# THUIR at the NTCIR-16 WWW-4 Task

Shenghao Yang, Haitao Li, Zhumin Chu, Jingtao Zhan, 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 participates in the English subtask of the NTCIR-16 We Want Web with CENTRE (WWW-4) task. This paper elaborates on our methods and discusses the experimental results. We adopt three methods, namely learning-to-rank methods, a pre-trained language model tailored for information retrieval, and BERT with prompt learning. Experimental results demonstrate the importance of designing pre-training task specifically for information retrieval. Results also suggest the relatively simple prompt method cannot effectively improve the ranking performance.

## TEAM NAME

THUIR

## SUBTASKS

English

## 1 INTRODUCTION

Ranking is an important task for information retrieval. Given a user query, the retrieval system returns a list of documents from a large corpus to satisfy the user intent. Various retrieval methods are proposed, including bag-of-words models [15], learning-to-rank methods [7], and neural ranking models [10, 21, 22]. The We Want Web with CENTRE task (WWW) is an ad hoc search task organized by NII Testbeds and Community for Information access Research (NTCIR). It provides test topics and a large document corpus. Participants design ranking models to maximize a metric of interest.

We participate in the WWW task [16] this year. We adopt several re-ranking models: (1) list-wise learning-to-rank methods, i.e., LambdaMART [1] and Coordinate Ascent [19]. (2) a popular pre-trained language model tailored for information retrieval, namely PROP [10], which stands for pre-training with representative words prediction. (3) a BERT [2] model tuned with prompt learning to align pre-training with fine-tuning for better performance. We call it BERT-Prompt for short.

According to the experimental results, PROP achieves the best performance among our submitted models, demonstrating the importance of designing a pre-training task suitable for the ranking task. BERT-Prompt performs comparably to the learning-to-rank methods. We speculate that the relatively simple prompt method results in not optimal performance of BERT-Prompt. We will explore better prompt methods in the future.

[0]This work is supported by the Natural Science Foundation of China (Grant No. 61732008), Beijing Academy of Artificial Intelligence (BAAI) and Tsinghua University Guoqiang Research Institute.

## 2 ENGLISH TASK

In the WWW-4 English subtask, we submit two learning-to-rank runs (New runs), two PROP runs (New runs), and a BERT run (New run). This section will introduce our models in detail.

### 2.1 Learning-to-Rank methods

Learning-to-rank (LTR), as a series of machine learning algorithms for the re-ranking task, have been applied in many areas such as information retrieval, data mining, etc.

*2.1.1 Dataset.* We adopt MQ2007 and MQ2008 as the training data[14] and use the data provided in WWW1-3 [9, 11, 17] as the validation set in the English task. There are about 1700 topics with labeled documents in MQ2007 and about 800 in MQ2008. The corresponding documents are extracted from the GOV2 web page corpus [12] (about 25m pages). In addition, WWW1-3 datasets contain 260 topics with labeled document from the ClueWeb12-B13 dataset[1] (about 5 million pages). Each topic has approximately 213 relevance labeled documents on average. The relevance labeling is in 5-level setting ranging from 0 to 4 with increasing relevance. The test set of WWW-4 contains 50 topics and 100 candidate documents per topic retrieved by BM25.

*2.1.2 Feature Extraction.* Features are very important for learning-to-rank methods. In order to better extract features, we use various methods to preprocess HTML files. Specifically, we use bs4 package[2] to parse HTML documents, and then ignore <script> and <style> tags. We use some natural language processing methods to make HTML more standardized, including lowercasing, tokenization, removing stop words, and stemming. Finally, four fields of the HTML document are obtained: the whole html content, the uniform resource locator (URL) of this html, the anchor texts, and the title. We use the same preprocessing process for the query content to make them in the same.

We extract the following eight features in four fields: term frequency (TF), inverse document frequency (IDF), TF * IDF, document length (DL), BM25, LMIR.ABS, LMIR.DIR and LMIR.JM. In this way, $4 \times 8 = 32$ features has been extracted.

The IDF calculation formula is shown in Eq 1, where $D$ represents the total number of documents in the corpus and $D_i$ represents the number of documents containing the word $t_i$. The calculation formula of BM25 is shown in Eq 2. We set the parameters as $k_1 = 1.2, k_2 = 100, b = 0.75$. The calculation formula of LMIR is shown in Eq 3, and the details can be referred to Zhai et al.'s work[20].

$$IDF(t_i) = \log \frac{|D|}{|D_i + 1|} \tag{1}$$

[1]https://lemurproject.org/clueweb12/
[2]https://beautifulsoup.readthedocs.io/zh_CN/v4.4.0/

**Table 1: WWW-4 official results of THUIR runs based on the gold relevance assessments.**

| Run | Model | Mean nDCG | | Mean Q | | Mean nERR | | Mean iRBU | |
|---|---|---|---|---|---|---|---|---|---|
| THUIR-E-CO-NEW-1 | PROP | 0.3596 | 5 | 0.2931 | 2 | 0.5102 | 4 | 0.7449 | 7 |
| THUIR-E-CO-NEW-2 | PROP | 0.3670 | 3 | 0.2944 | 1 | 0.5289 | 1 | 0.7544 | 5 |
| THUIR-E-CO-NEW-3 | BERT-Prompt | 0.3222 | 13 | 0.2494 | 14 | 0.4281 | 18 | 0.7166 | 16 |
| THUIR-E-CO-NEW-4 | LambdaMart | 0.3094 | 16 | 0.2288 | 16 | 0.4672 | 13 | 0.7510 | 6 |
| THUIR-E-CO-NEW-5 | Coordinate Ascent | 0.3405 | 6 | 0.2667 | 8 | 0.4783 | 9 | 0.7545 | 4 |

$$BM25(d,q) = \sum_{i=1}^{M} \frac{IDF(t_i) \cdot TF(t_i,d) \cdot (k_1+1)}{TF(t_i,d) + k_1 \cdot \left(1 - b + b \cdot \frac{len(d)}{avgdl}\right)} \tag{2}$$

$$\log p(q|d) = \sum_{i:c(q_i;d)>0} \log \frac{p_s(q_i|d)}{\alpha_d p(q_i|C)} + n \log \alpha_d + \sum_i \log p(q_i|C) \tag{3}$$

*2.1.3 Training.* We use Ranklib package[3] to implement learning-to-rank algorithm. We treat the format of the training data (also testing/validation data) as the same as that of Letor datasets [14] and use the default parameter in Ranklib to train. We choose Lambdamart and Coordinate Ascent as the run4 and run5 in final submission because these models perform well in the validation set.

## 2.2 PROP

Recently, pre-trained language models, e.g., BERT, have boosted the performance of ad hoc search [13, 21, 22]. PROP is a pre-trained language model tailored for information retrieval. It adds the **R**epresentative w**O**rds **P**rediction (ROP) task in the pre-training task, which is based on the assumption that the query should be a representative set of words in the document. Specifically, a pair of word sets is sampled from the document as a pseudo query, and the sampling strategy is based on multinomial unigram language model with Dirichlet prior smoothing. The pseudo query with higher query likelihood is considered as the more "representative" query of the document, and the preference for pseudo query pairs is trained using pairwise approach in pre-training. PROP uses the same architecture as BERT and incorporates the Masked Language Model (MLM) objective besides the ROP objective in pre-training.

Following Nogueira and Cho [13], we concatenate the query and document as input of PROP. The output embedding of [CLS] token is fed into a linear layer to obtain the relevance scores of the query-document pairs. The scores are used to re-rank the candidate documents. We use transformers[4][18] library to implement the PROP model.

*2.2.1 Dataset.* We train PROP on the dataset collected from WWW 1-3. For all 260 topics, we divide the training set and validation set in the ratio of 4:1. The train set contains 208 topics. In order to perform pairwise training, we convert the labeled documents of each topic into <topic,doc1,doc2> tuples , and finally we get approximately 6M tuples. The models are trained to predict a higher score for the more relevant document in a tuple. For the validation set, since the labeled documents are obtained after the first stage of retrieval, we directly re-rank the labeled documents for each query.

*2.2.2 Training.* Here we describe two methods for selecting the model checkpoints. The first method selects the best performing checkpoint on a validation set. Specifically, we use nDCG@10 to evaluate the re-ranking performance on the validation set. We tune the hyper-parameters based on the validation performance and choose the best checkpoint. Our submitted run1 corresponds to this checkpoint. The second method inherits the tuned hyper-parameters of run1 but uses the full labeled data for training without validation. In this way, we can utilize more training data compared with run1. This model corresponds to run2.

## 2.3 BERT-Prompt

Prompt learning is a new approach to apply pre-trained language models to downstream tasks. It has improved performance in many downstream tasks [6]. Its core idea is to convert the downstream task into a form like pre-training task to maximize the performance of the pre-trained model. We use *cloze prompt* [6] for re-ranking task based on a Masked Language Model, i.e., BERT. Specifically, we feed BERT in "[query] [mask] [document]" format and predict the probability of all words in the vocabulary at the [mask]. This custom input format is called *template* [3] in prompt learning. We utilize "yes", "and", and "so" as the positive words. "but", "yet", and "however" are the negative words. This process of aligning words with labels is called *verbalizer* [3] in prompt learning. The relevance score equals the subtraction results between the average probability values of positive words and negative words.

In fact, some novel work proposes not to use discrete words as template and verbalizer but to use continuous embedding [4, 5, 8]. These approaches can overcome the instability of designing discrete words manually and can use continuous embedding as learnable parameters in training. Since we just want to test the effectiveness of prompt learning on the re-ranking task, the template and verbalizer we designed are relatively simple.

The training process of BERT-Prompt is consistent with run1. The new ranking list of test set obtained by BERT-Prompt is run3 in our final submission. We use Openprompt[5][3] and transformers library to implement BERT-Prompt model.

## 2.4 Experimental Results

WWW-4 collects gold relevance assessments given by topic creators. Table 1 shows the metric scores and rank positions of our five submitted models. PROP achieves the best overall performance on four evaluation metrics. Therefore, it is important to utilize a

[3]https://github.com/codelibs/ranklib
[4]https://github.com/huggingface/transformers
[5]https://github.com/thunlp/OpenPrompt

**Table 2: Case study on topic 217 and document 2bd06c76-f5f3-4be1-98c8-0d66dbdf41a6. In this case, the high idf topic term "inventor" appears less frequently. It results in a poor ranking performance of learning-to-rank method. While neural ranking model can capture semantic relevance rather than lexical matching and performs better. Specifically, the relevant document in this case is ranked by PROP at the first position, while Coordinate Ascent ranks it at the 77th position.**

| Topic: inventor of the Web<br>Description: Who is the inventor of the World Wide Web?<br>Rank result: PROP(1) Coordinate Ascent(77) Gold relevance judgment(L2) |
|---|
| https://www.famous**inventor**s.org/tim-berners-lee tim berners-lee \| biography, inventions and facts famous **inventor**shome about blog contacttim berners-lee tim berners-lee invented “world wide **web**” and “html”. tim berners-lee (formally sir timothy john berners-lee) is the **inventor** of the world wide **web**. he was born in britain on june 8th, 1955 and graduated from oxford university with a first class honors degree in physics. he was influenced by his parents’ interest in computers and technology, as they were part of the team who built the first commercial computer. as a child he was deeply interested in trains and took them apart to learn how they worked. at college he built his first computer using an old television, a soldering iron and a processor. |

**Table 3: Case study on topic 225 and document 2250aaa2-6a41-40a4-8924-72c782129ec9. In this case, the topic terms appears at later positions in the document and the topic terms "signifier" and "saussure" even not appears in following table. Limited to the input length, PROP can’t obtain enough relevant information and performs poorly. While, learning-to-rank method is not limited by the input length and performs better. Specifically, the relevant document in this case is ranked by PROP at the 53th position, while Coordinate Ascent ranks it at the 4th position.**

| Topic: signifier saussure theory<br>Description: You want to know the meaning of term "signifier" in linguist Saussure’s theory<br>Rank result: PROP(53) Coordinate Ascent(4) Gold relevance judgment(L2) |
|---|
| https://www2.slideshare.net/mattheworegan/stuart-hall-representation-**theory** stuart hall - representation **theory** slideshare uses cookies to improve functionality and performance, and to provide you with relevant advertising. if you continue browsing the site, you agree to the use of cookies on this website. see our user agreement and privacy policy. slideshare uses cookies to improve functionality and performance, and to provide you with relevant advertising. if you continue browsing the site, you agree to the use of cookies on this website. see our privacy policy and user agreement for details.slideshareexploresearchyouupload login signupsubmit search home explore successfully reported this slideshow. we use your linkedin profile and activity data to personalize ads and to show you more relevant ads. |

**Table 4: Case study on topics where both PROP and Coordinate Ascent performs well or poorly.**

| topic ids | topic | Mean nDCG | | Mean Q | |
|---|---|---|---|---|---|
| | | PROP | CA | PROP | CA |
| 201 | Timnit Gebru Google | 0.8002 | 0.8002 | 0.8435 | 0.7488 |
| 245 | chicken breast recipes | 0.8880 | 0.9450 | 0.9075 | 0.9524 |
| 250 | trek emonda price | 0.9653 | 0.9351 | 0.9888 | 0.9896 |
| 203 | idf inventor | 0 | 0 | 0 | 0 |
| 206 | DC and Marvel characters | 0 | 0 | 0 | 0 |
| 207 | dirty loops bassist | 0 | 0 | 0 | 0 |
| 220 | half life | 0 | 0.0367 | 0 | 0.0095 |

pre-training task that is similar to the ranking task. Note that run2 performs slightly better than run1. We believe the main reason is that run2 uses more topics in training. But the performance of run2 does not improve significantly limited by the total number of topics. BERT-Prompt underperforms PROP and performs comparably with the learning-to-rank methods. Therefore, prompt learning does not substantially improve the ranking performance as expected. We speculate that the used prompt learning approach is relatively simple. The input format we design is placing the "[mask]" token in the middle of query and document, and we specify the word of the corresponding labels. These designs are manual and may not be the potentially best prompt learning approach. There are many novel approaches to prompt learning recently, but we have only adopted a simple approach in this task. We believe that with a more sophisticated design and more advanced methods, the performance can be further improved.

## 2.5 Case Study

In this section, we will investigate the performance of neural ranking models and learning-to-rank methods by presenting some cases. Specifically, we choose PROP (THUIR-E-CO-NEW-2) and Coordinate Ascent (THUIR-E-CO-NEW-5) as neural ranking model and learning-to-rank method for the comparison. We will investigate the ranking results of a relevant document in a specific topic. Due to space limitation, we only present approximately first 100 words of the relevant document. The topic terms are bolded display in the following tables.

Intuitively, the neural ranking models can capture the semantic similarity between topic and document, while the learning-to-rank methods focus on lexical features and may encounter the "semantic gaps" problem. In the case shown in Table 2, for the relevant document, the neural ranking model can focus on the information related to the founders of the World Wide Web, not only on the high IDF word "inventor". We find that the word “inventor” appears nine times in the document. The term frequency of this word is not high compared to other documents that are ranked higher by the learning-to-rank method. It leads to a lower ranking for this relevant document in learning-to-rank methods. Although the neural ranking model has better semantic matching ability, it is limited by the input length (512) of transformer model, resulting in poor performance on long document, such as the document of the case in Table 3. The topic term appears later in the document, even in the approximately first 100 words we present, the two topic terms "signifier" and "saussure" do not appear. But the learning-to-rank methods do not suffer from this limitation and performs better.

In addition, We note that the two models perform differently under different topics. As shown in Table 4, both models achieve good performance under the first few topics, but perform poorly in the later ones. Through the observation we find that the latter several topics generally present a vague search intention or the topic terms are ambiguous. It results in the bad ranking results. In comparison, the first few topics are more complete in their formulation and contain some distinct terms. It facilitates the models to determine the relevant documents more accurately.

## 3 CONCLUSION

In NTCIR-16 WWW-4 task, we participate in the English subtask. We investigate PROP, BERT based neural ranking models, and learning-to-rank methods. Experimental results show the importance of pre-training. We also observe that the BERT-Prompt method does not achieve the expected improvement. In the future, we will try to further optimize our prompt learning design approach to further improve the ranking performance of BERT-Prompt.

## REFERENCES


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

[2] 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).

[3] Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998* (2021).

[4] Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. Warp: Word-level adversarial reprogramming. *arXiv preprint arXiv:2101.00121* (2021).

[5] Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259* (2021).

[6] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586* (2021).

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

[8] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *arXiv preprint arXiv:2103.10385* (2021).

[9] Cheng Luo, Tetsuya Sakai, Yiqun Liu, Zhicheng Dou, Chenyan Xiong, and Jingfang Xu. 2017. Overview of the NTCIR-13 we want web task. *Proc. NTCIR-13* (2017).

[10] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021. Prop: Pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 283–291.

[11] Jiaxin Mao, Tetsuya Sakai, Cheng Luo, Peng Xiao, Yiqun Liu, and Zhicheng Dou. 2019. Overview of the NTCIR-14 we want web task. *Proceedings of NTCIR-14* (2019), 455–467.

[12] Kevin S McCurley. 2008. Observations about the GOV2 TREC data set.

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

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

[15] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[16] Tetsuya Sakai, Sijie Tao, Zhumin Chu, Maria Maistro, Yujing Li, Nuo Chen, Nicola Ferro, Junjie Wang, Ian Soboroff, and Yiqun Liu. 2022. Overview of the NTCIR-16 We Want Web with CENTRE (WWW-4) Task. In *Proceedings of NTCIR-16*. to appear.

[17] Tetsuya Sakai, Sijie Tao, Zhaohao Zeng, Yukun Zheng, Jiaxin Mao, Zhumin Chu, Yiqun Liu, Maria Maistro, Zhicheng Dou, Nicola Ferro, et al. 2020. Overview of the NTCIR-15 we want web with CENTRE (WWW-3) task. *Proceedings of NTCIR-15. to appear* (2020).

[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] Stephen J Wright. 2015. Coordinate descent algorithms. *Mathematical Programming* 151, 1 (2015), 3–34.

[20] 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 New York, NY, USA, 268–276.

[21] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1503–1512.

[22] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2022. Learning Discrete Representations via Constrained Clustering for Effective and Efficient Dense Retrieval. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) *(WSDM '22)*. Association for Computing Machinery, 1328–1336. https://doi.org/10.1145/3488560.3498443