LIU Xi

xiliu1

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: LIU Xi

Your Andrew ID: xiliu1

Homework 3

1 Experiment 1: Baselines

	Ranked Boolean	BM25 BOW	Indri BOW
P@10	0.1500	0.3000	0.2300
P@20	0.1800	0.2950	0.2800
P@30	0.1667	0.2967	0.2900
MAP	0.0566	0.1304	0.1277

BM25: k_1=1.2, b=0.75, k_3=0.0

Indri:mu=2500, lambda=0.4

2 Experiment 2: Different representations

	Indri BOW (body)	0.1url 0.2 keywords 0.2 title 0.3 body 0.2 inlink	0.1 url 0.2 keywords 0.3 title 0.3 body 0.1 inlink	0.0 url 0.3 keywords 0.3 title 0.3 body 0.1 inlink	0.0 url 0.2 keywords 0.3 title 0.4 body 0.1 inlink	0.0 url 0.2 keywords 0.3 title 0.5 body 0.0 inlink
P@10	0.2300	0.1600	0.2300	0.2500	0.2200	0.2200
P@20	0.2800	0.1750	0.2050	0.2050	0.2200	0.2900
P@30	0.2900	0.2033	0.2267	0.2267	0.2400	0.3167
MAP	0.1277	0.0973	0.1073	0.1081	0.1082	0.1300
Time (ms)	20551	23180	22789	24429	23345	24443

Describe your strategy for setting the weights on the different representations.

Assuming keyword, body and title weights more while url and inlink weights less, at the beginning, I setup very close weights to each field, and gradually decreasing the weights of url and inlink. Also, I experimented with different combination of weights on keyword, title, and body to find which part should be weight more.

Discuss any trends that you observe

Increasing the weight of body field could improve MAP comparing among multi-representations. However, comparing with only searching in the body field, multi-representations could improve MAP, though some assignments of weights may harm the precision of top documents.

For this corpus, decreasing the weight of url and inlink could improve MAP, in other word, url and inlink weight less.

Whether the different representations behaved as you expected

Yes, I expected multi-representations could improve MAP than simply searching in body, because it extends the searching scope to get more documents matched. Besides, if there is a match in the title and keyword, the documents should be relevant to the query.

Precision and Recall characteristics of each representation

Compared with simply searching in body field, multi-representation could improve the precision on top documents, and a good combination of weights could improve the recall (MAP).

For each combinations of weights, smaller even zero weights for url and inlink, and larger weights for keywords, body and title could improve the precision for top documents, especially for top 30 documents. As for recalls, larger weights on url and inlink, that is, smaller weights on body, title, and keywords could harm the recall, this is probably because they do not contain query terms and the matching terms in other fields are under-weighted.

How the differences in accuracy (if any) relate to different computational cost

Multi-representations cost more than simply searching in the body, because the search engine has to iterate inverted files for each term in all fields, and combine the evidences. However, if we queries for the same fields, the different weights of combinations cost similar amount of time, because no matter how much each field weights, the search engine queries each inverted list in the same way.

Any other observations that you may have

Compared with column 1, column 2 gets precision and recall improved by down weighting inlink, and compared with column 3, column 4 gets precision and recall improved by down weighting url. This indicates for this corpus, inlink and url may not contain sufficient information about the document contents.

3 Experiment 3: Sequential dependency models

Example Query: Provide your structured query for query "fickle creek farm".

102:#wand(0.3 #and(fickle creek farm)

0.2 #and(#near/1(creek farm) #near/1(fickle creek))

0.5 #and(#window/8(creek farm) #window/8(fickle creek)))

	Indri BOW (body)	**0.3 AND 0.3 NEAR 0.4 WINDOW**	**0.2 AND 0.3 NEAR 0.5 WINDOW**	**0.1 AND 0.4 NEAR 0.5 WINDOW**	**0.3AND 0.5 NEAR 0.2 WINDOW**	**0.2 AND 0.4 NEAR 0.4 WINDOW**
P@10	0.2300	0.3600	0.3700	0.3600	0.3500	0.3700

P@20	0.2800	0.3650	0.3750	0.3750	0.3750	0.3750
P@30	0.2900	0.3700	0.3733	0.3667	0.3633	0.3633
MAP	0.1277	0.1755	0.1747	0.1858	0.1872	0.1878

Describe how you set the weights for the different components of the sequential dependency model.

Based on the assumption that #NEAR and #WINDOW work better than #And, I started with relatively equal weights (0.3, 0.3, 0.4) and experimented gradually increasing weights for #NEAR and #WINDOW, and decreasing weights for #AND. I also experimented with smaller weights for #NEAR and #WINDOW to test if my assumption is correct.

Discuss any trends that you observe;

First of all, sequential dependency model could improve the precision for top documents and recall. This is because with position restrictions (#NEAR and #WINDOW), documents with query terms appearing close to each other, which is more relevant based on our common sense, get higher score.

Then across all combination of weights, I found equal weights for #NEAR and #WINDOW, and smaller weight for #AND works the best for both precision and MAP.

Whether the more complex query behaved as you expected;

I expected decreasing weight for #AND could improve the top document precision, because #AND has larger searching scope for requiring neither the sequence nor the distances of terms, and over-weighted documents with query terms far away from each other. This works as I expected.

I expected increasing weight for #NEAR could improve the top document precision, because #NEAR only matches documents with query terms with a strict distance and sequence, which should be more relevant in our common sense, and it works as I expected.

Whether the improvement in accuracy (if any) is worth the increased computational cost;

Sequential dependency model improves the precision for top documents by nearly 30-50%, and improves MAP by no less than 50%. Though it takes nearly 2 times of querying time, most of the time is spent on I/O, thus the overall additional time cost is minor. Therefore, the improvement in accuracy is worth the increased computational cost.

Any other observations that you may have

I also expected with the same weight for #AND, increasing the weight for #WINDOW could improve the recall, because #NEAR may be too strict to miss some highly relevant document terms, because in our daily life the sequence of terms may not affect the expression. However, this does not work as I expected, I think it is because this corpus is large enough to have documents matching the restrictions set by #NEAR.

4 Experiment 4: Multiple representations + SDMs

Example Query: Provide your structured query for query "fickle creek farm".

102:#WAND (

0.3 #AND(#WSUM(0.0 fickle.url 0.2 fickle.keywords 0.3 fickle.title 0.0 fickle.inlink 0.5 fickle.body)

#WSUM(0.0 creek.url 0.2 creek.keywords 0.3 creek.title 0.0 creek.inlink 0.5 creek.body)

#WSUM(0.0 farm.url 0.2 farm.keywords 0.3 farm.title 0.0 farm.inlink 0.5 farm.body))

0.7 #WAND(0.2 #AND(fickle creek farm)

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

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

	Indri BOW (body)	**w=1.0 (Exp 2)**	**0.9**	**0.7**	**0.5**	**0.3**	**0.1**	**w=0.0 (Exp 3)**
P@10	0.2300	0.2200	0.3800	0.3900	0.3900	0.3900	0.3800	0.3700
P@20	0.2800	0.2900	0.3900	0.3900	0.3900	0.3900	0.3850	0.3750
P@30	0.2900	0.3167	0.3600	0.3600	0.3600	0.3600	0.3567	0.3633
MAP	0.1277	0.1300	0.1915	0.1923	0.1924	0.1923	0.1892	0.1878

Discuss any trends that you observe

Compared with baseline, the combination definitely achieved better accuracy.

Multi-representations and sequential dependency model alone cannot achieve as good accuracy as the combination. And the half and half weights achieved the best accuracy. This is probably because multi-presentations and sequential dependencies could make up each other's disadvantages.

Whether the more complex query behaved as you expected

Because SDMs has better precision for top documents and MAP, I expected the heavier weights on SDMs could achieve better accuracy. But queries do not behave as I expected. I think this is because multi-representations could help to extend the searching scope, and reward the documents with terms in title and keywords, which is more relevant in our common sense.

In the experiment, a balanced weight achieved the best accuracy. This is probably because multi-presentations and sequential dependencies could make up each other's disadvantages.

Whether the improvement in accuracy (if any) is worth the increased computational cost;

Because the search engine has to iterate inverted files for each term in all fields, and combine the evidences, multi-representations cost more than simply searching in the body. Also, sequential dependency model has 2 more iterations than simply searching with #AND, it costs more. However, most of the time is spent on I/O, thus the overall additional time cost is minor.

Compared with the baseline and multi-representation, MR+SDMs improves the precision for top documents by at least 40-50%, and improves MAP by over 50%. Though it takes more computational time, the improvement in accuracy is worth the increased computational cost. However, compared with SDMs, the improvement is relatively less significant, but is still worth the cost if we have sufficient computational resources.

Any other observations that you may have

By introducing the sequential dependency model with weight 0.1, it outperforms than multi-representations, so it is with introducing multi-presentations with small weight to sequential dependency model. 0.9-0.1 and 0.1-0.9, or 0.3-0.7 and 0.7-0.3 do not make a big difference on the result. Although multi-presentations and sequential dependencies could make up each other's disadvantages, the improvement could probably brought by sequential dependency model, and its influence is very strong. This is reasonable because the length of title and keywords are limited and may mismatch many query terms, so their influences are actually not very strong.