Xiaoyu He

xiaoyuh1

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.

   Yes. Yupin Huang told me that all experiment 4 and 5 should based on the previous experiments. So I redid all the experiments.

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

3. Yes. I shared my code of auto testing the trec eval service on the github.

4. 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

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

Xiaoyu He

xiaoyuh1

Homework 3

1 Experiment 1: Baselines

	Ranked Boolean AND	Indri			
		BOW		Query Expansion	
		Your System	Reference System	Your System	Reference System
P@10	0.4050	0.3450	0.3550	0.3400	0.3650
P@20	0.4500	0.3975	0.4075	0.3725	0.3750
P@30	0.4867	0.4017	0.4033	0.3867	0.3833
MAP	0.2078	0.1994	0.2029	0.1970	0.2063
win/loss	N/A	8	11	10	10

1.1 Parameters

I tried several combination of indri score, the result is as follows:

Indri Param	1000, 0.4	1000, 0.7	2500, 0.4	2500, 0.7
P@10	0.3300	0.3150	0.3450	0.2950
P@20	0.3600	0.3500	0.3975	0.3925
P@30	0.3550	0.3567	0.4017	0.3883
MAP	0.1855	0.1836	0.1994	0.1854

Combination 2500, 0.4 gives highest P@10 and MAP score, and the result is near the reference system. So I choose mu= 2500, lambda = 0.4 for my indri model.

1.2 Discussion

The Ranked Boolean AND yields the highest P@10, P@20, P@30, and MAP. The Ranked Boolean wins in every aspect. Without fine tuning the parameter, query expansion is a waste of computation.

Reference system with query expansion ranks the second palso performs good. For reference system, query expansion improves the results in the P@10 and MAP, but decrease in P@20 and P@30. Thus I think, in overall, query expansion decreases the precision, but the increase in recall compensates for the loss in precision, thus improves the MAP score.

For my indri system, query expansion lowers the P@X and MAP. Thus I think indri mu and lambda parameters have some indirect affect on the effectiveness of query expansion, so the query expansion in my system does not improve the result at all. Interestingly, my system with expansion win 10 query while without expansion only win 8 query. This shows that, query expansion improves the overall win rate, but greatly decrease the MAP score for some particular queries, thus the overall MAP for 20 queries decreases.

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	0.4050	0.3550	0.3650	0.3650	0.3850	0.4300	0.3850	0.3700
P@20	0.4500	0.4075	0.3750	0.3975	0.4075	0.4525	0.4100	0.4050
P@30	0.4867	0.4033	0.3833	0.3867	0.4000	0.4383	0.3900	0.4017
MAP	0.2078	0.2029	0.2063	0.2185	0.2164	0.2244	0.2146	0.2121
win/loss	N/A	11	10	11	10	10	10	10

2.1 Parameters

Indri mu = 2500, Indri lambda = 0.4, fbTerms = 10, fbMu = 0.0, fbInitialWeights = 0.5

2.2 Discussion

With feedback documents greater than 10, all query expansion tests begin to beat the Rank Boolean AND in the MAP score. But, except fbDoc=40 beats the Ranked Boolean in the P@10, other query expansion tests all suffers in the precision. We can infer from the test result that, query expansion expanded the query vocabulary, decrease the precision but improves the recall, thus improved the overall MAP score.

Feedback Documents at 40 gives best P@10, P@20, P@30, and MAP. However, the number of wins/loss are basically the same among different fbDocs.

This is the MAP score for the first query (single query instead of 20):

Boolean	10	20	30	40	50	100
0.3085	0.0847	0.0979	0.1177	0.2536	0.0867	0.0793

As we can see, except fbDocs = 40, others all gives very low score on query 1 compares to Ranked Boolean AND. For query 1, although Indri Expansion all loss, but fbDocs = 40 have much higher MAP score, thus makes its overall MAP score higher.

Generally speaking, from 10 to 40, increasing fbDocs improves the performance. Including more documents will collect more relevant terms from the top documents. However, beyond 40, the documents are becoming less relevant. The term collected from those “less relevant” documents are also less relevant. In this case, we have higher chance retrieving some irrelevant word which appears frequent in the documents. For example, if we retrieve all the documents, then we will learn the most “likely” words in the whole collection, instead of learning the characteristic of particular set of 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	0.4050	0.3550	0.3550	0.4300	0.4150	0.4350	0.4450	0.4500
P@20	0.4500	0.4075	0.4025	0.4525	0.4700	0.4750	0.4700	0.4750
P@30	0.4867	0.4033	0.4217	0.4383	0.4383	0.4433	0.4417	0.4400
MAP	0.2078	0.2029	0.2020	0.2244	0.2330	0.2377	0.2398	0.2396
Win/loss	N/A	11	11	10	10/9	12	11/8	12

3.1 Parameters

Indri mu = 2500, Indri lambda = 0.4, fbDocs = 40, fbMu = 0.0, fbInitialWeights = 0.5

3.2 Discussion

Except fbTerms=5, all other tests surpass the Rank Boolean AND. The performance of using only 5 feedback terms is similar to Indri BOW. That is, similar to not doing query expansion at all.

The general trend is, for P@10, P@20 and MAP, the system improves as we collect more terms.

As far as I understand, a good query expansion can find some other terms, to reflect more aspects of the query. fbTerms=5 have too less terms to be able to capture other aspects of the query. As the feedback term increases, the query terms start to contains some other useful terms.

However, P@30 peak at fbTerms=30, then starts to decline. It looks like that after fbTerms=30, we start to collect more garbage terms in the query term tail, which doesn't affect the P@10 and P@20 too much, but is misleading and harmful for P@30.

In terms of the win/loss, fbTerms=30,40,50 win one more query than fbTerms=5,10,20. The win/loss shows an increasing trend.

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	0.4050	0.3550	0.4400	0.4650	0.4700	0.4350	0.3950	0.3450
P@20	0.4500	0.4075	0.4475	0.4550	0.4675	0.4575	0.4275	0.3975
P@30	0.4867	0.4033	0.4483	0.4383	0.4400	0.4333	0.4217	0.4017
MAP	0.2078	0.2029	0.2507	0.2491	0.2440	0.2352	0.2149	0.1994
Win/loss	N/A	11	13	12	12	12	11	8

4.1 Parameters

Indri mu = 2500, Indri lambda = 0.4, fbDocs = 40, fbTerms=30, fbMu = 0.0

4.2 Discussion

The result is surprising that fbOrigWeight=0 ranks the best in turns of the MAP score. There is a steady improvement in the MAP score as the weight decreases.

P@10 and P@20 peak at fbOrigWeight=0.4. However, P@30 and MAP steadily decreases as we increase fbOrigWeight.

Thus we can infer that, more weights on the original query terms improves the precision, good query expansion greatly improves the recall. Even though P@10 and P@20 suffers from too low(zero) fbOrigWeight, but improved recall rate compensated for the loss in precision, thus improved the MAP score.

Too much weight on the original query is bad. High fbOrigWeight destroys the benefits of query expansion. Why the precision starts to decrease when fbOrigWeight > 0.4? This is very similar to the case where we set fbTerms=5. In this case, too small weight on the query expansion terms is unable to reflects other aspects of the query, thus unable to harness the power of query expansion.

5 Experiment 5: Smoothing on longer queries

	Indri BOW, Reference System	Query Expansion, fbTerms = 10					
		µ					
		500	1500	2500	3500	4500	5500
P@10	0.3550	0.4100	0.4000	0.4300	0.3850	0.3700	0.3650
P@20	0.4075	0.3900	0.4400	0.4525	0.4400	0.4450	0.4450
P@30	0.4033	0.3850	0.4317	0.4383	0.4383	0.4417	0.4467
MAP	0.2029	0.2115	0.2264	0.2244	0.2140	0.2137	0.2133
Win/loss	11	9	10	10	11	9	9

	Indri BOW, Reference System	Query Expansion, fbTerms = 20					
		µ					
		500	1500	2500	3500	4500	5500
P@10	0.3550	0.4000	0.4350	0.4150	0.4150	0.4050	0.3850
P@20	0.4075	0.3800	0.4525	0.4700	0.4575	0.4475	0.4400
P@30	0.4033	0.4033	0.4350	0.4383	0.4417	0.4400	0.4533
MAP	0.2029	0.2170	0.2298	0.2330	0.2291	0.2260	0.2160
Win/loss	11	11	11/8	10/9	10	11	9

	Indri BOW, Reference System	Query Expansion, fbTerms = 30					
		µ					
		500	1500	2500	3500	4500	5500
P@10	0.3550	0.4400	0.4450	0.4350	0.4400	0.4250	0.4200
P@20	0.4075	0.4325	0.4650	0.4750	0.4725	0.4625	0.4575
P@30	0.4033	0.4333	0.4483	0.4433	0.4483	0.4450	0.4533
MAP	0.2029	0.2259	0.2355	0.2377	0.2336	0.2315	0.2273
Win/loss	11	11	13	12	12	11	10

5.1 Parameters

Indri mu=2500, Indri lambda=0.4, fbDocs=40, fbMu=0.0, fbOrigWeight=0.5

5.2 Discussion

For different fbTerms, we achieved best MAP score around mu=2500. The behavior or P@10, P@20 and MAP score are similar, they all peak around mu=1500~2500.

P@30 behaves differently, for fbTerms=10,20,30, P@30 steadily increases as the mu increases. This suggests that higher mu increase the recall, regardless of how many feedback terms we are collecting.

For shorter queries, mu have bigger impact on the precision. Below is P@10 for different mu and fbTerms.

fbTerms	500	1500	2500	3500	4500	5500
10	0.4100	0.4000	0.4300	0.3850	0.3700	0.3650
20	0.4000	0.4350	0.4150	0.4150	0.4050	0.3850
30	0.4400	0.4450	0.4350	0.4400	0.4250	0.4200

As we can see from the above table, when fbTerms=0, increase mu from 2500 to 5500 dramatically decreases the P@10 from 0.43 to 0.36. However, when fbTerms=30, P@10 only decreases from 0.445 to 0.42. Thus we can infer that, when fbTerms are small, large mu greatly damaged P@10.

There is also a big gap when increase mu from 1500 -> 2500, fbTerms=10, P@10 increase from 0.40 -> 0.43, P@20 0.39 -> 0.44, P@30 0.38 -> 0.43. The same gap happens when increase mu from 500 -> 1500, fbTerms=20, P@10 increase from 0.40 -> 0.43, p@20 0.38 -> 0.45, P@30 0.40 -> 0.35.

However, for fbTerms=30, there is no huge gap like the previous one, the scores are comparatively stable as the mu changes. So I think, increasing fbTerms have some what same smoothing effect as increase mu. Increase fbTerms, with longer query terms, we eliminates the gap caused by mu, and make mu have less impact on the precision and recall.

Some other thoughts:

$$p(q_i \mid d) \quad = \frac{tf_{q_i,d} + \mu \, p_{MLE}(q_i \mid C)}{length(d) + \mu} \qquad p(q_i \mid d) = (1-\lambda) p_{MLE}(q_i \mid d) + \lambda p_{MLE}(q_i \mid C)$$

In this experiments, we are asked to report whether mu have different effect for long queries and short queries. But from the ppt slides, I think we should use lambda instead of mu, to change the smoothing effect for queries with different length. Mu is used to smooth the effect of document length. I don’t understand why we are testing mu instead of lambda. I would be grateful if you can explain this to me in the assignment feedback.