

Towards Better Web Search Performance: Pre-training, Fine-tuning and Learning to Rank

Haitao Li
DCST, Tsinghua University
Zhongguancun Laboratory
Beijing 100084, China
liht22@mails.tsinghua.edu.cn

Jia Chen
DCST, Tsinghua University
Zhongguancun Laboratory
Beijing 100084, China
chenjia0831@gmail.com

Weihsang Su
DCST, Tsinghua University
Zhongguancun Laboratory
Beijing 100084, China
swh22@mails.tsinghua.edu.cn

Qingyao Ai
DCST, Tsinghua University
Zhongguancun Laboratory
Beijing 100084, China
aiqy@tsinghua.edu.cn

Yiqun Liu*
DCST, Tsinghua University
Zhongguancun Laboratory
Beijing 100084, China
yiqunliu@tsinghua.edu.cn

ABSTRACT

This paper describes the approach of the THUIR team at the WSDM Cup 2023 Pre-training for Web Search task. This task requires the participant to rank the relevant documents for each query. We propose a new data pre-processing method and conduct pre-training and fine-tuning with the processed data. Moreover, we extract statistical, axiomatic, and semantic features to enhance the ranking performance. After the feature extraction, diverse learning-to-rank models are employed to merge those features. The experimental results show the superiority of our proposal. We finally achieve second place in this competition.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**; *Information retrieval*.

KEYWORDS

WSDM Cup 2023, Pre-training, Web Search, Learning to rank

ACM Reference Format:

Haitao Li, Jia Chen, Weihsang Su, Qingyao Ai, and Yiqun Liu. 2023. Towards Better Web Search Performance: Pre-training, Fine-tuning and Learning to Rank. In *Proceedings of The 16th International ACM WSDM Conference on Research and Development in Information Retrieval (WSDM'23)*. ACM, New York, NY, USA, 4 pages.

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WSDM'23, from February 27 to March 3, 2023,
© 2023 Association for Computing Machinery.

1 INTRODUCTION

In recent years, pre-trained language models (PLMs) have been proved effective in capturing rich language knowledge from large-scale corpus, reaching state-of-the-art performance in many downstream natural language processing (NLP) tasks [3, 7, 13, 16, 17]. These pre-trained language models are first trained on a large-scale corpus without supervision and then fine-tuned on labeled data for downstream tasks. An ideal PLM for Information Retrieval (IR) should have the ability to measure the relevance between a query and the candidate document. However, the pre-training tasks in the NLP field such as Masked Language Modeling (MLM) and Next Sentence Prediction (NSP), cannot directly enhance the relevance-matching ability of PLMs. Thus, researchers in the IR field have begun to explore the pre-training methods tailored for Web Search [1, 2, 4, 8, 9].

Thanks to the Pre-training for Web Search challenge in WSDM Cup 2023, a real-world long-tail user feedback dataset from Baidu Search is now available for exploring better pre-training methods for web search. The competition's goal is to achieve better ranking performance by leveraging advanced pre-training and fine-tuning methodologies.

In this paper, we introduce the solution of THUIR team, which achieves second place in the competition (DCG@10 10.04097 on the leaderboard). To be specific, we first conduct an exploratory analysis of the pre-training corpus and filter out a new corpus. Then, we pre-train a 12-layer transformer with MLM and CTR prediction tasks and fine-tune it with the expert annotated data. Finally, we extracted 20 features for each query-document pair. Diverse learning-to-rank methods are employed to aggregate these features into the final relevance score.

This paper is organized as follows: Section 2 introduces the pre-training and fine-tuning dataset. In Section 3, details of our method is elaborated. Then, the experimental setting and results are introduced in Section 4. Finally, we conclude our work and discuss the future direction in Section 5.

2 DATASET

In this section, we focus on the datasets provided by the competition.

2.1 Large Scale Web Search Session Data

The large-scale web search session data is real long-tail user feedback data provided by Baidu [19]. This data contains 2000 files in .gz format and each file is about 500M. This session data only provides the pre-processed tokens of the query and document.

To construct our pre-trained corpus, we first encode queries in both session data and expert annotation data into vectors with the pre-trained language model. Then, the inner product is employed to calculate the similarity of the two types of queries. If a query on session data has a similarity of less than 0.9 with all the queries annotated by experts, then we will filter it out from the session data. Moreover, we filtered out queries that did not click on any documents. We think these queries lack positive samples and cannot provide sufficient training information for the pre-trained language models.

2.2 Expert Annotation Dataset

Expert annotated data contains 7,008 queries and 397,552 query-document pairs. Since the official data has the problem of different queries with the same id, we first mix the original training set and the validation set and then remap the QIDS for all queries and ensure one query only have one unique identifier. After that, we encode all queries into vectors with the pre-trained language model. We calculate the similarity between the above queries and testing queries and select 20% most similar queries as the new validation set.

Finally, there are 298,537 training instances and 99,015 validation instances, which are named `train_new` and `test_new` respectively. Furthermore, we count the occurrence of all tokens and select the token with the top 50 occurrence frequency as the stop words.

3 METHOD

In this section, we present the complete solution of the Pre-training for Web Search task.

3.1 Pre-training

As pre-trained language models (PLMs) have shown great effectiveness in IR tasks, we adopt PLMs as our backbone model. To be specific, we used the official pre-training code released by Baidu to train a backbone model. Two tasks are employed in our method: MLM and CTR tasks. The loss function of these two tasks is as follows:

$$\mathcal{L}_{MLM} = - \sum_{\hat{x} \in m(x)} \log p(\hat{x} | x_{\setminus m(x)}) \quad (1)$$

$$\mathcal{L}_{CTR} = - \sum_{c \in C} c \log(s) + (1 - c) \log(1 - s) \quad (2)$$

where x , $m(x)$, and $x_{\setminus m(x)}$ denote the input sequence, the masked and the rest word sets in x , respectively. Besides, C is the complete click feedback set, c is the click signal on a result, and s is the estimated click probability of this result. We set the mask rate of the MLM task as 40% and stop the pre-training process if the total loss does not decline for 1k steps.

Table 1: Final features to train learning to rank models. BM25_title and BM25_content are only considered for the title and content of the document respectively. PROX-nonstop denotes the model results considering the stop words.

Feature ID	Feature Name	Feature ID	Feature Name
1	cross encoder	11	TF
2	query_length	12	IDF
3	title_length	13	PROX-1
4	content_length	14	PROX-2
5	query_freq	15	PROX-3
6	BM25	16	PROX-4
7	QLD	17	PROX-1-nonstop
8	BM25_title	18	PROX-2-nonstop
9	BM25_content	19	PROX-3-nonstop
10	TF-IDF	20	PROX-4-nonstop

3.2 Fine-tuning

For each query, the candidate documents are labeled with a relevance score $\in \{0, 1, 2, 3, 4\}$. We define the candidate documents with labels 3 and 4 as positive document d^+ and the candidate documents with labels 0 and 1 as d^- .

We use the Transformer encoder with a similar structure with BERT as the ranking model. The score of a query document pair is defined as follows,

$$Input = [CLS]query[SEP]title[SEP]content[SEP] \quad (3)$$

$$score(query, doc) = MLP(CLS[BERT(Input)]) \quad (4)$$

where CLS is the [CLS] token vector and MLP is a Multilayer Perceptron that projects the CLS vector to a score. We use the Cross Entropy Loss to optimize the re-ranking and retrieval model, which is defined as:

$$Loss = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}) \quad (5)$$

where y is the ground truth and \hat{y} is the predicted probability.

3.3 Learning to Rank

Learning-to-rank, a popular machine learning method for ranking, has been widely used in information retrieval and data mining [12, 15]. In this section, we implement several learning-to-rank methods with diverse features.

3.3.1 Dataset. We adopt `train_new` as the training data and the `valid_new` as the validation set. There are 298,537 training pairs and 99,015 validation pairs and there is no overlap between them. The relevance labeling is in a 5-level setting ranging from 0 to 4 with increasing relevance.

3.3.2 Feature Extraction. Feature engineering plays an essential role in learning to rank. Except for taking the scores of the above pre-trained language models as features, we also construct statistical features, axiomatic features [2], and semantic features to achieve better ranking performance. All the extracted features are shown as follows:

- **Statistical Features**

- **query_length** is the length of the query token.
- **title_length** is the length of the document title token.
- **content_length** is the length of the document content token.
- **query_freq** is the monthly search frequency of query. The queries are descendingly split into 10 buckets according to their monthly search frequency.
- **TF-IDF** [10] is a classical retrieval model based on bag-of-words. TF-IDF is the combination of term frequency (TF) and inverse document frequency (IDF). Their equations are shown as follows:

$$TF(t_{i,j}) = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (6)$$

$$IDF(t_i) = \log \frac{|D|}{|D_i + 1|} \quad (7)$$

$$TF - IDF = TF \times IDF \quad (8)$$

where $n_{i,j}$ denotes the number of words t_i in the document d_j , D represents the total number of documents in the corpus and D_i represents the number of documents containing the word t_i . We extracted the TF, IDF, TF-IDF of the title, content, and title + content respectively.

- **BM25** [11] is a highly effective sparse retriever based on exact word matching. The calculation formula of BM25 is shown in Eq 9.

$$BM25(d, q) = \sum_{i=1}^M \frac{IDF(t_i) \cdot TF(t_i, d) \cdot (k_1 + 1)}{TF(t_i, d) + k_1 \cdot \left(1 - b + b \cdot \frac{\text{len}(d)}{\text{avgdl}}\right)} \quad (9)$$

where k_1, b are hyperparameters. Grid search is employed to determine the optimal hyperparameters i.e. $k_1 = 1.6$ and $b = 0.87$. Similar to TF-IDF, we extracted three scores of BM25: title, content and title+content.

- **QLD** [18] is another representative traditional retrieval model based on Dirichlet smoothing. The equation for QLD is shown in Eq 10. We only extracted the QLD score of title+content.

$$\log p(q|d) = \sum_{i:c(q_i:d)>0} \log \frac{p_s(q_i|d)}{\alpha_d p(q_i|C)} + n \log \alpha_d + \sum_i \log p(q_i|C) \quad (10)$$

• Axiomatic Features

- **PROX_1** [6]: Averaged proximity score of query terms in the candidate document.
- **PROX_2** [6]: First appearance position of query terms appearing in the candidate document.
- **PROX_3** [6]: Number of query term pairs appearing in the candidate document within a distance of 5.
- **PROX_4** [6]: Number of query term pairs appearing in the candidate document within a distance of 10.
- **STM_1** [5]: The STM-1 axiom prefers to select queries that have a high semantic similarity to the document.
- **STM_2** [5]: The STM-2 axiom is used to distinguish two queries if they have a close similarity to the same document.
- **REG** [14]: The REG axiom tends to select those queries where the most diverse terms appear more often in the document.

• Semantic Features:

Table 2: Overall experimental results of different methods. The best compression method in each column is marked in bold.

Model	Vaild Set	Leaderboard
BM25(k1=1.6,b=0.86)	10.30	9.51
TF	9.14	-
IDF	8.92	-
TF-IDF	9.20	-
QLD	9.84	-
Cross encoder	10.06	8.58
Coordinate Ascent	11.04	-
LambdaMART	11.11	-
XGBoost	11.18	10.00
LightGBM	11.25	10.04

- **Dense Embedding**: The Transformer encoders (e.g., BERT, RoBERTa) output a contextualized vector for each input token and we select the CLS vector as the semantic embedding of a document. Then, we use dot product to get the score of the query and document as the feature

3.3.3 Modeling. We employ four methods of learning to rank: Lightgbm, XGBoost, LambdaMART and Coordinate Ascent. For LambdaMART and Coordinate Ascent, Ranklib¹ package is applied to implement them. For Lightgbm and XGBoost, we reproduce them with their official packages.

4 EXPERIMENT

4.1 Feature Selection

We generate tens of features, and the learning ranking models are trained based on these features. However, there are complex homogenization relationships between features and many features are distributed inconsistently between the training and test sets. To make the model more effective, we further perform feature analysis and selection. To be specific, we perform heuristic feature selection based on ablation experiments. With fixed model hyperparameters and random seeds, we eliminate one or a set of similar features at a time to train the learning-to-rank model. After feature selection, 20 features are reserved to train the final model. The final features are displayed in Table 1.

4.2 Experimental Results

The performance comparisons of different methods are shown in Table 2. DCG@10 is employed to evaluate the performance of ranking systems. We derive the following observations from the experiment results.

- Traditional lexical matching models such as BM25, QL, and TF-IDF show promising performance. It can be observed that BM25, which has been carefully tuned, achieves a score of 9.51 on the leaderboard, which exceeds most teams in the competition.
- Despite the extensive training data, the performance of the cross encoder is not satisfactory. After careful examination, we find

¹<https://github.com/codelibs/ranklib>

that the official paddle code may exist bugs. However, there is no time to correct this mistake for our team. Even so, the score of the cross encoder still improves the final learning to rank model, which indicates that the cross encoder provides the information on different aspects of the dataset.

- Compared to a single model, learning to rank models achieve better performance, which indicates that fusing different information from different combinations of datasets is beneficial for ranking. Consequently, LightGBM achieves the highest score and yields our final submission.

5 CONCLUSION

This paper describes our solution for WSDM Cup 2023 Pre-training for Web Search task. We extract statistical, axiomatic, and semantic features to enhance the ranking performance. We finally achieve second place in this competition. In the future, we will explore more pre-training methods for web search based on search engine logs.

REFERENCES

- [1] Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932* (2020).
- [2] Jia Chen, Yiqun Liu, Yan Fang, Jiaxin Mao, Hui Fang, Shenghao Yang, Xiaohui Xie, Min Zhang, and Shaoping Ma. 2022. Axiomatically Regularized Pre-training for Ad hoc Search. (2022).
- [3] 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).
- [4] Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, and Yiqun Liu. 2021. Pre-training Methods in Information Retrieval. *arXiv preprint arXiv:2111.13853* (2021).
- [5] Hui Fang and ChengXiang Zhai. 2006. Semantic term matching in axiomatic approaches to information retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 115–122.
- [6] Matthias Hagen, Michael Völske, Steve Göring, and Benno Stein. 2016. Axiomatic result re-ranking. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 721–730.
- [7] Yinhan Liu, Mylène Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
- [8] 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.
- [9] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Yingyan Li, and Xueqi Cheng. 2021. B-PROP: bootstrapped pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1513–1522.
- [10] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, Vol. 242. Citeseer, 29–48.
- [11] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.
- [12] Weihang Su, Xiangsheng Li, Yiqun Liu, Min Zhang, and Shaoping Ma. [n.d.]. THUIR2 at NTCIR-16 Session Search (SS) Task. ([n.d.]).
- [13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [14] Hao Wu and Hui Fang. 2012. Relation based term weighting regularization. In *Advances in Information Retrieval: 34th European Conference on IR Research, ECIR 2012, Barcelona, Spain, April 1-5, 2012. Proceedings 34*. Springer, 109–120.
- [15] Shenghao Yang, Haitao Li, Zhumin Chu, Jingtao Zhan, Yiqun Liu, Min Zhang, and Shaoping Ma. 2022. THUIR at the NTCIR-16 WWW-4 Task. *Proceedings of NTCIR-16. to appear* (2022).
- [16] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems* 32 (2019).
- [17] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining Language Models with Document Links. *arXiv preprint arXiv:2203.15827* (2022).
- [18] ChengXiang Zhai. 2008. Statistical language models for information retrieval. *Synthesis lectures on human language technologies* 1, 1 (2008), 1–141.
- [19] Lixin Zou, Haitao Mao, Xiaokai Chu, Jiliang Tang, Wenwen Ye, Shuaiqiang Wang, and Dawei Yin. 2022. A large scale search dataset for unbiased learning to rank. *arXiv preprint arXiv:2207.03051* (2022).