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/10.20736/0002001317

THUIR at the NTCIR-17 FairWeb-1 Task: An Initial Exploration of the Relationship Between Relevance and Fairness

Yiteng Tu
Renmin University of China
Beijing 100872, China
yitengtu16@gmail.com

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

Zhumin Chu
DCST, Tsinghua University
Zhongguancun Laboratory
Beijing 100084, China
chuzm19@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

The fairness of search systems has become an important research topic for the IR community. This paper presents and discusses the efforts of the THUIR team in developing effective and fair retrieval models and ranking algorithms in the NTCIR-17 FairWeb-1 Task [22]. Specifically, we utilize several different methods in all 5 submitted runs including reranking, learning-to-rank, and search result diversification algorithms to deal with the group fairness problem in web search. The final report of the FairWeb-1 Task indicates that our methods have outperformed other competitors on both result relevance and fairness. In terms of the GFR (Group Fairness Relevance) metric, our methods respectively outperform the second-ranked team by 9.74%, 17.8%, and 19.8% on three topics of queries.

KEYWORDS

Information Retrieval, Fairness, Learning-to-rank, Research Result Diversification

TEAM NAME

THUIR

SUBTASKS

FairWeb

1 INTRODUCTION

Nowadays, search engines are essential tools for humans as they efficiently filter out redundant information and help people find relevant information that satisfies their information needs. However, most search engines tend to only focus on ranking documents based on their popularity and relevance with respect to the user submitted queries [4, 7–9, 23]. This phenomenon can lead to issues such as biased information filtering and unbalanced result distributions for different users. For example, when people search for movies on a particular topic, highly popular movies produced in developed regions often dominate the top search results. This is unfair to movies produced in relatively underdeveloped regions or less well-known movies, which could hinder the long-term prosperity of the industry. Therefore, how to ensure exposure fairness without hurting the quality of result relevance in search systems has become an important challenge for search engines.

The FairWeb task is a new NTCIR pilot task that considers "not only document relevance from a viewpoint of search engine users but also group fairness from a viewpoint of entities that are being sought" [22]. Compared to traditional ad-hoc retrieval tasks, participants need to make a precise trade-off between the relevance and fairness of search results in order to produce an effective retrieval system. We, the THUIR team, have participated in this task and submitted 5 runs using different methods. Specifically, we explore the impact of methods such as neural network reranking, learning-to-rank, and search result diversification on the relevance and fairness of retrieval results.

The official results [22] indicate that our approaches achieve the best results on all relevance and fairness metrics. Among all methods, LightGBM [5] and PM2 [3] algorithms are the best, taking the first place in the majority of evaluation metrics. Also, we observe that for queries of different topics, methods demonstrating excellent performance in relevance also perform well in terms of fairness. Based on this, we believe that in ad-hoc retrieval settings, relevance and fairness are not two opposing factors to a certain extent, and it is possible for us to achieve a win-win situation for both aspects.

2 METHODS

In the FairWeb-1 task, we submit five runs generated by different methods (Table 1). Details of these submissions are elaborated one by one in this section.

2.1 Run 1: Sparse Retrieval

In our first attempt, we choose two classical sparse retrieval strategies, BM25 [16] and QLD [15] for relevant document retrieval. Then we use a simple fusion strategy, Reciprocal Rank Fusion (RRF), to combine the results of different ranking lists to obtain the final answer.

2.1.1 Data Process. The official candidate document collection, ChuWeb21D-60 [1], is composed of about 49.8M HTML web pages. Although the original HTML document contains some structured information, it also contains a large number of redundant tokens,

[1]https://github.com/chuzhumin98/Chuweb21D

Table 1: An overview of THUIR's submissions on the NTCIR-17 FairWeb-1 Task

Run Number	Run Name	Description
Run 1	THUIR-QD-RG-1	Directly aggregate the retrieved results of sparse retrieval by RRF
Run 2	THUIR-QD-RG-2	Learning-to-rank based on sparse and dense relevance features
Run 3	THUIR-QD-RR-3	Add feature information to the query text for reranking
Run 4	THUIR-QD-RR-4	A search result diversification algorithm, PM2
Run 5	THUIR-D-RR-5	A search result diversification algorithm, xQuAD

so we decide to extract the main text from the HTML document. Following Yang et al. [24], we employ the bs4 [2] python package to parse these HTML files to extract their text information for the next stage's retrieval.

The query data consists of three topics, researchers (R topics), movies (M topics), and Youtube contents (Y topics), each containing fifteen queries. Each query is divided into two sections, *Query* and *Description*. In the following, we use *Q-queries*, *D-queries*, and *QD-queries* to respectively denote querying with only the *Query* section, querying with only the *Description* section, and querying with both sections together.

2.1.2 Document Retrieval. As for document retrieval, we choose BM25 and QLD, two classic sparse retrieval algorithms. BM25 [16] adopts the tf-idf signal to measure term weights and calculate the relevance score between a query and a document. QLD [15] is another efficient statistical probabilistic model whose relevance score is regarded as the probability of generating a query when given a document. Their calculation formulas are shown below:

$$BM25(d,q) = \sum_{t_i} \frac{IDF(t_i) \times TF(t_i,d) \times (k_1+1)}{TF(t_i,d) + k_1 \times (1 - b + b \times \frac{len(d)}{avgdl})} \tag{1}$$

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

We respectively conduct the retrieval using these two algorithms for *Q-queries*, *D-queries*, and *QD-queries*. Then we add RM3 [6] pseudo-relevance feedback to both algorithms and repeat the retrieval process. For each type of query, we have 4 sparse retrieval ranking lists.

2.1.3 Reciprocal Rank Fusion. After we have several ranking lists, we need to integrate their results. Since that *Q-queries* are too short and can easily lead to ambiguity, in this run, we only integrate the 8 retrieval results of *D-queries* and *QD-queries*. Reciprocal rank fusion (RRF) is a simple but effective rank-based aggregation method. Given a set of ranking lists R of a query, we compute the RRF score of a document d:

$$RRF(d) = \sum_{r \in R} \frac{1}{k + r(d)} \tag{3}$$

where $r(d)$ denotes the position of document d in ranking r, and k is a hyper-parameter.

[2] https://beautifulsoup.readthedocs.io/zh_CN/v4.4.0/

2.2 Run 2: LightGBM

In Run 1, we only consider ranking signals at the term level, which is relatively unilateral. Therefore, in the second run, we decide to incorporate semantic-level neural network signals. Following previous work [2, 10, 24], we construct a learning-to-rank model based on both types of signals. For neural models, we choose MonoBERT [13] reranker and MonoT5 [14] reranker that are capable of fine-grained interactions between queries and documents. Following Chen et al. [2], we select the lightweight LightGBM [5] as our learning-to-rank model.

2.2.1 Reranker. MonoBERT [13] concatenates the query and the document together as input to a BERT model and then feeds its [CLS] token into a single-layer neural network to obtain the relevance score. As for MonoT5 [14], the reranking task is cast as a sequence-to-sequence task. It ranks the documents according to the generation probabilities of the "true" token with the following input format:

$$Query : \{query\}\ Document : \{document\}\ Relevance :$$

In this step, we employ two models to rerank all retrieved documents of the three types of queries (*Q*, *D*, *QD*) in Run 1, so that we can get 6 new ranking lists.

2.2.2 Learning-to-rank. For each query, we use the 12 sparse retrieval scores from Run 1 and the 6 neural scores described above as the features to conduct learning-to-rank through LightGBM [5]. Since FairWeb-1 does not provide annotated relevance labels as training data, we adopt the data from the NTCIR WWW2-3 [12, 19] for training LightGBM.

2.3 Run 3: Query Augmentation

In Run 3, we begin to take the group fairness factors into account. In this run, we incorporate fairness information into the semantics of queries. A naive way to achieve this goal is simply adding the entity attribute information to the query text. Let's take the example of M topics that need to consider regional fairness. We simply add a suffix *", and these movies are from Africa/America/Antarctica/..."* to the query. For each value of the attribute, we generate a ranking list via the MonoT5 reranker.

Next, we need to synthesize the ranking lists of different values into a single ranking list of that attribute. Based on the target distribution of the attribute, we randomly sample one attribute value at a time, take out the head element of the list corresponding to this value, and delete the element in the lists of the other values. After having ranking lists of different attributes, we still use RRF to merge their results.

2.4 Run 4: PM2

In Run 4 and Run 5, we try to make a more accurate estimate of the attribute score. Due to the limitations of the web crawler, we attempt two different ways of estimating attribute scores for different topics. For the ranking process with group fairness, we note that the target fairness distribution of an attribute in this task can be regarded as the distribution of subtopic importance in search result diversification algorithms. Thus we decide to employ the search result diversification algorithms to solve the group fairness problem. Specifically, for a document d and a subtopic t_i (a particular value of an attribute), we roughly assume that all subtopics occur with equal probability. Therefore, we can readily estimate the document's coverage of this subtopic $P(d|t_i)$ by the estimated attribute score $P(t_i|d)$:

$$P(d|t_i) = \frac{P(t_i|d)P(d)}{P(t_i)} \propto P(t_i|d) \tag{4}$$

In this run, the algorithm we adopt is PM2 [3].

2.4.1 Estimation of Attribute Scores. On the one hand, for M topics and Y topics, since the names of the movies or the video creators are not explicitly given in the document text, we first need to extract all possible entities from a document to find out the possible target entities. We utilize the Stanford NER [3] toolkit to extract all person, organization, and location (only used for M topics) name entities in each document for the next step's web crawling. For each document, we randomly sample ten entities that have not been searched for crawling and record their results, while the entities that have already been searched are directly fetched from the search history. Specifically, during the crawling process, we select the first movie or video creator on the search results' first page whose edit distance from the input text is no more than 3. Then we record the region and rating information or number of subscribers. If one document can be extracted for entities, we directly consider the proportion of crawled entities' attribute values as its attribute distribution $P(t_i|d)$. Otherwise, we directly use the target distribution of the attribute.

On the other hand, for R topics, due to Google Scholar's [4] strict limitation on crawlers, we can only roughly estimate based on the document content. We approximate the document's gender distribution through the relative proportions of gender-related terms that appeared in the document. However, there is still nothing we can do about the h-index information. Therefore, we do not take the h-index into account for R topics in the following part.

2.4.2 PM2. PM2 [3] is constructed based on the Sainte-Lague formula used for voting in New Zealand parliamentary elections. It determines the proportion of seats for a party based on the number of ballots it receives. It is reflected in search problems that the higher the importance of a subtopic, the lower the number of corresponding documents in the selected set, and the more priority should be given to improving this subtopic. From all the subtopics t_i, it selects the one (t_{i^*}) that requires the largest improvement quotient:

$$qt_i = \frac{w_i}{2s_i + 1} \tag{5}$$

[3]https://nlp.stanford.edu/software/CRF-NER.html
[4]https://scholar.google.com/

where w_i denotes the importance of the subtopic to a query, or the target probability of the attribute in our task, and s_i represents the degree to which a subtopic is occupied in the selected document set. Then we add a document into the selected set according to the quotient qt as well as the subtopic coverage score $P(d|t_i)$:

$$d^* = \arg\max_d \lambda \cdot qt_{i^*} \cdot P(d|t_{i^*}) + (1-\lambda) \cdot \sum_{i \neq i^*} qt_i \cdot P(d|t_i) \tag{6}$$

Note that we directly utilize $P(t_i|d)$ to estimate $P(d|t_i)$, to account for the relevance factor, we separately apply min-max normalization to the scores of *QD-queries* and *D-queries* via MonoT5, then take the average of the two kinds of scores as the relevance score, and multiply it on the right side of e.q. 6 as the final score for selecting d^*. After that, update the ratio s_i:

$$s_i = s_{i^*} + \frac{P(d^*|t_i)}{\sum_{t_j} P(d^*|t_j)} \tag{7}$$

Finally, we use RRF to merge the results of different attributes considered by a query.

2.5 Run 5: xQuAD

In the last run, we turn to another search result diversification algorithm, xQuAD [21]. Its score is expressed as a linear combination of the relevance score and diversity score of a given document. The diversity score is calculated as the product of the subtopic importance, the document's coverage of the subtopic, and the novelty score:

$$\arg\max_d \lambda \cdot \overbrace{P(d|q)}^{relevance\ score} + (1-\lambda) \cdot \overbrace{\sum_i w_i \cdot P(d|t_i) \cdot \underbrace{\prod_{d_j \in S} (1 - P(d_j|t_i))}_{novelty\ score}}^{diversity\ score} \tag{8}$$

where S denotes the selected document set. We regard the average retrieval scores of BM25, QLD, and QLJM for *D-queries* as the relevance score $P(d|q)$. Here the original retrieval scores require min-max normalization.

3 EXPERIMENT RESULTS AND ANALYSIS

3.1 Evaluation Metrics

The official relevance metrics are Expected Reciprocal Rank (ERR) [1] and intentwise Rank-Biased Utility (iRBU) [20]. For fairness evaluation, the FairWeb task utilizes divergence functions including JSD (Jensen-Shannon Divergence), NMD (Normalised Match Distance), and RNOD (Root Normalised Order-aware Divergence), to calculate the similarity between the entity distribution in the results and the target distribution. The fairness of each attribute is measured using the GF (Group Fairness) [18] score, which takes into account both the user attention decay and the distribution similarity. As for the different attributes and relevance factors that need to be considered together for a query, the GFR (Group Fairness Relevance) [18] metric is employed for a comprehensive assessment.

3.2 Implementation Details

In Run 1 and Run 2, we adopt the sparse retrieval techniques implemented by Pyserini [11] with default hyper-parameters. In the

Table 2: The official relevance evaluation over different query topics, where R, M, and Y respectively represent queries about researchers topics, movies topics, and YouTube contents topcis. We present 5 runs by THUIR, 6 baseline runs, as well as the optimal results of other participants. ">" means stastistically significantly outperforms (according to a randomised Tukey HSD test with $B = 5,000$ trials and $\alpha = 0.05$ [17]) among all 28 runs. For example, in terms of ERR on all topics, THUIR-QD-RR-4 statistically significantly outperforms the runs ranked at 18 through 28. All the results are from [22].

	R topics		M topics		Y topics		All topics	
Run	Mean ERR	Mean iRBU	Mean ERR	Mean iRBU	Mean ERR	Mean iRBU	Mean ERR	Mean iRBU
THUIR-QD-RG-1	0.1918	0.6013	0.1608	0.4400	0.1099	0.4026	0.1542(>27-28)	0.4813(>26-28)
THUIR-QD-RG-2	**0.2638**	**0.6560**	0.2280	0.6923	0.1144	0.3749	0.2021(>22-28)	**0.5744(>22-28)**
THUIR-QD-RR-3	0.2276	0.5804	**0.2653**	**0.7230**	0.1293	0.3919	0.2074(>20-28)	0.5651(>23-28)
THUIR-QD-RR-4	0.2460	0.5957	0.2518	0.6859	**0.1438**	**0.4404**	**0.2139(>18-28)**	0.5740(>22-28)
THUIR-D-RR-5	0.1421	0.5351	0.1223	0.5316	0.1009	0.3649	0.1218(>27-28)	0.4772 (>26-28)
Best of Other Participants	0.2131	0.5582	0.2434	0.5819	0.1365	0.3755	0.1847(>26-28)	0.4977(>26-28)
run.bm25-depThre3-Q	0.1989	0.5489	0.1712	0.5035	0.0471	0.2202	0.1390(>27-28)	0.4242(>26-28)
run.bm25-depThre3-D	0.1509	0.4801	0.1564	0.4337	0.0266	0.1735	0.1113(>27-28)	0.3624(>27-28)
run.qld-depThre3-Q	0.1567	0.5518	0.1653	0.4958	0.0459	0.2514	0.1226(>27-28)	0.4330(>26-28)
run.qld-depThre3-D	0.1749	0.5695	0.1187	0.3728	0.0442	0.2194	0.1126(>27-28)	0.3872(>27-28)
run.qljm-depThre3-Q	0.2104	0.4971	0.2114	0.6026	0.0266	0.2010	0.1495(>27-28)	0.4336(>26-28)
run.qljm-depThre3-D	0.1459	0.4361	0.1478	0.4883	0.0520	0.2424	0.1152(>27-28)	0.3889(>27-28)

Table 3: The official fairness evaluation over the R topics. We present 5 runs by THUIR, 6 baseline runs, as well as the optimal results of other participants. ">" means stastistically significantly outperforms (according to a randomised Tukey HSD test with $B = 5,000$ trials and $\alpha = 0.05$) among all 28 runs.

Run	Mean GF^{JSD} (GENDER)	Mean GF^{NMD} (HINDEX)	Mean GF^{RNOD} (HINDEX)	Mean GFR
THUIR-QD-RG-1	0.5823(>26-28)	0.5569(>26-28)	0.5257(>26-28)	0.5698(>26-28)
THUIR-QD-RG-2	**0.5831(>26-28)**	**0.5841(>26-28)**	**0.5352(>26-28)**	**0.5914(>26-28)**
THUIR-QD-RR-3	0.4987(>26-28)	0.5247(>26-28)	0.4875(>26-28)	0.5222(>26-28)
THUIR-QD-RR-4	0.5086(>26-28)	0.5164(>26-28)	0.4720(>26-28)	0.5254(>26-28)
THUIR-D-RR-5	0.5351(>26-28)	0.5080(>26-28)	0.4841(>26-28)	0.5181(>26-28)
Best of Other Participants	0.5374(>26-28)	0.5195(>26-28)	0.4866(>26-28)	0.5274(>26-28)
run.bm25-depThre3-Q	0.5096(>26-28)	0.4977(>26-28)	0.4605(>26-28)	0.5064(>26-28)
run.bm25-depThre3-D	0.4694(>26-28)	0.4400(>26-28)	0.4155(>26-28)	0.4550(>26-28)
run.qld-depThre3-Q	0.5356(>26-28)	0.5152(>26-28)	0.4807(>26-28)	0.5227(>26-28)
run.qld-depThre3-D	0.5497(>26-28)	0.5306(>26-28)	0.4975(>26-28)	0.5389(>26-28)
run.qljm-depThre3-Q	0.4315(>26-28)	0.4362(>26-28)	0.3999(>26-28)	0.4428(>26-28)
run.qljm-depThre3-D	0.4120(>26-28)	0.4038(>26-28)	0.3824(>26-28)	0.4101(>26-28)

other runs, we directly use the official baseline sparse retrieval results [5] and filter out documents with empty content. In all RRF runs, we have $k = 60$. For the selection of reranker models, we choose *castorini/monot5-3b-msmarco-10k* for MonoT5 and *castorini/monobert-large-msmarco* for MonoBERT. The iteration number and learning rate of LightGBM are 1000 and 0.01 respectively. In Run 4 and Run 5, the values of the hyper-parameter λ are 0.5 and 0.25 respectively.

3.3 Results and Analysis

Table 2 shows the official relevance metrics of all our submitted runs. Run 2, Run 3, and Run 4 respectively achieve the best results on different topics. Across all topics, Run4 performs the best in terms of ERR, surpassing the second run by 3.13% as well as statistically significantly outperforming the runs ranked at 18 through 28. For iRBU, both Run2 and Run4 show excellent performance that is over 0.57. They form the top cluster that significantly outperforms the runs ranked at 22nd and beyond. Run 3 is right behind them on the two metrics. For fairness metrics, Run2 takes the lead on R topics (Table 3) with its GFR surpassing the second place by 3.8%. On M topics (Table 4), both Run2 and Run3 demonstrate strong performance, ranking in the top two for each fairness metric, and their GFR scores exceed 0.61. Run4 performs the best on Y topics, 7.2% over the second place in terms of GFR.

According to the experimental results, Run 2, Run 3, and Run 4 are significantly better than Run 1 and Run 5. Run 2-4 all employ a neural reranker for relevance reranking. The results show that even in a zero-shot scenario, the fine-grained reranker can still accurately assess the relevance between queries and documents and may promote fairness to some extent. In submissions without neural reranker, the performance of Run 1 is slightly better than Run 5. They are constructed based on our own sparse retrieval results and the officially provided baseline results, respectively. We speculate that there might be two reasons contributing to their different performance: 1) Run 1 takes into account the results of 8 sparse

[5] https://waseda.app.box.com/v/fairweb1baselines

Table 4: The official fairness evaluation over the M topics. We present 5 runs by THUIR, 6 baseline runs, as well as the optimal results of other participants. ">" means stastistically significantly outperforms (according to a randomised Tukey HSD test with $B = 5,000$ trials and $\alpha = 0.05$) among all 28 runs.

Run	Mean GF^{JSD} (ORIGIN)	Mean GF^{NMD} (RATINGS)	Mean GF^{RNOD} (RATINGS)	Mean GFR
THUIR-QD-RG-1	0.3395(>27-28)	0.4025(>27-28)	0.3684(>27-28)	0.3827(>27-28)
THUIR-QD-RG-2	**0.5684(>27-28)**	0.6330(>27-28)	**0.5788(>27-28)**	**0.6132(>27-28)**
THUIR-QD-RR-3	0.5391(>27-28)	**0.6433(>26-28)**	0.5683(>27-28)	0.6101(>27-28)
THUIR-QD-RR-4	0.5332(>27-28)	0.6118(>27-28)	0.5435(>27-28)	0.5875(>27-28)
THUIR-D-RR-5	0.4900(>27-28)	0.5307(>27-28)	0.4983(>27-28)	0.5066(>27-28)
Best of Other Participants	0.4768(>27-28)	0.5169(>27-28)	0.4758(>27-28)	0.4996(>27-28)
run.bm25-depThre3-Q	0.4135(>27-28)	0.4623(>27-28)	0.4283(>27-28)	0.4484(>27-28)
run.bm25-depThre3-D	0.3401(>27-28)	0.3993(>27-28)	0.3630(>27-28)	0.3789(>27-28)
run.qld-depThre3-Q	0.4275(>27-28)	0.4668(>27-28)	0.4351(>27-28)	0.4528(>27-28)
run.qld-depThre3-D	0.3122(>27-28)	0.3507(>27-28)	0.3208(>27-28)	0.3353(>27-28)
run.qljm-depThre3-Q	0.4716(>27-28)	0.5462(>27-28)	0.4871(>27-28)	0.5205(>27-28)
run.qljm-depThre3-D	0.4273(>27-28)	0.4606(>27-28)	0.4211(>27-28)	0.4456(>27-28)

Table 5: The official fairness evaluation over the Y topics. We present 5 runs by THUIR, 6 baseline runs, as well as the optimal results of other participants. ">" means stastistically significantly outperforms (according to a randomised Tukey HSD test with $B = 5,000$ trials and $\alpha = 0.05$) among all 28 runs.

Run	Mean GF^{NMD} (SUBSCS)	Mean GF^{RNOD} (SUBSCS)	Mean GFR
THUIR-QD-RG-1	0.3830(>27-28)	0.3638(>27-28)	0.3832(>27-28)
THUIR-QD-RG-2	0.3423(>27-28)	0.3141(>27-28)	0.3445(>27-28)
THUIR-QD-RR-3	0.3601(>27-28)	0.3297(>27-28)	0.3608(>27-28)
THUIR-QD-RR-4	**0.4112(>27-28)**	**0.3809(>27-28)**	**0.4107(>27-28)**
THUIR-D-RR-5	0.3550(>27-28)	0.3396(>27-28)	0.3523(>27-28)
Best of Other Participants	0.3315(>27-28)	0.3157(>27-28)	0.3428(>27-28)
run.bm25-depThre3-Q	0.2112	0.2039	0.2121
run.bm25-depThre3-D	0.1777	0.1731	0.1733
run.qld-depThre3-Q	0.2451	0.2391	0.2453
run.qld-depThre3-D	0.2155	0.2100	0.2147
run.qljm-depThre3-Q	0.2071	0.2038	0.2024
run.qljm-depThre3-D	0.2425	0.2329	0.2377

queries, while Run 5 only considers the results of 3 sparse queries, thus Run 1 includes more ranking information. 2) In our own sparse retrieval, the main content from all documents is extracted while redundant HTML tokens are removed. Additionally, both the *Query* section and the *Description* section of queries are used for retrieval, which helps to better capture their term-level similarity.

3.4 Discussions

Looking at the results across all topics, fairness-aware ranking methods do not show a significant advantage over methods that only consider relevance. This could be due to our inability to accurately assess the entity attribute distribution for each document.

We also observe that in different query topics, if a method performs exceptionally well in terms of relevance, it also has strong performance in terms of fairness. Therefore, we realize that to a certain extent, relevance and fairness are not two opposing goals; they can be jointly optimized within a certain degree. This is because search results with higher relevance contain more relevant entities. However, a real search engine needs to consider extra factors like popularity and personalization in addition to relevance. These extra factors can have a significant impact on the fairness of search result entities. Therefore, the attributes of these related entities should also exihibit randomness when only considering relevance factors and the query itself is unbiased. These large amounts of randomly distributed related entities not only improve fairness evaluation metrics, but also facilitate further optimization towards fairness. Hence, we can achieve a win-win situation for both relevance and fairness to some extent.

4 CONCLUSIONS

This paper presents our participation in the NTCIR-17 FairWeb-1 task. We submit 5 runs with various methods. We achieve first place in all metrics. Meanwhile, our results indicate that relevance and fairness are not in opposition to some degree and it is possible to achieve their joint optimization.

REFERENCES

[1] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 621–630.

[2] Jia Chen, Haitao Li, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023. THUIR at WSDM Cup 2023 Task 1: Unbiased Learning to Rank. *arXiv preprint arXiv:2304.12650* (2023).

[3] Van Dang and Bruce W Croft. 2013. Term level search result diversification. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 603–612.

[4] Qian Dong, Yiding Liu, Qingyao Ai, Haitao Li, Shuaiqiang Wang, Yiqun Liu, Dawei Yin, and Shaoping Ma. 2023. Iˆ 3 Retriever: Incorporating Implicit Interaction in Pre-trained Language Models for Passage Retrieval. *arXiv preprint arXiv:2306.02371* (2023).

[5] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).

[6] Victor Lavrenko and W Bruce Croft. 2001. Relevance-based language models. (2001), 120–127.

[7] Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. SAILER: Structure-aware Pre-trained Language Model for Legal Case Retrieval. *arXiv preprint arXiv:2304.11370* (2023).

[8] Haitao Li, Qingyao Ai, Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Zheng Liu, and Zhao Cao. 2023. Constructing Tree-based Index for Efficient and Effective Dense Retrieval. *arXiv preprint arXiv:2304.11943* (2023).

[9] Haitao Li, Jia Chen, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023. Towards Better Web Search Performance: Pre-training, Fine-tuning and Learning to Rank. *arXiv preprint arXiv:2303.04710* (2023).

[10] Haitao Li, Weihang Su, Changyue Wang, Yueyue Wu, Qingyao Ai, and Yiqun Liu. 2023. THUIR@COLIEE 2023: Incorporating Structural Knowledge into Pre-trained Language Models for Legal Case Retrieval. arXiv:2305.06812 [cs.IR]

[11] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2356–2362.

[12] 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.

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

[14] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713* (2020).

[15] Jay M Ponte and W Bruce Croft. 2017. A language modeling approach to information retrieval. In *ACM SIGIR Forum*, Vol. 51. ACM New York, NY, USA, 202–208.

[16] 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.

[17] Tetsuya Sakai. 2018. Laboratory experiments in information retrieval. *The information retrieval series* 40 (2018).

[18] Tetsuya Sakai, Jin Young Kim, and Inho Kang. 2023. A Versatile Framework for Evaluating Ranked Lists in terms of Group Fairness and Relevance. *ACM Transactions on Information Systems* (2023).

[19] 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* (2020).

[20] Tetsuya Sakai and Zhaohao Zeng. 2020. Retrieval evaluation measures that agree with users' SERP preferences: Traditional, preference-based, and diversity measures. *ACM Transactions on Information Systems (TOIS)* 39, 2 (2020), 1–35.

[21] Rodrygo LT Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. 2010. Explicit search result diversification through sub-queries. In *Advances in Information Retrieval: 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010. Proceedings 32*. Springer, 87–99.

[22] Sijie Tao, Nuo Chen, Tetsuya Sakai, Zhumin Chu, Hiromi Arai, Ian Soboroff, Nicola Ferro, and Maria Maistro. 2023. Overview of the NTCIR-17 FairWeb-1 Task. *Proceedings of NTCIR-17* (2023).

[23] Xiaohui Xie, Qian Dong, Bingning Wang, Feiyang Lv, Ting Yao, Weinan Gan, Zhijing Wu, Xiangsheng Li, Haitao Li, Yiqun Liu, et al. 2023. T2Ranking: A large-scale Chinese Benchmark for Passage Ranking. *arXiv preprint arXiv:2304.03679* (2023).

[24] 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).