THUIR at the NTCIR-15 WWW-3 Task

Zhumin Chu, Jingtao Zhan, 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
Beijing, China
yiqunliu@tsinghua.edu.cn

ABSTRACT

The THUIR team participated in both Chinese and English subtasks of the NTCIR-15 We Want Web-3 (WWW-3) task. This paper describes our approaches and results in the WWW-3 task. In the Chinese subtask, we tried two kinds of neural ranking models based on BERT, as well as a revived SDMM model. In the English subtask, we revived three learning-to-rank runs and a BM25 run we submitted in WWW-2 English subtask, we also tried a new ranking system based on BERT.

TEAM NAME

THUIR

SUBTASKS

Chinese, English

1 INTRODUCTION

Ad hoc Web search is a long-established research topic in information retrieval. From the traditional BM25[16], language models[21], towards a series of learning-to-rank models[9], and the newest deep learning models[5–7], many researchers have focused on improving the performance of Web retrieval.

We Want Web (WWW) is a series of ad hoc web search tasks, which improves communication among the researchers in the community. In the tasks, provided with topics and their descriptions set, as well as baseline ranking results, we need to return a ranking list under each query, to improve the ranking performance.

In this round of the NTCIR-15 WWW-3 task[17], we participated in both Chinese and English subtasks. In the Chinese subtask, we tried two kinds of neural ranking models based on BERT[5], as well as a revived SDMM model. We found that the reranking depths can greatly affect performance in the BERT model, using a smaller depth might lead to better performance. In the English subtask, we revived three learning-to-rank runs and a BM25 run we submitted in WWW-2 English subtask, we also implemented a new ranking model based on BERT. The results show that the learning-to-rank models perform better than BM25, but the BERT model does not perform as well as expected because the training set is limited.

This work is supported by the National Key Research and Development Program of China (2018YFC0831700), Natural Science Foundation of China (Grant No. 61732008, 61532011), Beijing Academy of Artificial Intelligence (BAAI) and Tsinghua University Guoqiang Research Institute.

2 CHINESE SUBTASK

2.1 BERT on Document Content (BERT-DC)

BERT [5] has been widely adopted in ranking tasks [13, 14]. Benefitted from the pretrain stage and Transformer architecture, it significantly outperforms other neural ranking methods and traditional IR techniques [3]. Thus we utilize it for the WWW-3 Chinese task. We exploit SogouQCL [24] as our training data and finetune the BERT model on the content and titles of documents, respectively.

Recently, several works [8, 22, 23] suggest that BERT's modeling process of document and query can be decoupled with minor influence on its ranking performance. Inspired by these works, we mask the attention from document towards query. In other words, the modeling process of document is independent of query but the query can still interact with document tokens in each layer. We ensemble this masked version of BERT with vanilla BERT as our final model.

We implement our models based on a widely-used library of transformers [18]. Most hyper-parameter settings are the same as Rodrigo et al. [13]. We adopt the bert-base-chinese model and finetune all the models for $300,000$ steps with sigmoid loss and a batch size of 80. Our experiments show that the optimal reranking depth is closely related to the dataset. For the NTCIR-14 Chinese task [11] and the NTCIR-13 Chinese task [10] task, BERT-DC performs best with a reranking depth of 100 and 40, respectively. We tuned the reranking depths on the NTCIR-14 Chinese task, and finally used depths of 70, 100, and 120 in our submitted runs.

2.2 BERT on Document Title (BERT-DT)

To validate that the training source must be consistent during training and testing, we also train the BERT model on the document title. In this section, we use Tiangong-ST [2], which provides session logs with click labels. The document title is used to train BERT. The maximal length of the document title is set as 15. Query and document title is concatenated as the input for the bert-base-chinese model. We train this model for $300,000$ steps with pairwise hinge loss and a batch size of 80.

2.3 Revived model on NTCIR-14 (SDMM)

The revived model is based on the Simple Deep Matching Model (SDMM) on NTCIR-14, which achieves the best-ranking performance in the NTCIR-14 Chinese task. We use the same trained model to generate a run for the queries on NTCIR-15. The result in *THUIR-C-CO-REV-5* in Table 1.

Table 1: WWW-3 Chinese subtask official results of THUIR 1-5 runs

Model	RUN	Rerank Depth	nDCG	Q	ERR	iRBU
BERT-DC	**THUIR-C-CO-NEW-2**	70	**0.4112**	**0.3525**	**0.5706**	**0.7751**
BERT-DC	**THUIR-C-CO-NEW-1**	100	0.4051	0.3464	0.5489	0.7493
BERT-DC	**THUIR-C-CO-NEW-3**	120	0.3940	0.3325	0.5169	0.7356
BERT-DT	**THUIR-C-CO-REV-5**	100	0.2705	0.2093	0.4065	0.6384
SDMM	**THUIR-C-CO-NEW-4**	100	0.2329	0.1728	0.3489	0.6011

2.4 Results

The results of BERT-DC model are shown in Table 1 (THUIR 1-3 runs). The three submitted runs use the same BERT-DC model but different reranking depths. The results show that the reranking depths greatly affect ranking performance. The THUIR-C-CO-NEW-2 run performs best due to its smallest reranking depth. Thus, we assume using a smaller reranking depth may lead to better ranking performance.

In a comparison of document content and title, we can find that BERT trained on the title is not as good as that on document content. It illustrates that the training source should be consistent during training and testing.

The revived model, which achieves best-ranking performance in the NTCIR-14 Chinese task, performs worse than BERT-DC. It illustrates that BERT, with powerful learning capacity to learning the interaction between query and document, can achieve marginal improvement than a simple deep IR model.

3 ENGLISH SUBTASK

In the WWW-3 English subtask, we have submitted three learning-to-rank runs (revived runs), one neural model run (new run), and one fine-grained BM25 run (replicated run). We'll introduce the details about our runs in this section.

3.1 Data Preprocess

To better feature extraction and token embedding, we conducted a very detailed data preprocessing job. We parsed the HTML documents with the bs4 package, to obtain the context of four fields: the whole HTML content, the uniform resource locator (URL) of this HTML, the anchor texts, and the title. We ignored the <script> and <style> tags in the HTML documents, and consider the incompleteness condition of the HTML documents, to make the procedure more robust.

Then, for the contexts of each field, we adopted some natural language processing methods to make them more standard. The methods include lowercase, stop words deleting, and stemming. We also split the URL information in the content, to make them become a series of terms rather than a whole. For example, for the URL 'https://www.baidu.com', we split it into four terms: 'https', 'www', 'baidu', 'com'. We assumed that this procedure can improve system performance, especially in navigational queries. Also, We adopted the same preprocessing procedures towards the contents of the queries, to make them the same.

As for the data preprocessing procedure, there are two details to explain here. The first is the difference between the preprocessing for the feature extraction and the token embedding. As many pre-trained token embedding undo the stop words deleting and stemming procedure while training, as their pre-trained corpus is large enough to be the inclusion of information redundancy in the original data. So these two steps are omitted when feeding the preprocessed result to the token embedding procedure. Another detail is about the MQ2007 and MQ2008 datasets[15]. We found that the queries content of these two datasets has already been stemmed. So we conduct the counter-stemming step for the queries content, to make the same format with the token set.

3.2 Feature Extraction

Features are quite important for learning-to-rank systems. For each pair of query and document, we have extracted 8 features in each field, that is totally $4\times8 = 32$ features. These eight types of features include Term Frequency (TF), Inverse Document Frequency (IDF), TF*IDF, Document Length (DL), BM25, LMIR.ABS, LMIR.DIR, LMIR.JM. The calculation formula for the BM25 score shows in the Eq 1, and we set the parameter $k_1 = 1.2, k_2 = 100, b = 0.75$. Also, the language model can be calculated with the formula Eq 2. The details and parameter selection can be seen in Zhai et al.'s work[21].

$$BM25(d,q) = \sum_{i=1}^{M} \frac{IDF(t_i) \cdot TF(t_i,d) \cdot (k_1+1)}{TF(t_i,d) + k_1 \cdot \left(1 - b + b \cdot \frac{len(d)}{avgdl}\right)} \tag{1}$$

$$\log p(q|d) = \sum_{i:c(q_i;d)>0} \log \frac{p_s(q_i|d)}{\alpha_d p(q_i|C)} + n \log \alpha_d + \sum_i \log p(q_i|C) \tag{2}$$

It is worth mentioning that we used a part of the ClueWeb12 data set (about 5,000,000 HTML documents) as a background data set to obtain IDF and BM25 features, making that these features become more representative.

3.3 Learning-to-Rank Systems

We can regard the learning-to-rank systems as a black-box model. We feed the features of the sample queries and their corresponding documents to the systems, after a series of the parameter optimization process, the systems return a model to predict the ranking list of the given queries. In our work, we tried three types of learning-to-rank models: LambdaMART[1, 19] , Coordinate Ascent[12] and AdaRank[20], and used the Ranklib[4] package to implement them.

https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Table 2: Evaluation of runs in the English subtasks on the WWW-3 topic set. The table shows the mean value and the rank of the metric among all 37 runs submitted in the subtask. *LM* and *CA* means *LambdaMART* and *Coordinate Ascent* respectively.

Run	Model	nDCG@10		Q@10		nERR@10		iRBU@10	
THUIR-E-CO-REV-1	CA	0.5994	19	0.6068	22	0.7190	18	0.9112	10
THUIR-E-CO-REV-2	LM	0.5876	23	0.6010	23	0.7133	21	0.8829	28
THUIR-E-CO-REV-3	AdaRank	0.6049	18	0.6241	18	0.7102	23	0.9007	15
THUIR-E-CO-NEW-4	BERT	0.5112	33	0.5250	32	0.6481	32	0.8579	34
THUIR-E-CO-REP-5	BM25	0.4767	35	0.4899	35	0.6021	35	0.8259	35

3.4 BERT Models

BERT models are quite popular in the region of natural language processing. Many researchers also tried to apply the BERT models into their fields, such as sentiment analysis, question answering, sentence tagging, and so on. In document retrieval, we are glad to see this task is a little bit similar to the question answering task. We have the two ”sentences”: one is the query content, the other the document content, we need to use these two ”sentences” to train a classification model. Naturally, we can apply the BERT models to help us solve this problem. Before the two sentences, we need to add a [CLS] token, the corresponding output of this token contains the classification information, we can just connect it directly to a classification layer. Between the sentence of the query and document, we need to add a [SEP] token, to represent that this is the divider of these two sentences. Our work is based on the bert-base-uncased pre-trained BERT model. The two sentences need to do token embedding to transform into the valid format, then we can feed the embedding vector into the BERT models, to fine-tune for a document reranking model.

3.5 About Revived Runs

In this round of WWW English subtask, we submitted three revived runs, including the LambdaMART, Coordinate Ascent, and AdaRank learning-to-rank models. To generated these runs, we kept the same process and parameters as those of the WWW-2 runs (THUIR-E-CO-MAN-Base-2, THUIR-E-CO-MAN-Base-3, THUIR-E-CO-MAN-Base-1, respectively). However, there exist some differences between the revived ranking system and the original one. First, we converted the Python2 code to Python3, and reorganised the code to make the project more readable and compact. Second, because of the loss of some important data (server stored with the original project has been crashed.), we replaced the background corpus (used to calculate IDF, BM25, and some language models) with a new one.

3.6 Results

Table 2 shows the performance of our runs in the English subtask, including the mean metric values and the ranks among all 37 runs submitted in the English subtask. We can find that the learning-to-rank models perform better than BM25, as we expected. On the other hand, BERT's performance did not meet our expectations, that might because the training data sets we used are not large enough (only 84834 query-document pairs extracted from the MQ2007 and MQ2008 data sets) to fine-tune for the BERT's parameters. We adopted these revived ranking systems to generate the reranking results under WWW-3 topics, and concatenated the results with the ranking list of the original systems under WWW-2 topics. Our submitted runs are the concatenated versions.

4 CONCLUSION

In the NTCIR-15 WWW-3 task, we participated in both Chinese and English subtasks. We tried BERT models in both Chinese and English subtasks, we also revived some high performance runs in the WWW-2 task. In the future, we would like to investigate how to leverage the embedding of the BERT models into the learning-to-rank models, and how to better combine the human relevance labels with the implicit relevance feedback.

REFERENCES

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

[2] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2019. TianGong-ST: A New Dataset with Large-scale Refined Real-world Web Search Sessions. In *Proceedings of the 28th ACM International on Conference on Information and Knowledge Management.* ACM, 2485–2488.

[3] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[4] V Dang. 2012. The Lemur Project-Wiki-RankLib. Lemur Project.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.* 55–64.

[7] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management.* 2333–2338.

[8] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *arXiv preprint arXiv:2004.12832* (2020).

[9] Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.

[10] Cheng Luo, Tetsuya Sakai, Yiqun Liu, Zhicheng Dou, Chenyan Xiong, and Jingfang Xu. 2017. Overview of the NTCIR-13 We Want Web Task. In *Proceedings of NTCIR-13.* 394–401. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings13/pdf/ntcir/01-NTCIR13-OV-WWW-LuoC.pdf

[11] Jiaxin Mao, Tetsuya Sakai, Cheng Luo, Peng Xiao, Yiqun Liu, and Zhicheng Dou. 2019. Overview of the NTCIR-14 We Want Web Task. In *Proceedings of NTCIR-14.* 455–467. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings14/pdf/ntcir/01-NTCIR14-OV-WWW-MaoJ.pdf

[12] Donald Metzler and W Bruce Croft. 2007. Linear feature-based models for information retrieval. *Information Retrieval* 10, 3 (2007), 257–274.

[13] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[14] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424* (2019).

[15] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 datasets. *arXiv preprint arXiv:1306.2597* (2013).

[16] Stephen E Robertson. 1997. Overview of the okapi projects. *Journal of documentation* (1997).

[17] Tetsuya Sakai, Sijie Tao, Zhaohao Zeng, Yukun Zheng, Jiaxin Mao, Zhumin Chu, Yiqun Liu, Maria Maistro, Zhicheng Dou, Nicola Ferro, and Ian Soboroff. 2020. Overview of the NTCIR-15 We Want Web with CENTRE (WWW-3) Task. In *Proceedings of NTCIR-15*. to appear.

[18] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv* abs/1910.03771 (2019).

[19] Qiang Wu, Chris JC Burges, Krysta M Svore, and Jianfeng Gao. 2008. *Ranking, boosting, and model adaptation*. Technical Report. Technical report, Microsoft Research.

[20] Jun Xu and Hang Li. 2007. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 391–398.

[21] Chengxiang Zhai and John Lafferty. 2017. A study of smoothing methods for language models applied to ad hoc information retrieval. In *ACM SIGIR Forum*, Vol. 51. ACM, 268–276.

[22] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. An Analysis of BERT in Document Ranking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1941–1944.

[23] Yuyu Zhang, Ping Nie, Xiubo Geng, Arun Ramamurthy, Le Song, and Daxin Jiang. 2020. DC-BERT: Decoupling Question and Document for Efficient Contextual Encoding. *arXiv preprint arXiv:2002.12591* (2020).

[24] Yukun Zheng, Zhen Fan, Yiqun Liu, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. Sogou-QCL: A New Dataset with Click Relevance Label. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 1117–1120.