**Your Name: Jiachen Li**

**Your Andrew ID: jiachenl**

**Homework 3**

# Statement of Assurance

I certify that all of the materials I submit are original works that were done by myself.

# Experiment 1: Baselines

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Ranked**  **Boolean** | **BM25**  **BOW** | **Indri**  **BOW** |
| **P@10** | 0.1500 | 0.2900 | 0.2400 |
| **P@20** | 0.1800 | 0.3050 | 0.2750 |
| **P@30** | 0.1667 | 0.3267 | 0.2967 |
| **MAP** | 0.0566 | 0.1325 | 0.1275 |
| **Time** | 00:09 | 00:09 | 00:09 |

# Experiment 2: Different representations

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Indri**  **BOW**  **(body)** | **0.20 url**  **0.20 keywords**  **0.20 title**  **0.20 body**  **0.20 inlink** | **0.05 url**  **0.10 keywords**  **0.20 title**  **0.60 body**  **0.05 inlink** | **0.05 url**  **0.20 keywords**  **0.10 title**  **0.60 body**  **0.05 inlink** | **0.05 url**  **0.10 keywords**  **0.05 title**  **0.60 body**  **0.20 inlink** | **0.20 url**  **0.05 keywords**  **0.10 title**  **0.60 body**  **0.05 inlink** | **0.05 url**  **0.05 keywords**  **0.05 title**  **0.80 body**  **0.05 inlink** |
| **P@10** | 0.2400 | 0. 1100 | 0.2100 | 0.2100 | 0.1800 | 0.2200 | 0.2300 |
| **P@20** | 0.2750 | 0.1600 | 0.2650 | 0.2600 | 0.1950 | 0.2650 | 0.2700 |
| **P@30** | 0.2967 | 0.1633 | 0.2533 | 0.2500 | 0.1933 | 0.2533 | 0.2633 |
| **MAP** | 0.1275 | 0.0853 | 0.1103 | 0.1113 | 0.0934 | 0.1125 | 0.1159 |
| **Time** | 00:09 | 00:09 | 00:09 | 00:09 | 00:09 | 00:09 | 00:09 |

**My strategy for setting weights:**

Given that different representations may not have the same importance, then my strategy is to assign the weights based on how important (in my opinion) the field is. To be specific, since body field does play an important role, I may always assign a higher weight for it, e.g. greater or equal than 0.6. Besides, since I’m not quite sure about the influence of each other field (e.g. url file may be important for short words, but may not be that useful for very long term), so I tried different combinations (i.e., in some queries, title has more weights while in another keywords field has more weights, etc). At last, I also made a query with the same weights for each field, which can work as another baseline to compare the importance among each field.

Discuss any trends that you observe; whether the different representations behaved as you expected; the Precision and Recall characteristics of each representation; how the differences in accuracy (if any) relate to different computational cost; and any other observations that you may have.

**My observations:**

According to my results, the results tend to be better as more weight is assigned to the body field and the overall performance of multiple representation cannot beat the baseline BOW on the body field. This result is actually not what I expect (I thought that multiple representation can be somewhat helpful, even a little. Maybe this is due to my weight setting), but it also not surprising since the body field really contains much information. Nevertheless, compared with the query with equal weights for each field, the multiple representation can be helpful when the appropriate weights are specified. Moreover, when I apply the different combination of weights for fields other than body, the results varies and the best one (with highest P@n and MAP) is achieved when higher weights are assigned to url and title fields. In addition, it can be seen from the time that multiple representation may not have much additional computational cost. So if you find that the multiple representations are useful, then don’t need to worry about the cost.

BTW, I also notice that the equal weights query has the worst result, which verifies that different fields don’t have the same contribution for the result.

# Experiment 3: Sequential dependency models

**Example Query:** Provide your structured query for query “fickle creek farm”.

#WAND(

0.3 #AND(fickle creek farm)

0.5 #AND(#NEAR/1(fickle creek) #NEAR/1(creek farm))

0.2 #AND(#WINDOW/8(fickle creek) #WINDOW/8(creek farm)) )

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Indri**  **BOW**  **(body)** | **0.60 AND**  **0.30 NEAR**  **0.10 WINDOW** | **0.50 AND**  **0.30 NEAR**  **0.20 WINDOW** | **0.50 AND**  **0.40 NEAR**  **0.10 WINDOW** | **0.45 AND**  **0.45 NEAR**  **0.10 WINDOW** | **0.30 AND**  **0.50 NEAR**  **0.20 WINDOW** |
| **P@10** | 0.2400 | 0.3500 | 0.3700 | 0.3700 | 0.3700 | 0.3700 |
| **P@20** | 0.2750 | 0.3600 | 0.3600 | 0.3700 | 0.3700 | 0.3750 |
| **P@30** | 0.2967 | 0.3500 | 0.3633 | 0.3567 | 0.3567 | 0.3667 |
| **MAP** | 0.1275 | 0.1807 | 0.1842 | 0.1852 | 0.1873 | 0.1909 |
| **Time** | 00:09 | 00:19 | 00:19 | 00:19 | 00:19 | 00:19 |

**My strategy for setting weights:**

In sequential dependency model, I think the key component is the #NEAR/1 operator, which gives the exact match for the phrase and could be extremely helpful if the query does contains such phrase (and this is often the case). Compared with #NEAR, the #WINDOW operator isn’t that strict and can match the terms in a longer range and ignore their orders, which sometimes can be helpful, but there is no guarantee. Based on that, I assign higher weights to #AND and #NEAR operators, and compare the performance by adjusting the weights mainly between #AND and #NEAR.

**My observations:**

It can be seen clearly from the results that the sequential dependency model improves the performance a lot, and trend is that the more weight is assigned to the #NEAR operator, the better result you will obtain (this matches my analysis above that the #NEAR operator plays an important role in sequential dependency model). The result of the complex sequential dependency model matches my expectation well. And, actually, since the sequence dependent situation is very likely to happen in the queries (for example, the “heart rate”, “national park”), making use of these information can have significant improvement on the result (especially the precision). According to the time used, queries in sequential dependency model take only twice the whole time as that of the BOW model. Given that the great improvement on the result, I think the increased computational cost is worthwhile.

Discuss any trends that you observe; whether the more complex query behaved as you expected; whether the improvement in accuracy (if any) is worth the increased computational cost; and any other observations that you may have.

# Experiment 4: Multiple representations + SDMs

**Example Query:** Provide your structured query for query “fickle creek farm”.

#WAND(

0.1 #AND(

#WSUM(0.05 fickle.url 0.05 fickle.keywords 0.05 fickle.title 0.8 fickle.body 0.05 fickle.inlink) #WSUM(0.05 creek.url 0.05 creek.keywords 0.05 creek.title 0.8 creek.body 0.05 creek.inlink) #WSUM(0.05 farm.url 0.05 farm.keywords 0.05 farm.title 0.8 farm.body 0.05 farm.inlink))

0.9 #WAND(

0.3 #AND(fickle creek farm)

0.5 #AND(#NEAR/1(fickle creek) #NEAR/1(creek farm))

0.2 #AND(#WINDOW/8(fickle creek) #WINDOW/8(creek farm))) )

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Indri**  **BOW**  **(body)** | **w=1.0**  **(Exp 2)** | **w1=0.9** | **w2=0.7** | **w3=0.5** | **w4=0.3** | **w5=0.1** | **w=0.0**  **(Exp 3)** |
| **P@10** | 0.2400 | 0.2300 | 0.2500 | 0.2900 | 0.3400 | 0.3600 | 0.3700 | 0.3700 |
| **P@20** | 0.2750 | 0.2700 | 0.3000 | 0.3200 | 0.3400 | 0.3650 | 0.3750 | 0.3750 |
| **P@30** | 0.2967 | 0.2633 | 0.3033 | 0.3233 | 0.3333 | 0.3567 | 0.3600 | 0.3667 |
| **MAP** | 0.1275 | 0.1159 | 0.1292 | 0.1502 | 0.1635 | 0.1841 | 0.1883 | 0.1909 |
| **Time** | 00:09 | 00:21 | 00:22 | 00:22 | 00:22 | 00:21 | 00:22 | 00:22 |

In this part, I picked the queries that have the highest MAP in experiment 2 and 3 as the components. Given that the multiple representations don’t help much and the sequential dependency model does improve the results a lot, I suppose that the result will be better as weight w1 decreases (i.e., more weights are assigned to query from experiment 3), and the result turns out to be the same as what I expect. Comparing the result with that of sequential dependency model, I have to say that there isn’t any improvement in accuracy, so that the additional computational cost for the multiple representation part may be wasted. Based on this, I think it’s not worthwhile to increase the computational cost for these queries, and we can use sequential dependency model instead. Nevertheless, since the multiple representations don’t need much additional computational cost (i.e., the main cost in this structured query is from the sequential dependency model part), then if in some queries where the result can be improved by multiple representation to some degree, it’s definitely worthwhile to do so.

Discuss any trends that you observe; whether the more complex query behaved as you expected; whether the improvement in accuracy (if any) is worth the increased computational cost; and any other observations that you may have.