SLWWW at the NTCIR-14 We Want Web Task

Peng Xiao Tetsuya Sakai

Waseda University,Tokyo, Japan
xp1994@fuji.waseda.jp tetsuyasakai@acm.org

Abstract. SLWWW participated in English subtask of the NTCIR-14 We Want Web Task (WWW-2), and this paper describes our approaches we implemented during the task.

Keywords: Web Search · Terms Dependency Model · Lambdamart

TeamName

SLWWW

SubTask

English subtask

1 Introduction

The WWW-2 task[1] is an ad-hoc web search task, the task organizers adopt Clueweb12-B13 as the document collection. Unlike the NTCIR-13 WWW task[2], WWW-2 provides one-sentence description for each query. We built three models in this task. Two of them are based on field-based sequential dependency model[3], while the other is based on LambdaMart[4]. Unfortunately, the performances of our models are not good.

2 Runs

2.1 Sequential dependency model

We submitted two runs based Field sequential dependency model[3]. A Recent study[5] shows that it works well on the WWW task. For the first run, we only considered the tokens in query and ignored the description. Then we transferred the original unstructured query into an indri structured query. For example, a three-term query 'w1 w2 w3' without description can be represented as:

$$
\begin{aligned}
&\#weight\{\\
&\quad \alpha 1\#combine(w1.titile\ w2.title\ w3.title)\\
&\quad \alpha 2\#combine(w1.inlink\ w2.inlink\ w3.inlink)\\
&\quad \alpha 3\#weight\{\\
&\qquad \beta 1\#combine(w1.body\ w2.body\ w3.body)\\
&\qquad \beta 2\#combine(\#1(w1.body\ w2.boday)\\
&\qquad\qquad \#1(w2.body\ w3.body))\\
&\qquad \beta 3\#combine(\#uw8(w1.body\ w2.boday)\\
&\qquad\qquad \#uw8(w2.body\ w3.body))\\
&\qquad\quad \}\\
&\quad \}
\end{aligned}
$$

Here, $\alpha 1$, $\alpha 2$, $\alpha 3$, $\beta 1$, $\beta 2$, $\beta 3$ control the weight for each field. We set ($\alpha 1$, $\alpha 2$, $\alpha 3$, $\beta 1$, $\beta 2$, $\beta 3$) = (0.2, 0.05, 0.75, 0.8, 0.1, 0.1) in our experiment. (#1) and (#uw8) are represented bigram and 8-term unordered windows respectively The one-sentence description also contains some semantics which are highly related to query.We removed the stop words in the description and gave some weights to the rest tokens. For example, assumed we have a three-term query 'w1 w2 w3' and one-sentence description 'w4 w5 w6'. After filtering the stop words, only tokens w4 w6 are left. Then the indri structured query can be represented as:

$$
\begin{aligned}
&\#weight\{\\
&\quad \alpha 1\#combine(w1.titile\ w2.title\ w3.title)\\
&\quad \alpha 2\#combine(w1.inlink\ w2.inlink\ w3.inlink)\\
&\quad \alpha 3\#weight\{\\
&\qquad \beta 1\#combine(w1.body\ w2.body\ w3.body)\\
&\qquad \beta 2\#combine(\#1(w1.body\ w2.boday)\\
&\qquad\qquad \#1(w2.body\ w3.body))\\
&\qquad \beta 3\#combine(\#uw8(w1.body\ w2.boday)\\
&\qquad\qquad \#uw8(w2.body\ w3.body))\\
&\qquad\quad \}\\
&\quad \alpha 4\#weight\{\\
&\qquad \omega 1\#combine(w4.title\ w6.title\ w3.body)\\
&\qquad \omega 2\#combine(w4.body\ w6.body\ w3.body)\\
&\qquad\quad \}\\
&\quad \}
\end{aligned}
$$

Here,$\alpha1$, $\alpha2$, $\alpha3$, $\alpha4$, $\beta1$, $\beta2$, $\beta3$, $\omega1$ and $\omega2$ control the weight for each field. We set ($\alpha1$, $\alpha2$, $\alpha3$, $\alpha4$, $\beta1$, $\beta2$, $\beta3$, $\omega1$, $\omega2$) = (0.2, 0.05, 0.7, 0.05, 0.8, 0.1, 0.1, 0.6, 0.4) in our experiment.

2.2 LambdaMart

LambdaMart is a state-of art learning to rank algorithm proposed by Burges [4] in 2010. LambdaMart is a boosted tree version of LambdaRank, which is based on RankNext. It is widely used in real word problems and achieves good performance. Open source RankLib[6] is utilized to implement LamdaMart model in our experiment.

3 Data and Experiment

3.1 Data set

We used the NTCIR-13 WWW English test collection as our training data. The collection contains 100 topics with 22,912 relevance assessments from the clueweb12-B13 document collection. There are 80 test topics for WWW-2, and we only reranked the BM25-based baseline run provided by the organizers.

3.2 Feature extraction

In order to run the LambdaMart model, we have to extract various features to represent the intrinsic relationship between query and documents. For details see Table 1.

Table 1. Extracted features

feature name	description
term frequency	term frequency in title, head, body and link respectively
document frequency	document frequency in title, head, body and link respectively
bm25	bm25 score
length	length of title, head, body and link
terms num	terms number in query
exact match	boolean whether the query is exact match in document
all occurs	boolean whether all tokens in query occur in document
uw8	count the condition that 2 terms occur in 8 size unordered window
bigram	count the occurrence of query bigrams in document

4 Results and Discussions

We submitted three runs in this task. Run SLWWW-E-CO-NU-Base-1 is the search result based on LambdaMart. Run SLWWW-E-CO-NU-BASE-3 and Run SLWWW-E-CO-NU-BASE-4 are the search results based on Sequential dependency model. The difference between the two is that SLWWW-E-CO-NU-BASE-3 only uses tokens in the query to rank the documents, and SLWWW-E-CO-NU-BASE-4 also uses tokens from description but gives less weight in the structured query. The result is shown in Table 2.

Table 2. Official results

RUN	Mean nDCG@10	Mean Q@10	Mean nERR@10
SLWWW-E-CO-NU-Base-1	0.2860	0.2665	0.4188
SLWWW-E-CO-NU-Base3	0.2775	0.2499	0.4034
SLWWW-E-CO-NU-Base4	0.2767	0.2498	0.4015

5 Conclusions

This paper described our approaches to the WWW-2 English subtask. SLWWW-E-CO-NU-BASE-1 achieve the best performance among the three approaches. SLWWW-E-CO-NU-BASE-3 has a similar performance with SLWWW-E-CO-NU-BASE-4. That is, LambdaMart performed slightly better than the sequential dependency model. However, compared with other teams, our approaches still need to be improved.

.

References

1. J. Mao, T. Sakai, C. Luo, P. Xiao, Y. Liu and Z. Dou Overview of the NTCIR-14 We Want Web Task. In Proc. NTCIR-14 2019.
2. C. Luo, T. Sakai, Y. Liu, Z. Dou, C. Xiong, and J. Xu. Overview of the NTCIR-13 We Want Web Task. In Proc. NTCIR-13, 2017.
3. D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In Proc. SIGIR, pages 472479, 2005.
4. Chris J.C. Burges from RankNet to LambdaRank to LambdaMART: An Overview
5. Gallagher, L., Mackenzie, J., Benham, R., Chen, R.C., Scholer, F., Culpepper, J.S.: RMIT at the NTCIR-13 we want web task. In: Kando, N., Fujita, S., Kato, M.P., Manabe, T. (eds.) Proceedings of the 13th NTCIR Conference on Evaluation of Information Access Technologies, pp. 402406. National Institute of Informatics, Tokyo (2017)
6. Dang, V. "The Lemur Project-Wiki-RankLib." Lemur Project,[Online]. Available: http://sourceforge.net/p/lemur/wiki/RankLib