THUIR at NTCIR-13 WWW Task

Yukun Zheng, Cheng Luo, Weixuan Wu, Jia Chen, Yiqun Liu, Huanbo Luan, Min Zhang, Shaoping Ma
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
yiqunliu@tsinghua.edu.cn

ABSTRACT

This paper describes our approaches and results in NTCIR-13 WWW task. In English subtask, we adopt several advanced deep models, like DSSM and DRMM. In Chinese subtask, we additionally make a few changes in models to ensure them work well in the Chinese context and train the Duet model with the weak-supervised relevance labels generated by various click models. Meanwhile, We extract 3 types of features from data corpus to train a learning to rank model.

Team Name

THUIR

Subtasks

WWW (Chinese, English)

Keywords

web search, ad-hoc retrieval, document ranking

1. INTRODUCTION

In NTCIR-13, THUIR group participated in WWW task, including both Chinese and English subtasks. In recent years, deep models have achieved good results in the field of information retrieval. In this task, we try to apply these successful works to Chinese and English subtasks in NTCIR13 WWW task, verifying these models' effects and looking forward to get a good performance.

In Chinese subtask, we submit five re-rank runs of the documents results, 3 runs from the Duet model using various click models, other two runs from DRMM and L2R model. Noting that the most advanced models like Duet and DRMM are proposed in the English search environment, we need to migrate them to Chinese context, ensuring that they can identically work well with the Chinese web corpus.

After transferring models' application scenario from English to Chinese, the next challenge is the lack of train data. Although there are 200 queries in each language offered in WWW task, due to deep model's data-hunger characteristic, only larger size of train data with high-quality relevance labels can lead to a better and more credible performance. Thus, we introduce a weakly-supervised method to get the relevance labels between queries and documents automatically. By this way, an URL's labels become its click probability calculated by click models. All the queries and its urls are extracted from the query logs of an commercial search engine—*Sogou* with millions of users everyday. As a result, the size of data from query logs is quite enough to train a neu-ir model.

What's more, with lots of click information in query logs, click models can predict the relevance of query-document pairs without the influence of position bias. Finally, we combine the changed models and the weak labels from click models. Four runs in Chinese task are generated by this method. we also use the provided 200 Chinese queries and high-quality query-documents pairs' relevance judgment to train a Learning to Rank model. 14 features are extracted from queries, documents and their interaction, like query length, the length of title in a document and semantic similarity between them. In order to take full advantage of existing click information from query logs, we propose a novel score algorithm to extend the feature vector to 17 dimensions. Therefore, taking account of user's click behavior, the trained L2R model performs better.

In the English subtask, we upload 4 re-rank runs respectively from DRMM, CDSSM, DSSM and L2R method. What is different from the Chinese case is the shortage of English query logs. Based on this reality, we train a L2R model with only 14 dimensions of features. In the left three methods, we directly use pre-trained models to re-rank the doc list, avoiding the lack of train data.

The rest of the paper is organized as follows. In Section 2, we list some related works. In Section 3 and 4, we introduce the methods we use in the Chinese and English subtask respectively. Section 5 concludes the paper.

2. RELATED WORK

After many years' development, information retrieval and recommendation systems can not be separated from learning to rank. Many models are proposed to improve the rank list, such as RankSVM[9], ListNet[2] and LambdaMart[1] etc. Plenty of previous works show that LambdaMart is much more better and stable than others, so we choose it as our learning to rank method in this task. The input of L2R models is the hand-crafted feature vectors extracted from queries and documents, including statistics and semantic features.

Since deep model have shown its powerful ability in computer vision , speech recognition and NLP last few years, researchers try to explore the potential of deep learning in information retrieval. Now deep learning has made much progress nowadays and lots of deep models have been proposed from different perspectives to address the puzzles in the field of IR, particularly in ad-hoc search. Huang et

al.[8] proposed DSSM, which can represent text strings in a continuous semantic space and model semantic similarity between two text strings. After that, Shen et al.[12] presented a new convolutional deep structured semantic models(C-DSSM) to learn low dimensional semantic vectors for search queries and Web documents. Guo et al.[7] propose DRMM model and emphasize that the IR task is different from that in NLP. In ad-hoc retrieval, how to learn the relevance of query and document pairs is the core of the problem, instead of the semantic match between query and doc. Mitra et al.[11] construct a duet model to learn from both local and distributed representation of text and prove its better performance than only single representation's.

Recently, a few researches pay attention to how to generate much cheaper and huger amount of relevance label for deep models. Dehghani et al.[5] proposed to use traditional IR models like BM25 as a weak supervision signal. MacAvaney et al.[10] utilize the news articles to train models mentioned above. Xiong et al.[14] train K-NRM, model with data labeled by click models. All these works inspire us to introduce this approach in the task.

In a search session, the system will collect many types of user behavior, in which click action implies some strong relation between query and the clicked document. Thus, click models are established to model user's click behavior and can predict the probability of the position where user will click next time. many click models are proposed based on different assumptions, like RCM[4], TCM[15], UBM[6], DBN[3] and PSCM[13]. The most recent work also takes time factors into account. All these models can supply us a large amount of relevance judgements between queries and documents, much stronger than the signals from probability models like BM25.

3. CHINESE DOCUMENT RANKING

Although a small Chinese data corpus were provided, it's not enough to train a model well. So we need to generate a larger data set for training. On the other hand, most existing models are proposed to run in the English context, instead of Chinese context. Consequently, we are mainly facing two challenges: (1) We don't have enough Chinese TREC-style and high-quality relevance judgements to support our experiment. (2) Models in previous works can not directly run on the Chinese corpus and we need us make some changes to get along well with the Chinese corpus. We will discuss about our approaches to addressing these two challenges in this section.

3.1 Weak Relevance Label Generation

We use click model to obtain enough Trec-style data for training. Click-through behavior during Web search reflects implicit feedback of users' click preferences. User clicks are biased toward many aspects: (1) position bias: users tend to prefer the documents higher in the ranking list; (2) novelty bias: previously unseen documents are more likely to be clicked; (3) attention bias states that the impact of visually salient documents.

Click model is designed to predict the clicked probability (Click- through rate) of document results in a search task. After eliminating the influence of various factors, we can infer the document relevance based on the click probability predicted by click models.

In this task, we adopt an open-source implementation of these models, including several popular click models, such as DBN, RCM, PSCM, TCM, and UBM. We trained these click models with a real-world dataset collected by Sogou. We removed all the queries that appeared less than 10 times (i.e. less than 10 sessions) since it seem unlikely to train a reliable click model with insufficient behavior data. For each query, at most 500 search sessions are selected for click model training to keep a balance between model precision and the amount of calculation. The statistics of our behavior dataset is shown in Table 1.

Figure 1 shows the distributions of documents relevance generated by five kinds of click models. The x-axis is the documents relevance ranged from 0 to 1 while the y-axis is the number of corresponding documents in logarithmic scale. We can see that more advanced the model is, the more evenly its distribution is. We finally submit three runs in Chinese task using TCM, DBN and PSCM.

For L2R method, we use the officially provided 200 queries, related documents and relevance judgements. We remove a lot of repeated urls under a same query. We collect 14 dimensions of statistical and semantic features and 3 dimensions of click behavior features. Table 3 shows all the feature types.

All the Statistical features are calculated after the Chinese words segmentation and the length is the number of Chinese tokens. We use a large pre-trained word embedding to get the cosine similarity between query and some parts of the document. The last type of feature is based on the click behavior collected from Sogou query logs. First, we find queries which both appear in logs and data corpus. Second, the score of all tokens in the clicked documents' title, tag h6-h1 and content is its frequency of appearance in a search process of this query. Finally, The corresponding dimensions in feature vector is the sum of the tokens' scores. This way help us to combine the click information and the exact match. However, there are still part of queries not existing in the query logs, so we use these queries to train a L2R model with only 14 features while those ones appearing in the query logs are extracted to 17 features. Finally, we merge the two part of results to one and submit it.

3.2 Modified models

We use the Duet and DRMM in this task. we extend the Duet model's n-gram list by adding top 5000 of most frequent Chinese n-gram terms into it. Meanwhile, we use jieba toolkit[1] to segment Chinese sentences into terms.

After the modifications on Duet model, we wonder if it can work well in the Chinese context and how much gap of performance it is between weak label and human judgement training method. So we train Modified model with the two types of data: (1) Weak Relevance Label: as mentioned before. We have estimated click probability for query-document pairs. (2) Strong Relevance Label: we have 200 queries which are released for NTCIR WWW task. For each query, there are some documents whose relevances are judged by professional assessors in a five level scale (from irrelevant to high relevant).

We verify the performance of Duet model trained with strong relevance labels by randomly splitting the strong relevance label dataset into two parts: training set with 150 queries while test set with 50 queries.

[1]https://github.com/fxsjy/jieba

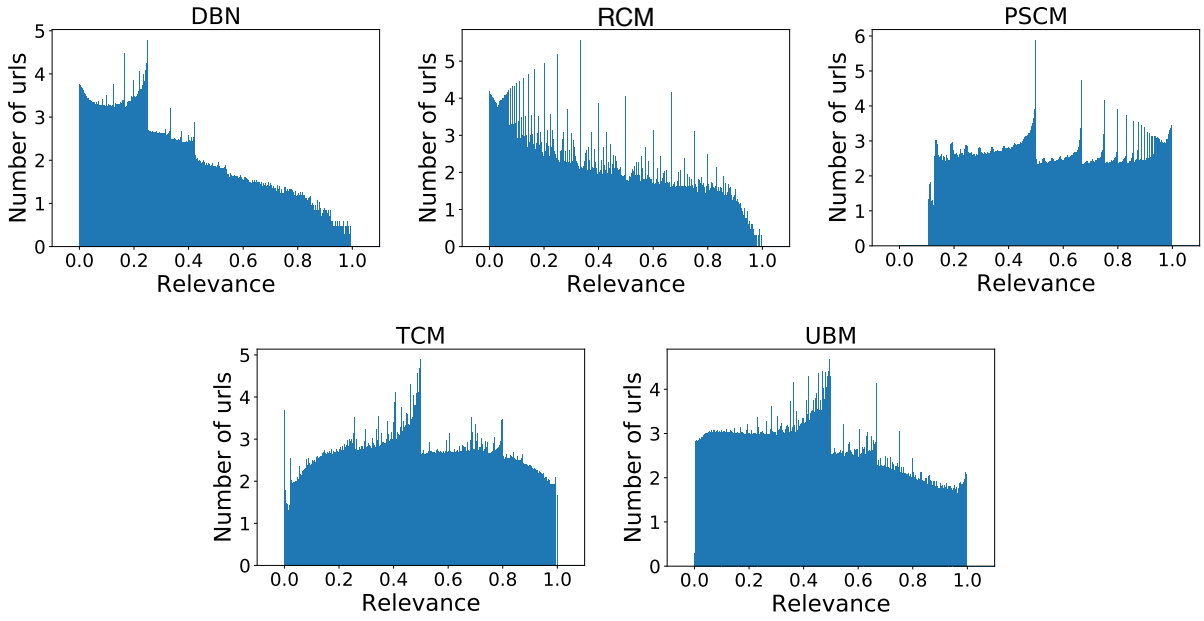

Figure 1: Distributions of click probability from various click models

Table 1: Evaluation of runs in Chinese and English. The table shows the mean metrics value and the rank among all the 19 Chinese and 13 English runs submitted in WWW task.

Subtask	Run	Model	nDCG@10	Q@10	nERR@10
Chinese	THUIR-C-CU-Base-1	Duet(PSCM)	**0.4828/11**	**0.4942/11**	**0.6443/11**
	THUIR-C-CU-Base-2	Duet(TCM)	0.4179/14	0.4235/14	0.5626/15
	THUIR-C-CU-Base-3	Duet(DBN)	0.4137/15	0.4144/15	0.5717/12
	THUIR-C-CU-Base-4	LambdaMart	0.4258/13	0.4335/13	0.5695/14
	THUIR-C-CU-Base-5	DRMM	0.4258/12	0.4335/12	0.5695/13
English	THUIR-E-PU-Base-1	DRMM	0.5323/7	0.5369/6	0.6754/7
	THUIR-E-PU-Base-2	CDSSM	0.5360/6	0.5304/7	0.6744/8
	THUIR-E-PU-Base-3	DSSM	**0.5679/2**	**0.5570/5**	**0.7118/3**
	THUIR-E-PU-Base-4	LambdaMart	0.3157/13	0.3018/13	0.4648/13

Table 2: The statistics of user behavior dataset.

Num of Queries	544,296
Num of Documents	12,973,303
Date	From Apr. 1st 2015 to Apr. 18 2015
Language	Chinese

we evaluate all the ranking models based on the test set. We use ERR, nDCG@10 and Q-measure, which is calculated with an open-source tool NTCIREVAL2. We also introduce a widely used baseline method BM25.

From table 4, we can see that the models which are based on strong labels are slightly better than that are based on weak labels across all evaluation metrics. We assume that rankers with strong labels are trained with much more query-doc pair samples, leading a better optimization of model's parameters. On the other hand. The approximative performances of models with weak label give us strong confidence that once these models are fed with enough data, they can perform better even than that with strong label.

Table 3: Types of features in L2R.

Type	Feature
Statistical	Query length
	Length of doc's content
	Length of doc's h1-h6
	Length of doc's title
Semantic	Similarity between query and doc's content
	Similarity between query and doc's h1-h6
	Similarity between query and doc's title
Behavior	Score of doc's content
	Score of doc's h1-h6
	Score of doc's title

Table 4: Comparison between models which are based on strong/weak relevance labels (The ranker with best performance trained on weak labels is marked in bold while that trained on strong labels is marked with underline.)

Dataset	ERR	nDCG@10	Q	#Pair
Strong	0.5418	0.5578	0.6800	469,790
DBN	0.5180	0.5374	**0.6385**	11,251
RCM	0.4924	0.4922	0.5761	18,554
PSCM	0.5151	0.5364	0.6344	59,296
TCM	0.5124	0.5412	0.6359	42,268
UBM	**0.5302**	**0.5542**	0.6333	34,537

We choose the absolute method (ABS) to make document pair in a pairwise model both in the task and the experiment mentioned above. We simply split the probability into 4 segments and each segments represents a relevance level respectively, e.g. the relevance score is 1 if the click probability is between 0.0 to 0.25. Then we adopt the method in Mitra et al.'s study [17] to organize document pairs. For a document pair <D+,D->, the relevance scores of two documents can be 3 v.s. 1/0, or 2 v.s. 0. The DRMM model needs pre-trained word embedding to represent queries and documents. So we trained it on a huge web corpus, including more than 4 millions Chinese words.

3.3 Evaluation Results

We upload five runs in the Chinese subtask. Table 1 shows the mean measure metrics of our approaches in Chinese subtask. We can see that THUIR-C-CU-Base-1 performs best among all five runs and even beats the DRMM model, indicating that the click model's effect positively affects the deep model's performance.

4. ENGLISH DOCUMENT RANKING

In English subtask, we submit 4 re-rank runs of documents list and all deep models used here is pre-trained. DRMM were trained by the rob04-title dataset while DSSM and CDSSM were utilized to calculate the semantic similarity between query and doc's title, tag h1-h6 and whole content. The three scores are added together after multiplied by the weights [1.0, 0.1, 0.2]. Table 1 shows us the performance of each runs in English subtask. From it, we can know that the DSSM model perform best while LambdaMart model gets the poorest performance.

5. CONCLUSIONS

During Chinese subtask, we try to run a few advanced deep models using query logs which are labeled by weak supervised method. In English subtask, we use some pre-trained models like DSSM and CDSSM, which output some satisfactory results without much efforts to adjusting parameters. Although our models can optimize the whole rank, they can not recognize the most relevant documents which should be ranked in the top 10 positions.

After the WWW task, we will try to find out why our models do not perform as well as we expect. We guess there may be two reasons: (1) the weak labels generated by various click models is different from artificial judgements. (2) Some settings in our models restrict their performances, such as the size of ngraphs list in Duet model. Thus, we will design some experiments to verify whether our guess is true. Xiong et al. prove that their model with only documents' title information can get the same results as models with documents' full text information do, sometimes even better. Due to the long time to train a model on documents' full text, maybe only using the information of documents' title is a good way. In the future, we want to try more new models and add some new designs to them, looking forward to having a better performance in the next two-years' task.

6. REFERENCES

[1] C. J. Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81, 2010.

[2] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *ICML '07*, pages 129–136. ACM, 2007.

[3] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *WWW '09*, pages 1–10. ACM, 2009.

[4] A. Chuklin, I. Markov, and M. d. Rijke. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 7(3):1–115, 2015.

[5] M. Dehghani, H. Zamani, A. Severyn, J. Kamps, and W. B. Croft. Neural ranking models with weak supervision. *arXiv preprint arXiv:1704.08803*, 2017.

[6] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *SIGIR '08*, pages 331–338. ACM, 2008.

[7] J. Guo, Y. Fan, Q. Ai, and W. B. Croft. A deep relevance matching model for ad-hoc retrieval. In *CIKM '16*, pages 55–64. ACM, 2016.

[8] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *CIKM '13*, pages 2333–2338. ACM, 2013.

[9] T. Joachims. A support vector method for multivariate performance measures. In *ICML '05*, pages 377–384. ACM, 2005.

[10] S. MacAvaney, K. Hui, and A. Yates. An approach for weakly-supervised deep information retrieval. *arXiv preprint arXiv:1707.00189*, 2017.

[11] B. Mitra, F. Diaz, and N. Craswell. Learning to match using local and distributed representations of text for web search. *arXiv preprint arXiv:1610.08136*, 2016.

[12] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. Learning semantic representations using convolutional neural networks for web search. In *WWW '14*, pages 373–374. ACM, 2014.

[13] C. Wang, Y. Liu, M. Wang, K. Zhou, J.-y. Nie, and S. Ma. Incorporating non-sequential behavior into click models. In *SIGIR '15*, pages 283–292. ACM, 2015.

[14] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power. End-to-end neural ad-hoc ranking with kernel pooling. *arXiv preprint arXiv:1706.06613*, 2017.

[15] Y. Zhang, W. Chen, D. Wang, and Q. Yang. User-click modeling for understanding and predicting search-behavior. In *KDD '17*, pages 1388–1396. ACM, 2011.