Your Name:Yiding Ou

Your Andrew ID: yidingo

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

2. Did you give help of any kind to anyone in developing their software for this assignment (Yes or No)?
   No
   If you answered Yes, provide the name(s) of anyone that you helped, and describe the type of help that you provided.

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
   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.

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

Your Name: Yiding Ou

Your Andrew ID: yidingo

Homework 2

Instructions

1 Experiment 1: Baselines

	Ranked Boolean	BM25 BOW	Indri BOW
P@10	0.0400	0.3700	0.4900
P@20	0.0800	0.3550	0.4250
P@30	0.0867	0.3400	0.3833
MAP	0.0079	0.0614	0.0973

2 Experiment 2: BM25 Parameter Adjustment

2.1 k_1

	k_1							
	1.2	0	10	40	0.5	0.8	1	1.4
P@10	0.3700	0.2000	0.2800	0. 1100	0. 4500	0.3900	0.3800	0.3700
P@20	0.3550	0.2250	0.2150	0. 1350	0. 4100	0. 3800	0.3650	0.3500
P@30	0.3400	0.2367	0.1967	0. 1600	0. 3700	0. 3433	0.3367	0.3300
MAP	0.0614	0.0540	0.0315	0. 0136	0. 0676	0. 0636	0.0620	0.0604

.

2.2 b

	b							
	0.75	0	0.25	0.5	1	0.15	0.2	0.1
P@10	0.3700	0.3500	0.5200	0.4800	0.2400	0.4800	0.4900	0.4800
P@20	0.3550	0.3800	0.4700	0.4150	0.2500	0.4500	0.4600	0.4400
P@30	0.3400	0.3567	0.4167	0.3800	0.2733	0.4333	0.4333	0.4167
MAP	0.0614	0.0865	0.1018	0.0800	0.0453	0.1074	0.1056	0.1063

2.3 Parameters

For K1, I first choose 0 to see the influence of absence of k1. Then I test the result when k1 growth to a large number. Then I test k1 with some numbers that is close to the default value which is 1.2. and I found 1.2 is not the best k1 value for this query and corpus.

For b, I firstly test 0, 0.25, 0.5, 0.75, 1, to see the overall trend of the p@n and MAP with the increase of b. then I found 0.25 has the best result, and I use the rest test to find out in the range of the best value of b for this query and corpus.

2.4 Discussion

For BM25, when k1 is 0, the term frequency part of the algorithm will become to tf / tf which is 1. So the term frequency will not have effect on the result, and since we does not consider repeated query term so uncommon words will have larger score since their document frequency is small. As I increase the K1 from 0.5 to 1 to 10 to 40, the MAP and p@n dropped. Which means the best value of k1 should less than 1. The value of MAP dropped significantly when I increase the k1 from 1 to 10. When we have a really large K1 the difference between length of documents will play a larger role, and the impact of tf will become less significant. And after I compare the k1 default value with 0.5, 0.8, 1, 1.4. I found 1.2 is not the best k1 value for this corpus and queries. The MAP and p@n drop consistently as we increase the k1 value from 0.5 to larger values. So the best value is smaller or equals to 0.5. the relevant document retrieved number for k = 0.5 is 285, for k = 0.8 is 276, for k = 1 is 272, which also proves my hypothesis.

For value of b, firstly I test it against 0, when b is equals to 0. The document length normalization function will be 1, and since longer document will have a larger chance to match the query terms, so long documents will be retrieved. Then I found when we increase b from 0.25 to 0.5 to 0.75 to 1, to p@n and MAP value keep dropping. And when b is equals to 1, the retrieved set has worst results. So I think in our corpus most long documents are relevant document of our query terms. In addition, during the previous experiments, I found when b is equals to 0.25, the retrieve set has the best results, so I test some other numbers that is close to 0.25, which are 0.1, 0.15, 0.2. Comparing those results with the result of 0.25, I found the MAP and p@n and the number of relevant document retrieved keep increase from 0.1 to 0.15 and drop from 0.15 to 0.25, so the best value for b shoud close to 0.15, between 0.1 and 0.2. The retrieved number for b = 0.15 is 337, b = 0.1 is 321, b = 0.2 is 336.

3 Experiment 3: Indri Parameter Adjustment

3.1 μ

	μ							
	2500	0	500	1000	2000	5000	6000	10000
P@10	0.4900	0.2700	0.4500	0.4400	0.4600	0.4600	0.4700	0.3900
P@20	0.4250	0.2450	0.3850	0.4100	0.4300	0.4300	0.4100	0.4050
P@30	0.3833	0.2600	0.3500	0.3767	0.3933	0.4000	0.4000	0.3800
MAP	0.0973	0.0492	0.0690	0.0796	0.0936	0.0948	0.0941	0.0871

3.2 λ

	λ							
	0.4	0	0.1	0.8	1	0.2	0.15	0.6
P@10	0.4900	0.4900	0.4800	0.3700	0.0000	0.4800	0.4800	0.4700
P@20	0.4250	0.4200	0.4100	0.3350	0.0000	0.4250	0.4150	0.3950
P@30	0.3833	0.4100	0.4067	0.3333	0.0033	0.3967	0.4000	0.3733
MAP	0.0973	0.0994	0.0997	0.0765	0.0002	0.0990	0.0992	0.0905

3.3 Parameters

For Mu, I choose 0 first, then I want to see the p@n and MAP's react when mu goes up, so I choose 1000, 2000, 5000 and 10000. And I found the MAP and p@n does not change much when mu is larger than 2000, so I choose 6000. To find further relationship.

For Lambda, Firstly I choose 0 and 1 just to try some boundary value, then I want find the growth tendency of MAP and p@n with respect of lambda's increase. So I choose 0.1, 0.6, 0.8. then I found the MAP and p@n is largest when lambda is around 0,1, so I choose 0.2 and 0.15 to find the range.

3.4 Discussion

For Mu, when it is equals to 0, we will not consider the appearance of terms regards to the whole collection of document, so under this circumstance, if some document miss some terms, it will be less likely to be retrieved, even it is a relevant document. And if some word is rare in this document but it is common in the whole collection, its score will be smaller. In addition, as the Mu increase, the MAP and p@n increase at first, and after Mu reach 2000, the MAP and p@n is stable from 2000 to 6000. So I think Mu only influence the result slightly, unless it is extremely large or equals to 0. If Mu is extremely large, documents with common terms will be likely to be retrieved.

For Lambda, when it is equals to 0, the algorithm will be the same of unsmooth algorithm, when we have terms that some are frequent terms and others are rare terms, they will have same weight, which may result in document with common term are more likely to be retrieved. When the lambda is equals to 1, the algorithm will only consider the ctf / length of collection. So documents with common terms will be retrieved and both of them will have same score. Except the boundary, the MAP and p@n keep dropping with Lambda increase from 0.1 to 0.8. it is interesting that the MAP and p@n are stable when lambda is in the range of [0, 0.2] the best value of lambda should around 0.1, and when lambda is 0, the MAP and p@n shows that the retrieve set is really good, which means terms in our queries are almost same common.

4 Experiment 4: Different representations

4.1 Example Query

#AND (#WSUM (0.2 indiana.url 0.3 indiana.title 0.4 indiana.keywords 0.1 indiana.body) #WSUM (0.2 child.url 0.3 child.title 0.4 child.keywords 0.1 child.body) #WSUM (0.2 support.url 0.3 support.title 0.4 support.keywords 0.1 support.body))

4.2 Results for the Query Set

	Indri BOW (body)	0.2 url 0.4 keywords 0.3 title 0.1 body	0.7 url 0.1 keywords 0.1 title 0.1 body	0.1 url 0.1 keywords 0.7 title 0.1 body	0.1 url 0.7 keywords 0.1 title 0.1 body	0.1 url 0.1 keywords 0.1 title 0.7 body
P@10	0.4900	0.3400	0.4100	0.3300	0.3400	0.4600
P@20	0.4250	0.3450	0.3500	0.3500	0.3300	0.4250
P@30	0.3833	0.3233	0.3200	0.3267	0.3133	0.3867
MAP	0.0973	0.0675	0.0690	0.0712	0.0661	0.0954

4.3 Weights

First I think keyword is the most important field so the first experiment I give it largest weight, but the outcome is not good. So I want to find out which field is most important when we retrieve documents, I test it with 0.7 to the test field and all other fields are 0.1.

4.4 Discussion

From my experiments, the most important fields is body, when I give it the largest weight, the MAP and p@n is the highest among all other experiments. I guess this is because some terms does not show up on url or title, and if we give those field large weights it will make us miss the relevant documents. However, body field decides whether this document is relevant to the query. In addition, the title field is secondary important field. When I parse the query terms, I does not delete terms for title, for instance, the term sea is a relative common term, so it probably will not show up on the title of a relevant document. If we further parse the query, the result probably will be better.

5 Experiment 5: Sequential dependency models

5.1 Example Query

#wand(0.5 #and(indiana child support) 0.25 #and(#near/1(child support) #near/1(indiana child)) 0.25 #and(#window/8(child support) #window/8(indiana child)))

5.2 Results for the Query Set

	Indri BOW (body)	0.5 AND 0.25 NEAR 0.25 WINDOW	0.1 AND 0.8 NEAR 0.1 WINDOW	0.8 AND 0.1 NEAR 0.1 WINDOW	0.1 AND 0.1 NEAR 0.8 WINDOW	0.45 AND 0.1 NEAR 0.45 WINDOW
P@10	0.4900	0.5000	0.3900	0.4800	0.4800	0.5100
P@20	0.4250	0.4200	0.3700	0.4200	0.4350	0.4300
P@30	0.3833	0.4100	0.3200	0.4100	0.3933	0.3933
MAP	0.0973	0.0930	0.0742	0.0941	0.0865	0.0917

5.3 Weights

Firstly, I want to find out which operator contributes most to the MAP and p@n, by giving a operator higher weight (0.8) and the other two operator 0.1. After I find out AND is the most important operator I run one experiment that gives AND 0.5 and the other two 0.25, and since the WINDOW operator is the secondary important operator, I ran another experiment that give AND and WINDOW 0.45 respectively, and give NEAR 0.1.

5.4 Discussion

Comparing to the default BOW trial, all other experiments retrieves more relevant documents, the default trial retrieved 306 relevant documents. except the third experiment that NEAR = 0.8 and AND, WINDOW = 0.1 which retrieved 277 documents. Which shows that with proper structured query, the Indri model can achieve a better performance. Among three operator, the NEAR operator has the worst result, I think this is because that NEAR operator needs term's order and distance at same time, which will filter out more documents, and some of them probably is relevant document. However, the WINDOW operator only needs terms existing between a given range, so terms are more likely to satisfy this operator and more relevant documents will be retrieved. As for AND operator the requirement is even more easy to fulfill, the document will be retrieved if given terms are in the same documents. This result in that SDM queries that has a higher weighted AND operator are more likely to retrieve more relevant documents.