Your Name: Chih-Wei Chang

Your Andrew ID: cchang3

Homework 2

Collaboration and Originality

Your report must include answers to the following questions:

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. Teng Zhong discussed the greedy algorithm NEAR with me, and we figured out some edge case where different greedy implementation might give different result. This helped me find out a bug which didn't appear in HW1.
   Also, Cheng-Ta Chung pointed out the document for the QryEval's Idx API to me which helped me figure out how to retrieve the index statistic.

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: Chih-Wei Chang

Your Andrew ID: cchang3

Homework 2

1 Experiment 1: Baselines

	Ranked Boolean	BM25 BOW	Indri BOW
P@10	0.1700	0.4200	0.4000
P@20	0.2800	0.3500	0.4700
P@30	0.3367	0.3667	0.4233
MAP	0.1071	0.1985	0.2057

2 Experiment 2: BM25 Parameter Adjustment

2.1 k_1

	k_1							
	1.2	0	0.5	5	10	100	1,000	10,000
P@10	0.4200	0.1100	0.4200	0.4000	0.3600	0.2700	0.2100	0.2100
P@20	0.3500	0.1350	0.3400	0.3550	0.3150	0.2400	0.2150	0.2100
P@30	0.3667	0.1533	0.3733	0.3667	0.3267	0.2933	0.2700	0.2667
MAP	0.1985	0.0665	0.2056	0.1653	0.1536	0.1085	0.0989	0.0983

.

2.2 b

	b							
	0.75	0.00	0.15	0.3	0.45	0.6	0.90	1.00
P@10	0.4200	0.3900	0.3800	0.4600	0.4300	0.4000	0.3900	0.3900
P@20	0.3500	0.4450	0.4450	0.4350	0.4050	0.3650	0.3650	0.3600
P@30	0.3667	0.4600	0.4367	0.4067	0.4033	0.3867	0.3400	0.3500
MAP	0.1985	0.1977	0.2126	0.2206	0.2122	0.2082	0.1826	0.1703

2.3 Parameters

I would like to see the result of setting k1 to zero. That is, what would happen if we ignore term frequency completely? Also, I'm interested in exploring the possible change in the result regarding to small change in k1. Besides, since k1 is not a bounded parameter, I'd also like to check the result for extreme value of k1.

As for b, since we can only choose b from 0 to 1, the natural choices would be using the same step size and check whether there would be a pattern regarding to the change in b.

2.4 Discussion

First, one of the interesting points is that even a small value of k1 can give a dramatic improvement in the precisions, which implies the importance of term frequency. Also, since b is non-zero in this setting, the value of k1 not only affects how the algorithm gives the consideration to term frequency, but it also affects the involvement of document length normalization. Given that, we might infer that even a small penalty or normalization in the length of document can be really helpful. Another interesting point is that the change in precision is decreasing with respect to the increasing in k1. That is, small k1 might be more sensitive when comparing to large k1. This can be observed from changing k1 from 0.5 to 5 and changing k1 from 5 to 10. Finally, I experimented large value of k1, which might make all weights near zero, to check if it would behave the same as unranked model. I expected it to be roughly the same as unranked, but it turns out it is still better than unranked model.

As for b, the intuition of changing in b is corresponding to how we emphasize the normalization in document length. It is hard to argue which b is the best one. However, from the result, we might claim that small b generally works fine. The reason behind this might be that, although we don't want long document dominate the retrieval result, the truth is that longer document might indeed contain more information and thus are more likely be relevant to the information need.

3 Experiment 3: Indri Parameter Adjustment

3.1 μ

	μ							
	2500	0	1	3	5	10	1,000	100,000
P@10	0.4000	0.4300	0.4300	0.4300	0.4300	0.4400	0.4200	0.3700
P@20	0.4700	0.4100	0.4100	0.4050	0.4050	0.4100	0.4400	0.4750
P@30	0.4233	0.3933	0.3933	0.3867	0.3867	0.3933	0.4267	0.5033
MAP	0.2057	0.1952	0.1953	0.1967	0.1965	0.1967	0.2143	0.1805

3.2 λ

	λ							
	0.4	0.0	0.2	0.5	0.6	0.8	0.9	1.0
P@10	0.4000	0.4000	0.3900	0.4000	0.4000	0.3700	0.3500	0.0000
P@20	0.4700	0.4750	0.4750	0.4700	0.4550	0.4100	0.3950	0.0150
P@30	0.4233	0.4233	0.4233	0.4167	0.4000	0.3967	0.3867	0.0200
MAP	0.2057	0.2142	0.2112	0.2012	0.1962	0.1798	0.1641	0.0020

3.3 Parameters

The choice of mu indicates how strong is our prior belief, while the choice of lambda indicates whether we put more focus on the collection distribution or the per document distribution. Since lambda is bounded in 0 to 1, we choose lambda uniformly to check if there is any obvious pattern. Since mu is unbounded, we can choose arbitrary large value to see how the strong prior belief affect our result. Also, we can see whether mu=0, e.g. no prior belief, will have significant impact on the performance. Besides,

since choosing belief is somewhat subjective, and people might choose it differently, we would like to see if the result is sensitive to the choose of mu. If it is not sensitive, then finding a proper range that gives reasonable good result might be suffice.

3.4 Discussion

First thing we can observe from the results for different mu is that the variation in precision is small regarding to the change in mu. In some sense, this implies that we can safely the value of mu, as long as it can represent how strong our prior belief is. Also, small prior belief is pretty much no different from no prior belief. That means, small prior belief might not have enough impact on how we score the documents. Yet, that does not mean larger mu is better. We can see that small or even zero prior belief can outperform large mu. This might be the result that such kind of estimation will converge eventually no matter how we choose the mu. That is, for long document, the ratio of the term frequency over the length of the document will eventually dominate the first time (document distribution term). So even we don't plug the prior belief in, we can still get a roughly good estimation for the distribution. Additionally, too large mu might even hurt the performance. The reason behind this might be that a too strong prior will affect our estimation too much.

As for lambda, it put different emphasis on document distribution and collection distribution. For large lambda, we roughly ignore the distinction among different documents and we treat them the same. For small lambda, we treat each document differently and ignore the potential common characteristics among them. Although the result of the experiments suggests that small lambda might be better, this result might in fact be relevant to the size of the collection and the intrinsic characteristics for the corpus. For example, if the collection is very large, the common pattern shared by all the documents might be more useful than small size collection. On the other hand, if each document in our collection is very different from each other, putting too much emphasis on the collection distribution might turns out to hurt our retrieval performance.

4 Experiment 4: Different representations

4.1 Example Query

146:#AND(
#WSUM(0.02 sherwood.url 0.06 sherwood.keywords 0.02 sherwood.title 0.45 sherwood.inlink 0.45 sherwood.body)
#WSUM(0.02 regional.url 0.06 regional.keywords 0.02 regional.title 0.45 regional.inlink 0.45 regional.body)
#WSUM(0.02 library.url 0.06 library.keywords 0.02 library.title 0.45 library.inlink 0.45 library.body))

4.2 Results

	Indri BOW (body)	0.05 url 0.15 keywords 0.15 title 0.40 body 0.25 inlink	0.00 url 0.15 keywords 0.10 title 0.35 body 0.40 inlink	0.02 url 0.06 keywords 0.02 title 0.75 body 0.15 inlink	0.70 url 0.05 keywords 0.05 title 0.10 body 0.10 inlink	0.02 url 0.06 keywords 0.02 title 0.45 body 0.45 inlink
P@10	0.4000	0.4000	0.3800	0.4100	0.3400	0.4000
P@20	0.4700	0.3800	0.3550	0.4400	0.3250	0.3750
P@30	0.4233	0.3667	0.3467	0.4267	0.3400	0.3600
MAP	0.2057	0.1597	0.1540	0.1905	0.1337	0.1616

4.3 Weights

Since the baseline simply matches the queries by only the body field, our goal would be explore how distributing the weights to different field might give us some advantages. It is straightforward that body might be the most important part in a document, so we always give body the most higher weight. However, we would also like to explore how different fields might affect the performance. For example, inlink is one important features that might give us some information, but we need to be careful about assigning weight to them given that inlink is not as abundant as body field. Also, we explored an extreme case for high URL weights because some people put some amount of emphasis on URL when doing Search Engine Optimization (SEO).

4.4 Discussion

We originally thought that distributing weights to different fields should help us improve our performance. However, it turns out none of the attempts worked successfully. At first, we thought that some terms might only appear in keywords field but not in body field. In such case, giving keywords field some weight should help us discover some relevant document. Yet, it might due to the fact that terms written in keywords field are usually very diverse (the original author might try to cover as most as domain as possible), giving weight to keywords field eventually hurt the precisions. Another point is that giving body the significant weight than other fields can actually re-produce the baseline's performance. So we can conclude that the decreasing in precisions might result from giving unimportant fields too much attention. On the other hand, some people working on SEO would put some efforts on formatting their URL to make it more search engine friendly. For example, people might put their keywords or the title and part of their description to their URL. So we thought that giving URL a significant weight might be a way to match documents. Although such approach does not produce the best result, it still give a reasonably good performance which might inspire us to design proper matching mechanism toward the URL field.

5 Experiment 5: Sequential dependency models

5.1 Example Query

146:#wand(
0.8 #and(sherwood regional library)

0.15 #and(#near/1(regional library) #near/1(sherwood regional))
0.05 #and(#window/8(regional library) #window/8(sherwood regional)))

5.2 Results

	Indri BOW (body)	0.80 AND 0.15 NEAR 0.05 WINDOW	0.80 AND 0.05 NEAR 0.15 WINDOW	0.00 AND 0.50 NEAR 0.50 WINDOW	0.20 AND 0.40 NEAR 0.40 WINDOW	0.33 AND 0.33 NEAR 0.33 WINDOW
P@10	0.4000	0.4500	0.4700	0.5100	0.4900	0.4800
P@20	0.4700	0.5400	0.5250	0.5400	0.5500	0.5600
P@30	0.4233	0.5400	0.5200	0.5267	0.5400	0.5467
MAP	0.2057	0.2834	0.2790	0.2975	0.3026	0.3006

5.3 Weights

The choice of different weights for AND, NEAR, and WINDOW basically indicate that whether we emphasize terms matching, ordered locality, or unordered locality. Since NEAR and WINDOW somehow give roughly the same constraints (and even more strict) as AND. We thought it might be a good idea to try to give the major weights to them, and only partial weight to AND. This means that NEAR and WINDOW can help us cover some matching that was originally provided by AND. AND should serve as an augmentation to these two operators.

5.4 Discussion

One thing to note is that even just distributing some minor weights to NEAR and WINDOW can significantly improve the performance. This might due to the fact that AND might mismatch irrelevant documents just because they contain query terms that spread in the long content. As a result, applying some constraints for the locality by NEAR and WINDOW could give us some advantages. Also, we can observe that either emphasizing NEAR or emphasizing WINDOW can give us the similar result, which might imply that when giving only minor weights, there is no too much difference between NEAR and WINDOW.

Next thing we explored is the result of eliminating AND completely. Presumably, NEAR and WINDOW can match the documents which would originally be matched by AND. The results confirmed this conjecture in that even without the AND, the precision can still be higher than the baseline. Also, we experimented that putting major weights on NEAR and WINDOW and only minor weight to AND. The result suggests that NEAR and WINDOW has covered what AND would cover, but adding a small amount of weight to AND can still boost the retrieval performance. Besides, the last experiment showed that evenly distributed weights to these three operators performed roughly the same as only NEAR and WINDOW would do. This might reveal that there could be a threshold for the weights of NEAR and WINDOW, where once passing the threshold, AND would no longer be able to give significant impact.