NTCIR 17 Conference: Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, December 12-15, 2023, Tokyo, Japan
DOI: https://doi.org/10.20736/0002001320

Overview of the NTCIR-17 Unbiased Learning to Rank Evaluation 2 (ULTRE-2) Task

Zechun Niu, Jiaxin Mao
Renmin University of China
P.R.C.
maojiaxin@gmail.com

Qingyao Ai
Tsinghua University
P.R.C.
aiqy@tsinghua.edu.cn

Lixin Zou
Wuhan University
P.R.C.
zoulixin15@gmail.com

Shuaiqiang Wang, Dawei Yin
Baidu Inc
P.R.C.
shqiang.wang@gmail.com,yindawei@acm.org

ABSTRACT

In this paper, we present an overview of the Unbiased Learning to Rank Evaluation 2 (ULTRE-2) task, a pilot task at the NTCIR-17. The ULTRE-2 task aims to evaluate the effectiveness of unbiased learning to rank (ULTR) models with a large-scale user behavior log collected from Baidu.com, a commercial Web search engine. In this paper, we describe the task specification, dataset construction, implemented baselines, and official evaluation results of the submitted runs.

KEYWORDS

Unbiased Learning to Rank, Evaluation, Web Search, Real-world behavior log

1 INTRODUCTION

Unbiased learning to rank (ULTR) [1, 4, 11, 12, 16, 18, 19] aims to train an unbiased ranking model with biased user behavior data. It has become a popular topic in the IR community as researchers have proposed many ULTR models, most of which are based on Inverse Propensity Score (IPS) [14], to mitigate multiple biases (e.g., position bias [10], trust bias [2], and selection bias [13]) in user behavior data. Theoretically, in the ideal case where the assumptions on user behavior are correct and the propensity estimation is accurate, it can be proved that the IPS-based models are unbiased. Empirically, due to the difficulties in collecting and sharing large-scale behavior logs in online systems, the evaluation of ULTR models mainly relies on simulation-based experiments with synthetic click data.

However, the mainstream simulation method is rather simple and the synthetic data may not match the complex real-world scenarios. Most simulation-based experiments only use a single user behavior model (usually PBM [5]) to simulate clicks, which may not fully capture the diverse user behavior patterns in the real world. Moreover, the propensity parameters and noise level used to generate the synthetic data are often hand-crafted, which may differ substantially from those in real click logs. As a result, although many ULTR models have achieved promising results on synthetic data, they still lack guarantees of effectiveness in real-world scenarios [21].

To make up for the above shortcomings, we launched a pilot task named Unbiased Learning to Rank Evaluation (ULTRE) [20] in the NTCIR-16. In the ULTRE task, we utilized multiple click models calibrated with real click logs to simulate various user behavior patterns. However, the click models used for generating synthetic clicks in ULTRE may still fail to describe the complex behaviors of real users, and the dataset constructed by ULTRE is relatively small. Therefore, we further propose the Unbiased Learning to Rank Evaluation 2 (ULTRE-2) task in the NTCIR-17. In the ULTRE-2 task, we evaluate the effectiveness of ULTR models with a new, large-scale user behavior log collected from a commercial Web search engine, Baidu.com. In addition to the real click log, we also provide rich display information (e.g., displayed height and displayed abstract) and other user behavior information (e.g., dwelling time and slip count), enabling the development of more advanced ULTR models. Besides, we preprocess the training data and provide a rich feature set, including both the traditional Learning-to-rank features and dense representations output by a BERT-based ranking model, so that the participants can easily train ULTR models without being limited by GPU resources.

The remainder of this paper is organized as follows: Section 2 describes the task specification and evaluation metric of the ULTRE-2 task. Section 3 details the dataset construction methodology. Section 4 lists the submitted runs from the participants and organizers. Section 5 reports and analyzes the official evaluation results for all the runs. Finally, Section 6 gives a brief conclusion of the ULTRE-2 task.

Table 1: Schedule of ULTRE-2 at NTCIR-17.

Time	Content
May 1, 2023	Dataset released
July 1, 2023	Registration due
Aug 1, 2023	Run submissions due
Aug 15, 2023	Final evaluation result released
Aug 15, 2023	Draft of task overview paper released
Sept 15, 2023	Participant paper submissions due
Nov 1, 2023	Camera-ready paper submissions due
Dec 2023	NTCIR-17 Conference in NII, Tokyo, Japan

Table 2: Statistics of the ULTRE-2 dataset.

	Training	Validation	Test
Unique queries	34,047	5,201	5,201
Sessions	1,052,295	5,201	5,201
Label	clicked or not (1) or (0)	relevance annotations (0-4)	relevance annotations (0-4)
Information	other behavior information and rich display information	No	No
Text	sequential token ids of original query, title, and abstract.		
Feature	pretrained and traditional features of 782 dimensions		

2 TASK DESCRIPTION

The Unbiased Learning to Rank Evaluation 2 (ULTRE-2) task is a pilot task in NTCIR-17, which concentrates on evaluating the effectiveness of ULTR models in a real-world Web search scenario.

2.1 Task Specification

In ULTRE-2, we construct and release a dataset based on the real user behavior logs from a Chinese Web search engine, Baidu, please see Section 3 for more details. With the provided initial ranking lists and query-document features, as well as rich user behavior data (e.g., click, dwelling time and slip count) and display information (e.g., displayed height and displayed abstract), participants are supposed to train a feature-based ranking model on the training set and use it to re-rank the ranking lists of the test queries.

Specifically, we encourage the participants to leverage the abundant types of user behavior data and the rich display information to develop more sophisticated ULTR models. For example, they can utilize the dwelling time data together with clicks to better understand users' preferences on the search results. They can also utilize the display information such as displayed height and multimedia type to obtain more accurate propensity estimations.

The schedule of ULTRE-2 is shown in Table 1.

2.2 Evaluation

In ULTRE-2, we use the nDCG@10 [9] metric based on 5-level (0-4) human relevance labels to evaluate the performance of the submitted runs. For a ranked list π of N documents, we use the following implementation of DCG@N:

$$DCG@N = \sum_{i=1}^{N} \frac{2^{rel_i} - 1}{\log_2(\mathrm{i}+1)}$$

$$nDCG@N = \frac{DCG@N_\pi}{DCG@N_{ideal}}$$

where rel_i is the relevance label of the i-th document, and $DCG@N_{ideal}$ means the DCG@N value of the ideal ranked list that sorts the documents by their relevance labels from largest to smallest.

3 DATASET CONSTRUCTION

The dataset in the ULTRE-2 task is constructed based on the Baidu-ULTR dataset [1], which is a public unbiased learning to rank dataset collected from the mobile Web search engine of Baidu. To facilitate the training of ULTR models by the participants, we preprocessed a part of the training sessions of Baidu-ULTR and extracted 782-dimension pretrained and traditional features. Table 2 shows the details of the ULTRE-2 dataset. The ULTRE-2 dataset can be downloaded at the Google Drive link [2].

3.1 Query Preprocessing

To construct the training set in ULTRE-2, we first sampled a part of the training sessions from the search logs of Baidu-ULTR. Then, we removed the sessions with less than 10 consecutively recorded candidate documents, as well as those without any clicks. After that, we removed the queries with less than 10 sessions. Finally, we gave unique ids to the queries and documents and stored the clicks, other user behavior data and display information, and original text into different files, respectively. The detailed data formats can be found in the description file of the ULTRE-2 dataset.

The validation set and test set are the same as those of the Unbiased Learning for Web Search task [3] in WSDM Cup 2023. Note that when calculating the nDCG@10 value of the submitted runs, the queries with less than 2 candidate documents and those without any relevant document are neglected.

3.2 Feature Extraction

For each query-document pair, we extracted the pretrained and traditional features of 782 dimensions. We leveraged a pretrained BERT-based model [4] trained on the entire training set of Baidu-ULTR by the winner of WSDM Cup 2023 to output the pretrained semantic features. Specifically, we inputted the query, title, and abstract of each query-document pair into the BERT-based model, and extracted the 768-dimension CLS embedding as the pretrained features. As for the traditional features, we computed the text length, TF-IDF scores, BM25 scores, and proximity scores following Chen et al. [5], and leveraged Min-Max normalization to map their values to [0-1]. All the extracted features in ULTRE-2 are described in Table 3.

[1] https://github.com/ChuXiaokai/baidu_ultr_dataset

[2] https://drive.google.com/drive/folders/1DLnzOt3BXpo5RjoX6p52XZOOIGFCzPsM

[3] https://aistudio.baidu.com/aistudio/competition/detail/534

[4] This BERT model is trained with debiased click signals and is not fine-tuned with relevance annotations. The code, check-point, and description of this model can be found at https://github.com/lixsh6/Tencent_wsdm_cup2023

[5] https://github.com/xuanyuan14/THUIR_WSDM_Cup

Table 3: Description of the extracted features in ULTRE-2.

Feature Name	Feature Description
Pretrained features	768-dimension CLS embeddings outputted by a pretrained BERT-based ranking model.
Pretrained score	The relevance score outputted by the same pretrained BERT-based ranking model.
query_length	Length of the query.
title_length	Length of the title.
abstract_length	Length of the abstract.
BM25	BM25 score of title+abstract using Pyserini[6] (k1=1.6, b=0.87)
BM25_title	BM25 score of title using Pyserini (k1=1.6, b=0.87)
BM25_abstract	BM25 score of abstract using Pyserini (k1=1.6, b=0.87)
TF-IDF	TF-IDF score of title+abstract w.r.t. the query.
TF	TF score of title+abstract w.r.t. the query.
IDF	IDF score of title+abstract.
proximity-1	Averaged times of query terms appearing in title+abstract.
proximity-2	Averaged position of query terms appearing in title+abstract.
proximity-3	Number of query term pairs appearing in title+abstract within a distance of 5.
proximity-4	Number of query term pairs appearing in title+abstract within a distance of 10.

4 PARTICIPATION AND SUBMITTED RUNS

Table 4 summarizes the statistics of the runs submitted by the organizers and participants. Although 4 teams (excluding the organizers) have registered for the ULTRE-2 task, we only received 5 runs from one team (excluding the baseline models from the organizers).

4.1 Baseline Runs

Using all the pretrained and traditional features, we implemented 10 baselines with the ULTRA [7] toolkit. We trained 3 click baselines that use the raw click data to train a ranking model with different types of loss functions. We also implemented three IPS-based models with different click models, namely, IPS-PBM, IPS-DCM, and IPS-UBM. The parameters of the click models are estimated via the expectation-maximization (EM) [6] or the maximum-likelihood estimation (MLE) algorithm. Moreover, we tried to utilize the dual learning algorithm (DLA) [3] to jointly learn the above click models with the ranking model. In addition, we reproduced the propensity ratio scoring (PRS) model proposed by Want et al. [17], which reweighs the pairwise losses using the propensities of both clicked and not-clicked documents. Following Ai et al. [3], all the baselines utilize a 3-layer deep neural network (DNN) as the ranking model, and use a listwise softmax loss function (except that PRS uses a pairwise lambda loss function). The batch size is set to 256, the learning rate is set to 0.01, and each model is trained for 10K steps.

4.2 Submitted Runs from Participants

We received five runs from the CIR team. The dla run is another implementation of the DLA method proposed by Ai et al. [3]. However, the CIR team utilized different input features and ranking model settings from those of the organizers. Moreover, the Aux-DLA-LC and Scratch-DLA-LC runs enhanced the DLA method with developed label correction and negative sampling techniques. Besides, the CIR team used the human annotation labels of the validation set to train a GBDT model, namely the two lgb runs. Since the last two runs utilized annotations, they ought to serve as the "skylines".

5 RESULTS AND ANALYSIS

We evaluated the nDCG@10 and DCG@10 performance of each submitted run on the test set, and the results are shown in Table 5. The Scratch-DLA-LC run from the CIR team achieves the best nDCG@10 performance of 0.5355. We also conducted paired difference t-tests across all the submitted runs, and the results are shown in Table 6. Next, we will analyze the evaluation results and discuss two research questions:

5.1 RQ1. How effective are the ULTR models on the real-world dataset from Baidu?

Comparing the nDCG@10 performance of the click baselines and basic ULTR models implemented by the Organizer team, we can find that the DLA models outperform the IPS and PRS models. Besides, PRS and IPS-UBM perform even significantly worse than the naive click-softmax baseline. These findings suggest that DLA is more effective on the real-world dataset from Baidu, which may be due to its ability to adaptively adjust the propensity estimation via a joint learning mechanism. Moreover, the Scratch-DLA-LC and Aux-DLA-LC run from the CIR team perform significantly better than the dla run from the same team[8]. This indicates that the label correction and negative sampling techniques proposed by the CIR team can improve the DLA model by alleviating the false negative problem in real-world scenarios[9].

[7]https://github.com/ULTR-Community/ULTRA_pytorch

[8]The performance of the "dla" run is different from the "DLA-PBM" run because they utilized different input features, different ranking models, and different hyper-parameters.

[9]For more details, please see the participant paper of the CIR team.

Table 4: Statistics of the submitted runs in ULTRE-2.

Team	Submitted run	Description
Organizer	click-point	This model uses raw click data to train the ranking model with a point-wise sigmoid loss.
	click-pair	This model uses raw click data to train the ranking model with a pairwise binary cross entropy loss.
	click-softmax	This model uses raw click data to train the ranking model with a list-wise softmax loss, the same as that used by Ai et al. [3].
	IPS-PBM	This model is proposed by Joachims et al. [11] with PBM as the propensity model. The parameters of PBM are estimated via the EM algorithm.
	IPS-DCM	Proposed by Vardasbi et al. [15], this model computes the propensities based on a DCM [8] click model. The parameters of PBM are estimated via the MLE algorithm.
	IPS-UBM	We implemented this model by leveraging the UBM [7] click model as the propensity model for computing inverse propensity scores. The parameters of PBM are estimated via the EM [6] algorithm.
	DLA-PBM	This model is proposed by Ai et al. [3], which jointly learns the ranking model and propensity model, under the user behavior assumptions of PBM.
	DLA-DCM	We extended the DLA method to the cascade scenario and implemented the DLA-DCM model. In this model, we computed the propensities based on the assumptions of DCM, and still used the dual learning algorithm to jointly learn the ranking model and propensity model.
	DLA-UBM	In this model, we used UBM as the propensity model in the dual learning algorithm.
	PRS	This model is proposed by Wang et al. [17], which integrates the inverse propensity weighting on both the clicked documents and the non-clicked ones to reweigh the pairwise losses. The assumed propensity model is also PBM, whose parameters are also estimated via EM.
CIR	dla	This model is another implementation of the DLA method proposed by Ai et al. [3]. The utilized input features are different from those of the organizers.
	Aux-DLA-LC	This model enhances the above DLA model using label correction and negative sampling techniques. Specifically, it first trains a DLA model with clicks to obtain corrected click labels, and then utilizes the corrected labels to continue to train the DLA model.
	Scratch-DLA-LC	This model also enhances the DLA model using label correction and negative sampling techniques. Differently, it retrains a new DLA model using the corrected click labels.
	lgbBase	This model uses 80% human annotation labels of the validation set to train a GBDT and utilizes the same input features as the unbiased neural ranking models implemented by the CIR team.
	lgbAdd	Except for the addition of the best model score to the input features, everything is the same as the lgbBase model.
total	15 runs	

5.2 RQ2. How do different propensity models affect the ULTR models?

In this subsection, we go deep into the effect of the utilized propensity models on the ULTR models. Table 7 shows the click prediction performance of different click models, whose parameters are estimated via the MLE or EM algorithms. It can be found that UBM performs best in click prediction, followed by PBM and DCM. However, from Table 5 we can see that IPS-UBM performs significantly worse than IPS-PBM and IPS-DCM, even worse than click-pair and click-softmax baselines. Therefore, it is not reliable to select a click model as the propensity model for the IPS method simply based on its click prediction performance. Besides, we can find that the DLA methods with different propensity models show comparable performance and it seems that the dual learning algorithm is less picky about the propensity model than IPS.

Table 5: Official results of the submitted runs in ULTRE-2. The best-performing run is in bold, and the second best-performing run is underlined.

Team	Submitted run	nDCG@10	DCG@10
Organizer	click-point	0.3326	6.9492
	click-pair	0.5100	11.0423
	click-softmax	0.5144	11.1399
	IPS-PBM	0.5199	11.2603
	IPS-DCM	0.5131	11.1057
	IPS-UBM	0.4875	10.6537
	DLA-PBM	0.5216	11.2095
	DLA-DCM	0.5199	11.2603
	DLA-UBM	0.5196	11.2377
	PRS	0.4970	10.4867
CIR	dla	0.5247	11.2031
	Aux-DLA-LC	0.5326	11.3898
	Scratch-DLA-LC	**0.5355**	11.4538
	lgbAdd	0.5333	<u>11.4616</u>
	lgbBase	<u>0.5350</u>	**11.4794**

6 CONCLUSIONS

In this paper, we summarized the ULTRE-2 pilot task in NTCIR-17, including the task specification, dataset construction, implemented baselines, and official evaluation results of the submitted runs. In ULTRE-2, we evaluated the effectiveness of various ULTR models with a large-scale real-world user click log from Baidu.com, and the Scratch-DLA-LC run from the CIR team finally achieved the best nDCG@10 performance of 0.5355. We found that the DLA models perform better than the IPS and PRS models on our real-world click dataset. Moreover, the false negative problem may be another worth-noting problem in real-world scenarios other than position bias and can be alleviated by label correction and negative sampling techniques. In the future, we hope that the ULTRE-2 dataset can serve as a benchmark for evaluating the effectiveness of ULTR models in real-world scenarios. Moreover, it would be interesting to further explore how to utilize the rich user behavior information to develop more sophisticated ULTR models.

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A general framework for counterfactual learning-to-rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 5–14.

[2] Aman Agarwal, Xuanhui Wang, Cheng Li, Michael Bendersky, and Marc Najork. 2019. Addressing trust bias for unbiased learning-to-rank. In *The World Wide Web Conference*. 4–14.

[3] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 385–394.

[4] Mouxiang Chen, Chenghao Liu, Jianling Sun, and Steven CH Hoi. 2021. Adapting interactional observation embedding for counterfactual learning to rank. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 285–294.

[5] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*. 87–94.

[6] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)* 39, 1 (1977), 1–22.

[7] Georges E Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations.. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 331–338.

[8] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient multiple-click models in web search. In *Proceedings of the second acm international conference on web search and data mining*. 124–131.

[9] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[10] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately Interpreting Clickthrough Data as Implicit Feedback. (2005).

[11] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 781–789.

[12] Harrie Oosterhuis. 2023. Doubly Robust Estimation for Correcting Position Bias in Click Feedback for Unbiased Learning to Rank. *ACM Transactions on Information Systems* 41, 3 (2023), 1–33.

[13] Harrie Oosterhuis and Maarten de Rijke. 2020. Policy-aware unbiased learning to rank for top-k rankings. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 489–498.

[14] Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70, 1 (1983), 41–55.

[15] Ali Vardasbi, Maarten de Rijke, and Ilya Markov. 2020. Cascade model-based propensity estimation for counterfactual learning to rank. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2089–2092.

[16] Ali Vardasbi, Harrie Oosterhuis, and Maarten de Rijke. 2020. When inverse propensity scoring does not work: Affine corrections for unbiased learning to rank. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1475–1484.

[17] Nan Wang, Zhen Qin, Xuanhui Wang, and Hongning Wang. 2021. Non-clicks mean irrelevant? propensity ratio scoring as a correction. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 481–489.

[18] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 115–124.

[19] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position bias estimation for unbiased learning to rank in personal search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 610–618.

[20] Yurou Zhao, Zechun Niu, Feng Wang, Jiaxin Mao, Qingyao Ai, Tao Yang, Junqi Zhang, and Yiqun Liu. 2022. Overview of the NTCIR-16 Unbiased Learning to Rank Evaluation (ULTRE) Task. In *Proceedings of the NTCIR-16 Conference on Evaluation of Information Access Technologies*.

[21] Lixin Zou, Haitao Mao, Xiaokai Chu, Jiliang Tang, Wenwen Ye, Shuaiqiang Wang, and Dawei Yin. [n. d.]. A Large Scale Search Dataset for Unbiased Learning to Rank. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Table 6: The results of the paired difference t-tests across the submitted runs. The runs are arranged in descending order according to the performance of nDCG@10. -/*//*** indicate that the p-value $\geq 0.05/< 0.05/< 0.01/< 0.001$, respectively.**

	lgbBase	lgbAdd	Aux-DLA-LC	dla	DLA-PBM	DLA-DCM	IPS-PBM	DLA-UBM	click-softmax	IPS-DCM	click-pair	PRS	IPS-UBM	click-point
Scratch-DLA-LC	-	-	***	***	***	***	***	***	***	***	***	***	***	***
lgbBase		-	-	***	***	***	***	***	***	***	***	***	***	***
lgbAdd			-	***	***	***	***	***	***	***	***	***	***	***
Aux-DLA-LC				***	***	***	***	***	***	***	***	***	***	***
dla					-	*	*	*	***	***	***	***	***	***
DLA-PBM						-	-	-	***	***	***	***	***	***
DLA-DCM							-	-	***	***	***	***	***	***
IPS-PBM								-	***	***	***	***	***	***
DLA-UBM									***	***	***	***	***	***
click-softmax										-	***	***	***	***
IPS-DCM											**	***	***	***
click-pair												***	***	***
PRS													**	***
IPS-UBM														***

Table 7: The click prediction performance of different click models. The best-performing model is in bold, and the second best-performing model is underlined.

Click Model	Log-likelihood	PPL@10	Conditional PPL@10
DCM	-0.2508	1.2680	1.3038
PBM	<u>-0.2055</u>	<u>1.2535</u>	<u>1.2535</u>
UBM	**−0.1949**	**1.2533**	**1.2417**