Your Name: Raghav Sharma

Your Andrew ID: rvsharma

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.

   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: RAGHAV SHARMA

Your Andrew ID: rvsharma

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	Value 1 = 0.1	Value 2 = 0.3	Value 3 = 0.5	Value 4 = 0.7	Value 5 = 0.9	Value 6 = 1.3	Value 7 = 1.5
P@10	0.3700	0.4500	0.4500	0.4500	0.4000	0.3900	0.3700	0.3700
P@20	0.3550	0.4350	0.4400	0.4100	0.3800	0.3700	0.3550	0.3500
P@30	0.3400	0.4100	0.3967	0.3700	0.3533	0.3367	0.3267	0.3267
MAP	0.0614	0.0768	0.0718	0.0676	0.0650	0.0632	0.0606	0.0598

2.2 b

	b							
	0.75	Value 1 = 0.1	Value 2 = 0.2	Value 3 = 0.3	Value 4 = 0.4	Value 5 = 0.5	Value 6 = 0.6	Value 7 = 0.9
P@10	0.3700	0.4800	0.4900	0.5000	0.5000	0.4800	0.4300	0.2700
P@20	0.3550	0.4400	0.4600	0.4500	0.4350	0.4150	0.4100	0.2900
P@30	0.3400	0.4167	0.4333	0.4233	0.4000	0.3800	0.3533	0.3000
MAP	0.0614	0.1063	0.1056	0.0945	0.0858	0.0800	0.0725	0.0520

2.3 Parameters

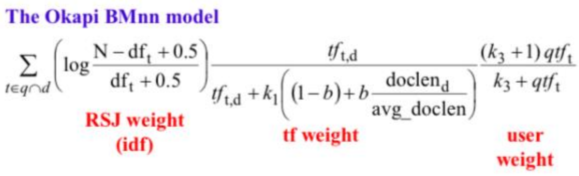
Parameter k1: The optimal range of k is usually kept at 1.2. in the experiment we have varied the k1 from 0 – 1.5. I have taken a mixture of the values for k1. In increments of 0.2. to see the effect on MAP and P@n, the value of k1 approches zero(at k = 0.1, value 1), and when the value of k1 approches the higher limit of 2.0 (at k1 = 1.5, value 7). The values 6 and 7, we can see the saturation of the P@n happening.

Value 1 we have the highest MAP, while value 2 and 3 are having highest P@10. Value 5 and 6 are around the default value of k1. And so are chosen to demonstrate the continuance of the term frequency and probability curve.

Parameter b: The default value is taken to be 0.75 for b. Since the range is from 0 – 1.0, I have taken values of b in increment of 0.1 starting from 0.1 to 0.9. Values 0.7 and 0.8 are left out because they are near to the default value of 0.75 which is already demonstrated for baseline. Value 7 is chosen because it shows some interesting points where the P@n rises as the value of n increases, something that doesn't happen in smaller values of b. At lower values of b from Values 1, 2 and 3 we have the highest MAP and P@10.

The changes are done in increments of 0.1(for b) and 0.2(for k1) to see the influence that changing the values has on the precision of the results. With this type of fine tuning, the precision can be maximized for a collection by observing the P@n and MAP values.

2.4 Discussion



The Okapi BMnn model

$$\sum_{t \in q \cap d} \left(\log \frac{\mathrm{N} - \mathrm{df}_t + 0.5}{\mathrm{df}_t + 0.5} \right) \frac{tf_{t,d}}{tf_{t,d} + k_1\left((1-b) + b\frac{\mathrm{doclen}_d}{\mathrm{avg\_doclen}}\right)} \frac{(k_3+1)\,qtf_t}{k_3 + qtf_t}$$

RSJ weight (idf)
tf weight
user weight

The score calculation for a document in BM25

[Taken from Slide 30 from Jamie's lecture (https://boston.lti.cs.cmu.edu/classes/11-642/Lectures/07-BestMatch-I.pdf)]

Parameter k1: This parameter deals with the saturation in term frequency. Higher the value of k1, it will take a higher frequency for the term to saturate. Lower the value of k1, the quicker the term will saturate (at a lower frequency). The optimal value of the parameters depends on the collection.

If the system designer knows the nature of the corpus (longer versus shorter documents, diversity between the documents), it helps to set the value of k1 and b for BM25. For longer documents (or diverse) its highly probable that the query will match, since it gives more chance for the terms to appear in the collection than the shorter documents. For example, a term like 'apple' can easily appear 10 to 100 times in a document and the document might have nothing to do with it. Therefore, setting the value of k1 makes the saturation point to be at a high value.

As we can see from the general trend in the experiment table for k1, the lower we set the value of k1, higher the precision. MAP has its highest value at k1 = 0.1 and P@10 have the highest values at 0.1,0.2, 0.3. From this, we can infer the nature of the collection. It most likely contains shorter length documents which are not very diverse in nature. This can also be inferred from the results from the parameter b variation as mentioned below.

Parameter b: This parameter shows the effect of field length normalization. If b → 0, the document length is ignored, however at full limit of b = 1.0, the field length is fully normalized.

Again, the MAP and P@n values are highest when b→ 0. From the formula given above we can infer that when b approaches 0, term-frequency weight has no impact on the score, and longer documents are more likely to be fetched than shorter documents. When the b → 1, the term-frequency plays an important role in the determination of the document score. For specific genre documents (eg medical, legal) which are lengthy it would be helpful to keep b lower to 0 as the documents that match the query will be more relevant to the information need. If the corpus is more generic, it is beneficial that we keep b closer to 1 and term-frequency will be important when calculating the document score. Thus for a generic collection of documents (news articles, encyclopedia pages, etc) it helps to keep b at a value closer to 1.

As we observe that the MAP and P@n are the highest closer to 1, this implies that the document collection is more specific in nature and the term frequency does not play an important role in the results.

An interesting thing to notice is that as the value of b approaches 1(b = 0.9) the precision increases with the value of n. that is P@30 > P@20 > P@10. So for this collection as the document length becomes more important we see more precise results for a larger result set.

3 Experiment 3: Indri Parameter Adjustment

3.1 μ

	μ							
	2500	Value 1 = 0	Value 2 = 1500	Value 3 = 2000	Value 4 = 3500	Value 5 = 5500	Value 6 = 7500	Value 7 = 10000
P@10	0.4900	0.2700	0.4600	0.4600	0.4700	0.4700	0.4500	0.3900
P@20	0.4250	0.2450	0.4100	0.4300	0.4250	0.4250	0.4150	0.405
P@30	0.3833	0.2600	0.4000	0.3933	0.3967	0.3967	0.3967	0.3800
MAP	0.0973	0.0492	0.0885	0.0936	0.0965	0.0945	0.0912	0.0871

3.2 λ

	λ							
	0.4	Value 1 = 0.0	Value 2 = 0.1	Value 3 = 0.2	Value 4 = 0.3	Value 5 = 0.6	Value 6 = 0.8	Value 7 = 1.0
P@10	0.4900	0.4900	0.4800	0.4800	0.4800	0.4700	0.3700	0.0000
P@20	0.4250	0.4200	0.4100	0.4250	0.4250	0.3950	0.3350	0.0000
P@30	0.3833	0.4100	0.4067	0.3967	0.3967	0.3733	0.3333	0.0033
MAP	0.0973	0.0994	0.0997	0.0990	0.0982	0.0905	0.0765	0.0002

3.3 Parameters

Parameter mu: The optimal value of the mu is between 1000 – 10000. And based on the question 2 the documents length is comparatively shorter. So, we would benefit from keeping the value of mu higher (approaching 10000). Most of the values selected in my experiment are > 2500 (optimal value). To demonstrate the smoothing effect of parameter mu I have incremented the value of mu by ~1500 for each sample. The value of mu = 0, is selected to discuss the effect of mu = 0, on the bag-of-words queries.

Parameter lambda: The bag of words average query length is ~3 words per query. The range of lambda is from [0.0, 1.0]. Thus, every term in the query is as important as every other term. As lambda approaches zero(Value 1), smoothing of the query sample decreases and there is no idf like effect present in the query. The idf effect corresponds to the second part of the equation $\lambda \frac{ctf(qi)}{length(c)}$. When the lambda approaches 1(value 7) we only have the idf like effect and the bayesian smoothing and mu have no effect.

$$
\begin{aligned}
p(q\,|\,d) &= \prod_{q_i \in q} p(q_i\,|\,d) \\
&= \prod_{q_i \in q} \left((1-\lambda)\frac{tf_{q_i,d} + \mu\, p_{MLE}(q_i\,|\,C)}{length(d)+\mu} + \lambda\, p_{MLE}(q_i\,|\,C) \right) \\
&= \prod_{q_i \in q} \left((1-\lambda)\frac{tf_{q_i,d} + \mu \frac{ctf(q_i)}{length(c)}}{length(d)+\mu} + \lambda \frac{ctf(q_i)}{length(c)} \right)
\end{aligned}
$$

Calculating the score in Indri model

[Taken from slide 26 of Jamie's lecture: https://boston.lti.cs.cmu.edu/classes/11-642/Lectures/08-BestMatch-II.pdf]

3.4 Discussion

Parameter mu: As the document collection is comprised of shorter documents, the larger value of mu is more important as probabilities are more granular. We can see for value 1 when the value of mu approaches zero, P@10 is 0.2700 and at P@20, the value of precision decreases and then increases at P@30. This suggests that there is no smoothing present at mu = 0 and the model behaves like a maximum likelihood model.

The value of MAP is highest at the optimal value of mu = 2500 i.e MAP = 0.0973. as the value of mu increases the MAP value remains almost stable at ~0.09xx. this again suggests that smoothing happens for larger mu values as probabilities become more granular.

Parameter lambda: The parameter lambda is related to the query length and relates how important is idf weighting for the collection. The bag of words query contains on an average of 3 terms which is considered on the lower side of the spectrum. Therefore we would benefit from a lower value of lambda.

The results have been summarized in the tables. As we would predict that a lower lambda is more beneficial for shorter queries as it removes the smoothing and the idf effect of the query term on the corpus. When the value of lambda is 0, there is no smoothing and no idf effect. This is inferred by the high value of MAP and P@n when lambda = 0.0. The value of MAP and P@n remains stable for lower values of lambda(<0.5) and MAP remains in 0.9 and P@10 at 0.4800 – 0.4900. However as the value of lambda is increased from 0.5 to 1.0 we can observe the value of MAP and P@n falling rapidly. And finally P@n and MAP approaches precision of the Bag-of-words #OR unranked retrieval model when lamda = 1.0.

4 Experiment 4: Different representations

4.1 Example Query

```
#AND (
        #WSUM ( 0.1 indiana.url   0.2 indiana.keyword      0.2 indiana.title   0.5 indiana.body    )
        #WSUM ( 0.1 child.url     0.2 child.keyword        0.2 child.title     0.5 child.body      )
        #WSUM ( 0.1 support.url   0.2 support.keyword      0.2 support.title   0.5 support.body    ))
```

4.2 Results for the Query Set

	Indri BOW (body)	**0.10 url** **0.20 keywords** **0.20 title** **0.50 *body* (A)**	**0.10 url** **0.20 keywords** **0.30 *title*** **0.40 *body* (B)**	**0.10 url** **0.70 *keywords*** **0.10 title** **0.10 body (C)**	**0.70 *url*** **0.10 keywords** **0.10 title** **0.10 body (D)**	**0.10 url** **0.10 keywords** **0.10 title** **0.70 body (E)**
P@10	0.4900	0.4600	0.4700	0.3400	0.4100	0.4600
P@20	0.4250	0.4050	0.4000	0.3300	0.3500	0.4250
P@30	0.3833	0.3733	0.3667	0.3133	0.3200	0.3867
MAP	0.0973	0.0913	0.0870	0.0661	0.0690	0.0954

4.3 Weights

As the Indri model combines evidence from the different fields of the document, it is beneficial to make each field a priority (with a higher designated weight) and see the impact on the precision parameters.

In Value column 1(A) and column 5(E), the weight for body field is kept high.

For column 2(B), the weight for title field is increased to study the impact of the query terms matching in title field of the documents. It would be interesting to see the impact of the queries matching the title field.

For column values 3(c) and 4(d), the fields of keywords and url, respectively are given weightage.

4.4 Discussion

Field- Body: As we can observe from columns A and E, increasing or giving weightage to a body field in the query has a high correlation to the MAP and P@n values when compared to BOW query structure. The maximum values of MAP and P@n occurs when the body field is given more weightage compared to any other field. And it approaches to the BOW structure. This may be because more the relevant terms occur in the body of the document more likely is that the document is relevant to the user and justified that it should rank higher. I had increased the weight of the body field from .5 to 0.7 from column A to column E and we can see that both P@n and MAP increases and approaches to the BOW.

Field- Title: if we give more importance to the title field as done in column B we observe that the P@10 increases significantly than for any other field (even body). This might be because, the title field of the document highlights the important topic in the document and mentions what the document is about, thus it is likely that if the query term matches the title field and we place that in the higher rank than the others, it will be more relevant to the user and likely to satisfy the user information need better.Thus the P@10 value is higher for the title field column B. However, as we fetch more and more document thequery terms are less likely to appear in the title field of the document and unlikelier that they will be fetched to the result. Thus, the value of P@20 and P@30 is lower than most other fields and the MAP is also lower than most other fields.

Field- keywords: In column C, the keywords field is given weight. It has the lowest precision score and MAP among all the fields. This may be because the keywords are different from what the user wants to search and may not be important for that document. Thus, the keywords for that body of the text may be completely different from the information need and thus, the system has a difficult time matching the relevant documents to the user query. This results in low precision of the results and documents being fetched.

Fields- url: In column D, the weightage given for the URl field is 0.7. As we notice that the outcome is a low P@n and MAP value but not as low as for the keywords field. The URL of a page might contain important information about what the page is about. It has similar properties as does a title field for that document. If a query matches with the URL field of the document then it is more likely to be a relevant document. Similar to the title field we observe that P@10 value for the URl is higher when compared to the P@20 and P@30 value, this again amy be because the URL might highlight what is important in the document and thus may give high precision and higher document ranks for first few documents. But as more documents are retrieved they are not as relevant for the user, and consecutively less and less documents match.

5 Experiment 5: Sequential dependency models

5.1 Example Query

```
#WAND (
        0.7 #and ( indiana child support )
        0.2 #and ( #near/1(indiana child) #near/1(child support) )
        0.1 #and ( #window/8(indiana child) #window/8(indiana child)) )
```

5.2 Results for the Query Set

	Indri BOW (body)	**0.7 AND 0.2 NEAR 0.1 WINDOW (A)**	**0.80 AND 0.10 NEAR 0.10 WINDOW(B)**	**0.50 AND 0.30 NEAR 0.20 WINDOW(C)**	**0.10 AND 0.70 NEAR 0.20 WINDOW(D)**	**0.10 AND 0.20 NEAR 0.70 WINDOW(E)**
P@10	0.4900	0.4800	0.4700	0.4900	0.4800	0.4200
P@20	0.4250	0.4350	0.4150	0.4350	0.3750	0.3950
P@30	0.3833	0.4067	0.4067	0.4033	0.3533	0.3600

MAP	0.0973	0.0884	0.0922	0.0900	0.0790	0.0720

5.3 Weights

In sequential dependency model, #AND operator represents the strict need that all the arguments to that operator must match and it is a strict checking. The #NEAR and #WINDOW parameter relaxes some of this requirement and searches for the terms in the /n proximity of each other.

In columns A, B and C, we have gradually reduced the importance of the AND operator from absolute weight in B to least weight in C and observed the result for P@n and MAP.

In columns D and E, NEAR and WINDOW operators, respectively are given highest weightage.

5.4 Discussion

#AND operators (column A, B, C) – From columns B we can see that the highest weightage given to the AND operator results in the maximum MAP value. The higher value of AND places a restriction on the system to fetch the most relevant document overall, but the value of P@n is lower than the columns A and C. The P@10 value for A and C column increases as we relax the requirement of strict sequence matching in these columns. We relax the requirement by giving more weightage to the proximity operators.

As we relax these requirements, we see a higher match taking place in the top 10 and 20 documents that the strict Boolean AND operators.

#NEAR operator (column D) – When the #NEAR inverted list operator is given more weightage we tend to get accurate results for the first 10 documents. These results are similar to the column A particularly for P@10 values. Thus the #NEAR operator has a similar precision as #AND operator and a lower recall as demonstrated by lower P@20 and P@30 values. This may be because the #NEAR operator looks for the terms in the proximity of each other and is able to find more matches than if it were a strict AND. Relaxing this requirements results in increase in precision for first 10 results, but as more results are fetched it is harder for the NEAR operator to provide a match and therefore there is lower precision for P@20 and P@30 values. Also, the sequence of the terms may be different than the one put in query and this might also be the cause of a lower precision value. For example, “time traveler’s wife” has a different sequence than “wife of time traveler” and we don’t check for the latter case, but it is still a probable occurrence in the documents.

#WINDOW operator (column E) – The window operator relaxes the sequential requirement of the window operator in that any term can occur in any sequence and it will look for all the possible sequences. However, it results in higher running time than AND and NEAR operator since it takes longer to travel down the inverted list. Since the sequential argument is relaxed for the WINDOW operator it has a higher recall than the NEAR operator as evidenced by the higher P@20 and P@30 values.

Although, the P@10 value for WINDOW is lower than the NEAR operator. Due to the possibility of occurrence of query terms in any order, and a larger proximity distance operator (4 times the number of query terms) the precision reduces for the top matches and thus P@10(WINDOW) < P@10(NEAR).