

**Your Name : Aditya Gautam**

**Your Andrew ID : agautam1@andrew.cmu.edu**

## **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.

**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 : Aditya Gautam**

**Your Andrew ID : agautam1@andrew.cmu.edu**

## **Homework 3**

### **1 Experiment 1: Baselines**

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

Document the parameter settings that were used to obtain these results.

#### **Answer 1)**

The parameters used for BM25 and Indri model are the default parameters as mentioned below.

BM25 parameters :  $k_1 = 1.2$ ,  $b = 0.75$ ,  $k_3 = 0$ .

Indri parameters :  $Mau = 2500$ ,  $\Lambda = 0.4$ .

## 2 Experiment 2: Different representations

Describe your strategy for setting the weights on the different representations. Describe how you expected the different weight combinations to perform (before the experiment), and why.

Discuss the trends and stability that you observed in Experiment 2; whether the different weight combinations behaved as you expected; how Precision and Recall tended to vary as the weights were varied; how the differences in accuracy (if any) relate to different computational cost; and your other observations, interpretations, or conclusions from running this experiment.

### Answer 2)

**Weights** on each of the field of the document determines the contribution of that field in the final ranking of the documents. For this experiments, I have tried to get the importance of various field with main emphasis on body and title, since I think they are very relevant in most context. However, in some single word query, inlink and url might play a significant role so in other query I made body weight to zero and kept equal weights on other fields.. In other query, I kept the weight of body around 80% and others were kept around 10% i.e. url and inlink. In most of the query done on the net, we usually finds the term in the title and the body, so to check the impact/relevance of this, in one of the query, I kept only these two fields with equal weightage of 50%. So, one of the five query contains only title and body. It may be possible that the words might be present in the keywords, inlink, url and title and ofcourse body, to fetch these type of pages, we need to give proper weightage to all the documents with little more emphasis on the body. So, in another query, I kept the weightage of body as 40% and other fields are kept at 10-20% just to make sure that to consider terms present in these fields. So, these combinations of queries and weights are chosen carefully to analyze the impact of various fields in the ranking.

As per my **expectation**, for the query, which have zero weightage to the body is expected to have the lowest MAP and P@ value, which is what has happened in all the queries mentioned below. For the query in which equal weightage is given to both body and title, it is expected to perform good but not better than 100% body weightage since the term frequency in the body is ideally higher than in the title. So, the total term frequency thus the score of doc will go down, likewise lower MAP and P@n. This is seen in the experiments performed.

One of the **trends** to observe is that body is the most important fields of the document as increasing or decreasing the weightage impact the performance (MAP and P@) in the same manner. I was expecting the URL and inlink should have good impact on the MAP but apparently their weightage doesn't matter a lot and very high importance is given to the body field. Apart from this, for single words query inlink places a significant role, as I saw a huge increase in MAP only on increasing the parameters.

It has been **observed** that MAP/recall varies greatly with the change in the weights as seen by the below tables. Giving low weights to the body field would lower the MAP drastically but precision at 10,20 and 30 doesn't vary a rate of MAP. Precision@30 remains more or less same until the weight of the body is drastically brought down. Variation in the MAP with the weight of body field is expected since many relevant documents would have term present in the body and keeping the body field weight to low cause relevant documents to go down in the rank thereby reducing MAP.

Also, not a very huge difference in the **computation cost** has been observed on running queries with different weightage however running these queries are expensive than the baseline query(bag of words) since it performing several operations and combining the results. **This additional computation cost is not worth it since there is not much difference in the MAP and P@N values.** The time taken to run a set full query is more or less same. **Looking at the results, we can conclude that body field is the most**

significant field out of all the fields in the documents and inlink field is quite relevant to some single words queries.

**Query set :**

69:sewing instructions  
79:voyager  
84:continental plates  
89:ocd  
108:ralph owen brewster  
141:va dmrv registration  
146:sherwood regional library  
153:pocono  
171:ron howard  
197:idaho state flower

**Table : Different representation**

	<b>Indri BOW (body)</b>	<b>0.1 url 0.0 keywords 0.3 title 0.6 body 0.0 inlink</b>	<b>0.1 url 0.1 keywords 0.2 title 0.4 body 0.2 inlink</b>	<b>0.00 url 0.00 keywords 0.5 title 0.5 body 0.00 inlink</b>	<b>0.2 url 0.3 keywords 0.3 title 0.0 body 0.2 inlink</b>	<b>0.1 url 0.00 keywords 0.00 title 0.8 body 0.1 inlink</b>
<b>P@10</b>	0.4000	0.3300	0.3000	0.3444	0.4444	0.3500
<b>P@20</b>	0.4700	0.4450	0.3550	0.4611	0.3842	0.4356
<b>P@30</b>	0.4233	0.4533	0.3900	0.4519	0.2479	0.4567
<b>MAP</b>	0.2075	0.2099	0.1881	0.2143	0.1397	0.2072

### 3 Experiment 3: Sequential dependency models

Describe your strategy for setting the weights for the different components of the sequential dependency model. Describe how you expected the different weight combinations to perform (before the experiment), and why.

Discuss the trends and stability that you observed in Experiment 3; how Precision and Recall tended to vary as the weights were varied; whether the more complex query behaved as you expected; whether the improvement in accuracy (if any) is worth the increased computational cost; and your other observations, interpretations, or conclusions from running this experiment.

Ans 3)

	<b>Indri BOW (body)</b>	<b>0.80 AND 0.10 NEAR 0.10 WINDOW</b>	<b>0.10 AND 0.80 NEAR 0.10 WINDOW</b>	<b>0.10 AND 0.10 NEAR 0.80 WINDOW</b>	<b>0.40 AND 0.30 NEAR 0.30 WINDOW</b>	<b>0.20 AND 0.40 NEAR 0.40 WINDOW</b>
<b>P@10</b>	0.4000	0.4500	0.2751	0.2875	0.4900	0.4778
<b>P@20</b>	0.4700	0.5200	0.4667	0.4889	0.5150	0.4889
<b>P@30</b>	0.4233	0.5100	0.4889	0.4938	0.4967	0.4815
<b>MAP</b>	0.2075	0.2718	0.2751	0.2874	0.2835	0.2805

Weights on each operator signifies the contribution of each operator score in the final score/ranking of the documents. In 3 queries, I have set the weight of AND, NEAR and WINDOW operator as very high keeping the other two very low. The reason for doing so is to check the performance i.e. MAP and P@N with respect to that operator. These weights distribution alone will give us a glimpse of the significance of operator in getting the better results among three operator for the given set of query. In another query, I have set the weight of NEAR and WINDOW operator equal and more or less same as that of AND operator, so in this combination, the contribution of each operator would be same. This is to check the impact of retrieving the documents when all types of operators are needed to get the best doc. In this last combination, I kept the weightage of AND low as compared to others operator.

My expectation of different combination was that high weightage of NEAR operator and some weightage of AND operator in some query would give the most relevant result. To my surprise, MAP value was highest for the combination in which WINDOW was given the highest weightage. With window, terms can present in any order, so I expected the lower MAP as compared to NEAR.

One of the main trend to observe is that using NEAR and WINDOW operator significantly improve the overall MAP and P@N value, as shown in the table above. The MAP value increased directly by 40-45% on using any of these operator, even when the weightage given to the NEAR or WINDOW operator was low. However, changing the weightage among the NEAR, WINDOW and AND operator doesn't cause a very huge improvement in the MAP value, but they are very high as compared to the baseline.

Coming to the precision, there is not a very significant change in the precision on adding or varying the weights of different operators since cumulative term frequency of the top documents wouldn't vary a lot but there is significant improvement in the MAP as all the documents with high relevance would be fetch at a higher rank with the use of NEAR and WINDOW operator.

Behavior on addition of NEAR and WINDOW operator is as expected i.e. higher MAP value but getting higher MAP value on increasing the weightage of WINDOW operator and decreasing the weightage of NEAR was little surprising since I believe the WINDOW operator sometime may fetch irrelevant documents since term could be present in any order and the paragraph wouldn't have the same meaning as intended by the query.

Also, not much difference in the **computation cost** has been observed on running queries with different weightage however running these queries are expensive than the baseline query(bag of words) since it performing several operations and combine the results later. **This additional computation cost is worth it since there is huge difference in the MAP and P@N values.**

So, to conclude, we can say that sequential dependency model with proper weightage given to different operators significantly improves the performance of the results and helps in retrieving more relevant results as compared to baseline(bag of word) approach.