might be what the user wants or not, it depends. So generally, based on the computation cost, fbdocs=30 is quite reasonable.

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	0.4000	0.4950	0.4450	0.4600	0.4650	0.4750	0.4900	0.4850
P@20	0.3675	0.4425	0.4500	0.4425	0.4500	0.4475	0.4525	0.4475
P@30	0.3417	0.4350	0.4333	0.4267	0.4317	0.4400	0.4400	0.4333
MAP	0.1071	0.1358	0.1289	0.1324	0.1365	0.1348	0.1387	0.1388
Win/loss	N/A	Win	Win	Win	Win	Win	Win	Win

3.1 Parameters

Constant value: Indri:mu=2500; Indri:lambda=0.4; fb=true; fbDocs=10; fbMu=0; fbOrigWeight=0.5

3.2 Discussion

In my experiment output, feedback terms as 40 shows the best result. from fbterms=5 to 40, the precision and map is increasing, this is because as the feedback terms increased, more terms are included in the expanded query, they are supportive to the precision of user's information need. When the terms goes up to 50, more terms are included, they might not be necessarily related to information needed, but appear since they are in the top 10 relevant documents and has a higher scores. Therefore the precision goes down a little bit, while it is still pretty high compared to fbterms below 30.

As a example, we can compare same query with different expanded query (Query 721).

Fbterms=5→721: #wand(0.009500 data 0.008600 census 0.003900 application 0.003600 hmda 0.003300 item)

Fbterms=40→721: #wand(0.009500 data 0.008600 census 0.003900 application 0.003600 hmda 0.003300 item 0.003300 available 0.003200 use 0.003200 report 0.002600 area 0.002400 metropolitan 0.002400 county 0.002300 information 0.002300 institution 0.002200 censtat 0.002200 rom 0.002100 tract 0.002000 cd 0.002000 year 0.001900 1990 0.001900 aggregate 0.001900 code 0.001900 number 0.001900 zip 0.001800 household 0.001800 transportation 0.001800 detailed 0.001700 ma 0.001600 2000 0.001500 each 0.001500 income 0.001500 also 0.001500 years 0.001500 order 0.001400 ffiec 0.001400 model 0.001400 work 0.001400 from 0.001300 drive 0.001300 charge 0.001300 mhi)

By comparing the result, we can see that the later query include some words can support information need : "Census data applications". Such as household, year, tract, county, metropolitan, income, etc. These words are very closely related to census data, some of them are the unit of census tract, some of them are the attributes of census data.

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	0.4000	0.4950	0.4400	0.4500	0.4850	0.4850	0.4800	0.4850
P@20	0.3675	0.4425	0.4300	0.4475	0.4500	0.4600	0.4550	0.4375
P@30	0.3417	0.4350	0.4300	0.4350	0.4450	0.4383	0.4383	0.4267
MAP	0.1071	0.1358	0.1311	0.1435	0.1381	0.1374	0.1353	0.1335
Win/loss	N/A	Win	Win	Win	Win	Win	Win	win

4.1 Parameters

Indri:mu=2500; Indri:lambda=0.4; fb=true; fbDocs=10; fbTerms=40; fbMu=0; fbOrigWeight=0.5; fbInitialRankingFile=TEST_DIR/Indri-Bow.fbRank

From the experiment 3, I learnt that when fbTerms is 40, the results are the best, therefore in this section I kept fbTerms as 40 and I assume that it can also work best for the above queries.

4.2 Discussion

This section analyzes how much weight we should put on the original weight and the expanded weight. The conclusion of my experiment is around 0.5, while they are all actually lower than the referenced system. Throughout the experiment, I found that from 0 to 0.4 or 0.6, the precision is going up, and from 0.6 to 1, the precision is going down. This indicates that if we put too much weight on the original query, the expanded query will have few or no impact on the results.

5 Experiment 5: Smoothing on longer queries

	Indri BOW, Reference System	**Query Expansion, fbTerms = 10**					
		λ					
		0.4	**0**	**0.2**	**0.6**	**0.8**	**0.5**
P@10	0.4950	0.4600	0.4550	0.4600	0.4650	0.4450	0.4650
P@20	0.4425	0.4425	0.4475	0.4475	0.4275	0.4300	0.4375
P@30	0.4350	0.4267	0.4317	0.4300	0.4217	0.4033	0.4233
MAP	0.1358	0.1324	0.1313	0.1329	0.1295	0.1237	0.1307
Win/loss	N/A	Loss	Loss	Loss	Loss	Loss	loss

	Indri BOW, Reference System	**Query Expansion, fbTerms = 20**					
		λ					
		0.4	**0**	**0.2**	**0.6**	**0.8**	**0.5**
P@10	0.4950	0.4650	0.4800	0.4550	0.4550	0.4400	0.4650
P@20	0.4425	0.4500	0.4675	0.4650	0.4600	0.4425	0.4550
P@30	0.4350	0.4317	0.4467	0.4417	0.4267	0.4100	0.4317
MAP	0.1358	0.1365	0.1375	0.1361	0.1326	0.1262	0.1356
Win/loss	N/A	Win	Win	Win	Loss	Loss	Loss

	Indri BOW, Reference System	**Query Expansion, fbTerms = 30**					
		λ					
		0.4	**0**	**0.2**	**0.6**	**0.8**	**0.5**
P@10	0.4950	0.4750	0.4800	0.4800	0.4650	0.4450	0.4800
P@20	0.4425	0.4475	0.4725	0.4700	0.4550	0.4425	0.4500
P@30	0.4350	0.4400	0.4500	0.4417	0.4317	0.4150	0.4400
MAP	0.1358	0.1384	0.1383	0.1394	0.1352	0.1285	0.1373
Win/loss	N/A	Win	Win	Win	Loss	Loss	Loss

5.1 Parameters

Indri:mu=2500; fb=true; fbDocs=10; fbTerms=10 or 20 or 30; fbMu=0; fbOrigWeight=0.5

In this section, I still kept my fbDocs as 10 dues to the running speed. Based on experiment I think fbOrigWeight has an optimal weight as 0.5, therefore I kept it in this experiment.

5.2 Discussion

Compared to reference system, all results in fbterms=10 loss, for fbterms=20, lambda from 0 to 0.4 win, forfbterms=30, lambda from 0 to 0.4 win. In all three scenarios, the precision at lambda = 0 is relatively high. For lambda, if the query is short, the lambda should be small, if the query is long, lambda should be large. In our case, the query is the relatively short, so the smaller lambda is performing well, since the smaller the lambda is, the more weight is putting on the document term frequency.

If we compared vertically, we find that the larger fbterms have a better result, especially for lambda=0.2. This can be caused due to the longer the query is, the larger the maximum likelihood of query in collection, the document term frequency will be larger.