THUIR at the NTCIR-14 WWW-2 Task

Yukun Zheng, Zhumin Chu, Xiangsheng Li, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma

Department of Computer Science and Technology, Institute for Artificial Intelligence, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
yiqunliu@tsinghua.edu.cn

Abstract. The THUIR team participated in both Chinese and English subtasks of the NTCIR-14 We Want Web-2 (WWW-2) task. This paper describes our approaches and results in WWW-2 task. In the Chinese subtask, we designed and trained two neural ranking models on SogouQCL dataset. In the English subtask, we adopted learning to rank models by training them on MQ2007 and MQ2008 dataset. Our methods achieved the best performances in both Chinese and English subtasks.

Team Name. THUIR

Subtasks. Chinese and English

Keywords: web search · ad-hoc retrieval · document ranking

1 Introduction

A lot of learning to rank approaches have been proposed to address document ranking problem, such as AdaRank [20], LambdaMART [18] and etc. All these learning to rank algorithms usually need to be trained on effective hand-crafted features in the learning process. IR community has applied deep learning methods to advance state-of-the-art retrieval technologies. Guo et al. [5] suggested that most of recent neural ranking models can be generally classified into two categories according to the network architectures: 1) *Representation-focused model.* Models in this category first learn vector representations for textual queries and candidate documents separately with deep neural networks. Then the relevance is calculated by measuring the similarities between the two representations. This line of research includes DSSM [7], C-DSSM [15] and ARC-I [6], etc. 2) *Interaction-focused model.* ARC-II [6], DRMM [5], MatchPyramid [13] and K-NRM [19] belong to this category. The term-level interactions between queries

This work is supported by Natural Science Foundation of China (Grant No. 61622208, 61732008, 61532011) and the National Key Research and Development Program of China (2018YFC0831700).

and candidate documents are calculated first in these models. Then, the neural networks learn query-document matching patterns from these interactions. Mitra et al. [11] proposed to take advantages of both architectures in the Duet model. Luo et al. [9] showed the effectiveness of neural ranking models trained on large-scale weakly supervised data in ad-hoc retrieval. Self-attention mechanism [17] has been introduced into a number of NLP tasks, which helps models achieve better performances.

In the Chinese subtask, due to the effectiveness of neural method in the ad-hoc retrieval task, we design a Deep Matching Model with Self-Attention (DMSA), which combines both the interaction-focused and representation-focused frameworks and incorporates both weakly supervised relevance and human relevance in the training process. Besides, we also design a Simple Deep Matching Model (SDMM) which sequentially models the interaction of the query and each sentence. Specifically, we apply a local matching layer to capture the exact matching and semantic matching signals. We applied these two models reranking on the top results of baselines run. Experiment results show SDMM's state-of-the-art performance among all the submitted runs [10].

In the English subtask, we only try some learning to rank methods and BM25 because of the lack of large English datasets with relevance judgments. We submitted baseline run and another BM25 run based on a fine-grained document index as well as three runs of different learning-to-rank models. The submitted runs of learning to rank models, i.e., AdaRank [20], LambdaMART [18] and Coordinate Ascent [16], belong to pair-wise or list-wise methods, which are popular methods to be used in document ranking task. In our experiment, the results show that the learning to rank models perform much better than BM25 [10].

2 Chinese Subtask

2.1 Dataset

We adopt Sogou-QCL dataset [21] as the training data and use the NTCIR-13 test set as the validation set in the Chinese subtask. Sogou-QCL contains about 500 thousand queries and more than 9 million documents. All the query-document pairs are annotated with five click model-based relevance labels. In this paper, we choose PSCM-based relevance labels to train our model. Besides, Sogou-QCL provides a smaller dataset with 2,000 queries and about 50 thousand documents, where all the query-document pairs have 4-point scaled relevance labels from human annotators.

2.2 Deep Matching Model with Self Attention

In the Chinese subtask, we design a deep matching model with self-attention mechanism (DMSA). Figure 1 shows the framework of DMSA, which consists of two weakly supervised relevance predictors, BM25 score predictor (BM25 predictor) and click model-based relevance predictor (CM predictor), and a multi-relevance fusion predictor. BM25 predictor and CM predictor are used to predict

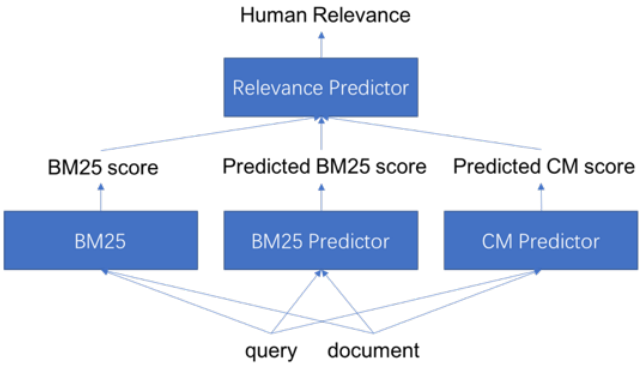

Fig. 1. The framework of DMSA.

the BM25 score and click model-based relevance respectively and share the same framework as shown in Figure 2. Multi-relevance fusion predictor are adopted to predict the human relevance based on the real BM25 score, the predicted BM25 score and the predicted score of click model-based relevance.

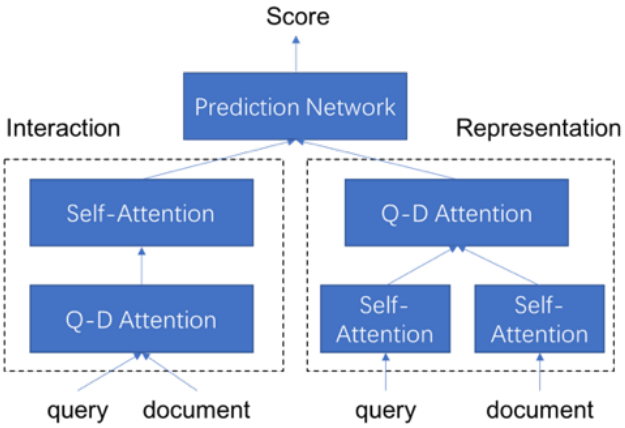

Fig. 2. The framework of weakly supervised relevance predictor.

Weak supervised predictor. In the weakly supervised predictor, we use an interaction-focused sub-model and a representation-focused sub-model to process the input of question and document terms simultaneously. Through each sub-model, we get two learned representations of the document. Then we use a multilayer perceptron to predict the weak relevance label based on the concatenation of the two learned representations of documents.

In the interaction-focused and representation-focused sub-models, we adopt the self-attention mechanism, which is very popular in NLP tasks, such as machine reading comprehension. We formulate the implementation of the attention mechanism in DMSA. Given a query $Q = \{q_1, ..., q_n\}$ and a document $D = \{d_1, ..., d_m\}$ as the inputs, where the query and the document consist of several terms, we first utilize a GRU [1] to learn the context-aware representations of the texts.

$$u_1, ..., u_n = \text{GRU}(q_1, ..., q_n) \tag{1}$$
$$v_1, ..., v_m = \text{GRU}(d_1, ..., d_m) \tag{2}$$

Given $U = \{u_1, ..., u_n\}$ and $V = \{v_1, ..., v_m\}$, the query-document attention is conducted as follows:

$$s_j^i = W^q u_j \odot W^d v_i \tag{3}$$
$$a_j^i = \exp(s_j^i) / \sum_{t=1}^{n} \exp(s_i^t) \tag{4}$$
$$c_j = \sum_{i=1}^{n} a_j^i u_i \tag{5}$$
$$h_j = W^h [c_j, v_j] \tag{6}$$

where $H = \{h_0, ..., h_m\}$ is the learned representation of the document after the query-document attention. In the self attention stage, we feed the term sequence of the query or the document as the input to conduct the attention with itself.

In the prediction network, we first get the representation vector of the document by adding all the term vector together and then feed it into a multilayer perceptron to predict the weak relevance label.

Multi-relevance fusion predictor. We use the real BM25 score, the predicted BM25 score and the predicted click model-based relevance score as the input and adopt a multilayer perceptron with one hidden layer to predict the human relevance.

2.3 Simple Deep Matching Model

Figure 3 shows the framework of our simple deep matching model (SDMM), which contains a local matching layer and a recurrent neural network (RNN) layer. The local matching layer aims to capture the semantic matching between query and sentence. The basic idea is to follow IR heuristics [4, 12] and qualify them into retrieval models.

Following the idea in [3], we apply term-level interaction matrix with both exact query matching and semantic query matching. Specifically, for a given query $\mathbf{q} = [w_1, w_2, ..., w_m]$ and a document $\mathbf{d}$ with T sentences, where each

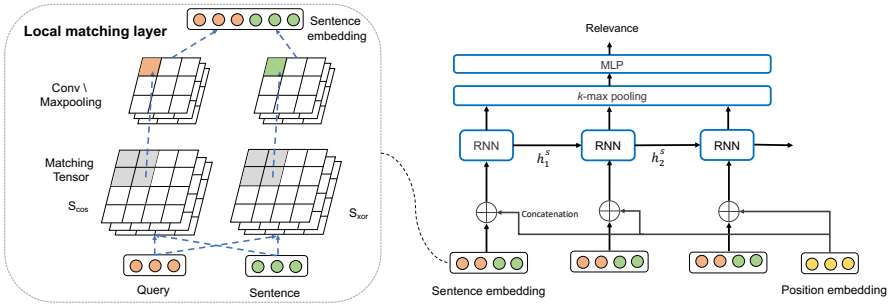

Fig. 3. The framework of SDMM.

sentence is $\mathbf{s} = [v_1, v_2, ..., v_n]$, we construct a semantic matching matrix M^{cos} and an exact matching matrix M^{xor}, which are defined as follows:

$$M_{ij}^{cos} = cos(w_i, v_j), \tag{7}$$

$$M_{ij}^{xor} = \begin{cases} 1, & w_i = v_j \\ 0, & otherwise \end{cases} \tag{8}$$

Exact matching and semantic matching provide critical signals for information retrieval as suggested by [4, 12]. To further incorporate term importance to the input, we extend each element M_{ij} to a three-dimensional representation vector $S_{ij} = [x_i, y_j, M_{ij}]$ by concatenating two term embeddings as in [3], where $x_i = w_i * \boldsymbol{W_c}$ and $y_j = v_j * \boldsymbol{W_c}$. $\boldsymbol{W_c}$ is a compressed matrix to be learned during training. The proximity of each word matching is retained in these matching matrices.

Based on two interaction matrices, we further apply CNN to generate local relevance embedding, which is also called sentence embedding. Note that CNN is more efficient than spatial GRU applied in [3] and it can also capture the relation between several adjacent words. The final sentence embedding is represented by concatenating the signals from two interaction matrices:

$$\boldsymbol{s} = [CNN(\boldsymbol{S}^{cos}), CNN(\boldsymbol{S}^{xor})] \tag{9}$$

Based on the sentence embedding from local matching layer, our model sequentially processes each input sentence, which is transferred to a RNN module:

$$h_t^s = RNN(h_{t-1}^s, s_t), t = 1, ..., T \tag{10}$$

where T is the number of total sentences of a document. Modeling sentences by RNN module is able to capture the context information in neighboring sentences. The RNN we used is Gated Recurrent Unit (GRU).

The hidden state $h^s_{1:T}$ are then utilized to estimate relevance by a k-max pooling layer and a full connected layer. k-max pooling layer selects top-k signals over all the sentences and full connected layer maps hidden states to a relevance score.

2.4 Experiment Setup

DMSA. We train the DMSA model in a point-wise and multi-task method to simultaneously predict human relevance, BM25 score and click model-based relevance of a query-document pair. We adopt mean squared error (MSE) as the loss function with Adadelta as the optimizer. The learning rate is 0.01 and The dropout rate is 0.2.

DSMM. We train the DSMM model in a point-wise learning method with mean squared error (MSE) as the loss function. We adopt Adadelta as the optimizer and the learning rate is 0.1.

2.5 Submitted Runs and Evaluation

We submitted 5 runs which were tested by the DMSA and SDMM models based on different numbers of top results in the baseline run, as shown in Table 1.

Table 1. Overview of runs in the Chinese subtasks.

Run	Model	Re-rank Range
THUIR-C-CO-MAN-Base-1	DMSA	10
THUIR-C-CO-MAN-Base-2	DMSA	100
THUIR-C-CO-MAN-Base-3	DMSA	45
THUIR-C-CO-CU-Base-4	SDMM	100
THUIR-C-CO-CU-Base-5	SDMM	40

Table 2 shows the evaluation results and ranks of our five submitted runs in the Chinese subtask. *THUIR-C-CO-CU-Base-5* achieves the best performance among all submitted runs, which is generated by SDMM model trained on weakly supervised data.

Table 2. Evaluation of runs in the Chinese subtasks. The table shows the mean value and the rank of the metric among all 10 runs submitted in the subtask.

Run	nDCG@10		Q@10		nERR@10	
THUIR-C-CO-CU-Base-5	**0.4916**	1	**0.4610**	1	**0.6374**	1
THUIR-C-CO-MAN-Base-2	0.4835	3	0.4604	2	0.5973	4
THUIR-C-CO-MAN-Base-1	0.4748	4	0.4479	4	0.6019	3
THUIR-C-CO-MAN-Base-3	0.4706	5	0.4364	5	0.5829	5
THUIR-C-CO-CU-Base-4	0.4458	9	0.4189	9	0.5663	7

3 English Subtask

Table 3. The features extracted for training learning-to-rank models

ID	Features
1	TF (Term frequency)
2	IDF (Inverse document frequency)
3	TF*IDF
4	DL (Document length)
5	BM25
6	LMIR.ABS
7	LMIR.DIR
8	LMIR.JM

In English subtask, we adopted learning-to-rank models. We introduce the details of our models in this section.

3.1 Features Extraction

First, we preprocessed the html files by using methods, including lowercasing, tokenization, removing stop words, and stemming.

Next, to train learning-to-rank model, we extracted features shown in Table 3. We extracted these eight features for four fields of a document: whole document, anchor text, title, and URL. In this way, we extracted $4 \times 8 = 32$ features for each document in total.

3.2 Dataset

We chose the MQ2007 and MQ2008 [14] as our training set. Although they provide the features we required, we calculated these features with our own algorithms to ensure the consistentency with the validation and test sets. At the same time, we used the NTCIR-13 WWW English testset [8] and its annotation results as our validation set, as its construction process is almost the same as that of this year's testset.

3.3 Methods and Results

We used the Ranklib [2] package to implement the learning-to-rank algorithms. We chose the LambdaMART, AdaRank, and Coordinate Ascent as the methods of our final submissions, because these models performed well on validation set. In the meantime, we submitted the baseline run and another BM25 run based on a fine-grained document index. Table 4 shows the performance of our runs in the English subtask, including the mean metric values and the ranks among the all 19 runs submitted in the English subtask. It indicates that our three learning-to-rank methods achieve the best performances among all runs submitted in the English subtask, while there is no significant difference between them.

Table 4. Evaluation of runs in the English subtasks. The table shows the mean value and the rank of the metric among all 19 runs submitted in the subtask. *LM* and *CA* means *LambdaMART* and *Coordinate Ascent* respectively.

Run	Model	nDCG@10		Q@10		nERR@10	
THUIR-E-CO-MAN-Base-1	AdaRank	0.3444	4	0.3249	6	0.5048	1
THUIR-E-CO-MAN-Base-2	LM	0.3512	2	0.3391	1	0.5026	2
THUIR-E-CO-MAN-Base-3	CA	0.3536	1	0.3256	4	0.4805	4
THUIR-E-CO-PU-Base-4	BM25	0.3294	8	0.3161	8	0.4692	8
THUIR-E-CO-PU-Base-5	baseline	0.3258	11	0.3043	11	0.4779	5

4 Conclusion

In NTCIR-14 WWW-2 task, we participated and got the best performances of runs in both Chinese and English subtasks. In the Chinese subtask, we designed two deep ranking models, which have been shown to be effective in ad-hoc retrieval. In the English subtask, we adopt learning to rank methods and trained them on MQ2007 and MQ2008 dataset. In the future, we would like to investigate how to better combine human relevance labels and weak supervised relevance labels in the ad-hoc retrieval task and how to better take fine-grained matching signals into our ranking models.

References

1. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
2. Dang, V.: The lemur project-wiki-ranklib. lemur project (2012)
3. Fan, Y., Guo, J., Lan, Y., Xu, J., Zhai, C., Cheng, X.: Modeling diverse relevance patterns in ad-hoc retrieval. International ACM SIGIR Conference on Research and development in Information Retrieval pp. 375–384 (2018)
4. Fang, H., Tao, T., Zhai, C.: A formal study of information retrieval heuristics. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 49–56. ACM (2004)
5. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: A deep relevance matching model for ad-hoc retrieval. In: CIKM'16 (2016)
6. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: NIPS'14 (2014)
7. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: CIKM'13 (2013)
8. Luo, C., Sakai, T., Liu, Y., Dou, Z., Xiong, C., Xu, J.: Overview of the ntcir-13 we want web task. NTCIR-13 (2017)
9. Luo, C., Zheng, Y., Mao, J., Liu, Y., Zhang, M., Ma, S.: Training deep ranking model with weak relevance labels. In: Australasian Database Conference. pp. 205–216. Springer (2017)
10. Mao, J., Sakai, T., Luo, C., Xiao, P., Liu, Y., Dou, Z.: Overview of the ntcir-14 we want web task. In: Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies (2019)

11. Mitra, B., Diaz, F., Craswell, N.: Learning to match using local and distributed representations of text for web search. In: WWW'17 (2017)
12. Pang, L., Lan, Y., Guo, J., Xu, J., Cheng, X.: A deep investigation of deep ir models. arXiv preprint arXiv:1707.07700 (2017)
13. Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., Cheng, X.: Text matching as image recognition. In: AAAI'16 (2016)
14. Qin, T., Liu, T.Y.: Introducing letor 4.0 datasets. arXiv preprint arXiv:1306.2597 (2013)
15. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: Learning semantic representations using convolutional neural networks for web search. In: WWW'14 (2014)
16. Uysal, I., Croft, W.B.: User oriented tweet ranking: a filtering approach to microblogs. In: Proceedings of the 20th ACM international conference on Information and knowledge management. pp. 2261–2264. ACM (2011)
17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NIPS'17. pp. 5998–6008 (2017)
18. Wu, Q., Burges, C.J., Svore, K.M., Gao, J.: Adapting boosting for information retrieval measures. Information Retrieval **13**(3), 254–270 (2010)
19. Xiong, C., Dai, Z., Callan, J., Liu, Z., Power, R.: End-to-end neural ad-hoc ranking with kernel pooling. In: SIGIR'17 (2017)
20. Xu, J., Li, H.: Adarank: a boosting algorithm for information retrieval. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 391–398. ACM (2007)
21. Zheng, Y., Fan, Z., Liu, Y., Luo, C., Zhang, M., Ma, S.: Sogou-qcl: A new dataset with click relevance label. In: SIGIR'18 (2018)