An Exploration of Learning-to-re-rank Using a Two-step Framework for Fair Ranking

Fumian Chen
Institute for Financial Services Analytics
University of Delaware
fmchen@udel.edu

Hui Fang
Department of Electrical and Computer Engineering
University of Delaware
hfang@udel.edu

Abstract

As fairness has been attracting increasing attention in search engines, producing both fair and relevant rankings has become a major challenge for many information retrieval systems. In 2022, TREC fair ranking track launched an ad-hoc retrieval task for Wikipedia editors, which returns not only relevant documents of each query but also provides fair exposure to each document. The main goal of our participation in this track is to explore a fairness-aware two-step fair ranking framework using supervised learning-based re-rankers. Given a query, we first retrieve relevant documents and then use the re-ranker to re-rank the documents so that the final ranking is fair evaluated by the attention-weighted rank fairness (AWRF). We utilize the simple yet well-performed BM25 retrieval model to obtain relevant documents of each query. We tested a gradient-boosted tree-based (GBDT) LambdaMART model and a neural-based multilayer perceptron model. To better train the supervised learning models, we also extracted contextual-based features to augment our training data. We encoded fairness through a pre-processing method by constructing fairness scores. Our results also show the possibility of leveraging supervised learning-based re-rankers and contextual features.

1 Introduction

Information retrieval (IR) systems, such as open-web search, play a crucial role in our daily lives. Countless IR algorithms have been proposed to improve relevance-based user utility for better search experience (Zehlike, Yang, & Stoyanovich, 2021) in the past decades. Concerns about algorithmic transparency and fairness, however, are rising as well, especially when historical data has shown disproportional exposure to different groups, and providing fair exposure and equal opportunity is crucial for long-term sustainability. In this paper, we discuss our participation at TREC 2022 fair ranking track [1]. We explored the two-step fairness-aware framework with the BM25 retrieval model and fairness-aware re-ranker using supervised learning algorithms, as shown in Figure 1. Our goal is to examine the possibility of utilizing the supervised learning re-ranker and also the value of contextual features in fair-ranking tasks. Since most of the current fair ranking frameworks focus on a single fairness group, the intersectional fairness setting provided by TREC is also a great source for us to explore intersectional fairness. In the following sections, we first describe the task as long as the evaluation metrics provided by TREC and then we elaborate on our methods and discuss their performance.

[1] https://fair-trec.github.io/

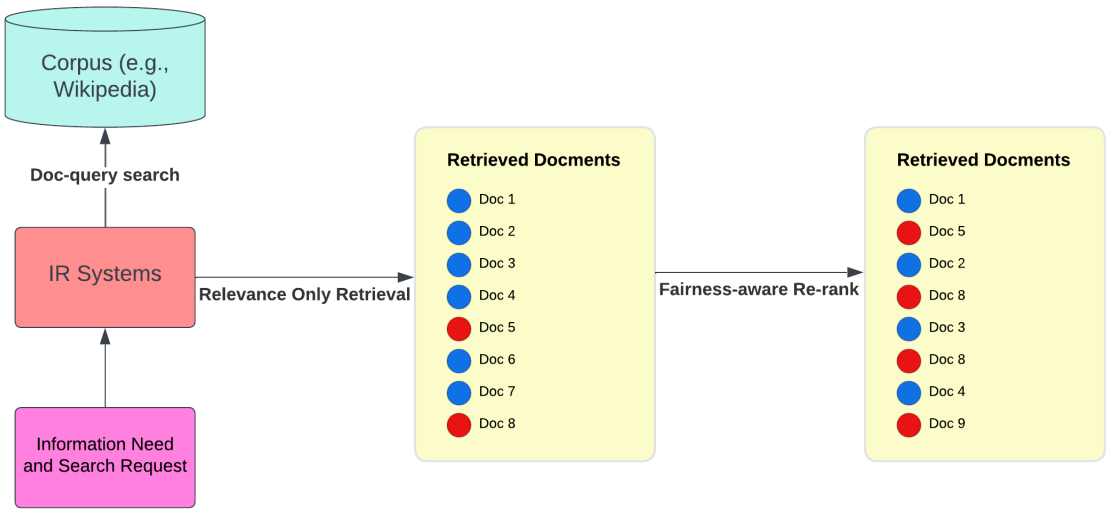

Figure 1: Two-step fairness-aware ranking framework.

2 Task Description and Evaluation Metrics

In TREC 2022 fair ranking track, there are two tasks and we participated in task 1 (the WikiProject Coordinators) to provide a relevant and fair document list for editors searching for work to do. We were provided a corpus of documents with more than six million English Wikipedia articles and a set of queries associated with binary annotated relevant documents. Each document is also associated with zero to many fairness categories. The task output is a single ranking per query, which consists of 500 articles. Ideally, the final ranked list of documents should be (1) relevant to given queries and (2) fair with respect to multiple fairness categories by providing fair exposure to articles.

Submitted runs will be judged by both relevance and fairness. Relevance is judged by nDCG[2] with binary relevance and logarithmic decay, which is broadly used in various ranking tasks.

$$nDCG@k = \frac{DCG@k}{iDCG@k} \tag{1}$$

where $iDCG$ is the ideal discounted cumulative gain:

$$iDCG = \sum_{i=1}^{rel\_docs_k} \frac{rel_i}{log_2(i+1)} \tag{2}$$

Fairness is judged by attention-weighted ranking fairness (AWRF) (Sapiezynski, Zeng, E Robertson, Mislove, & Wilson, 2019; Raj, Wood, Montoly, & Ekstrand, 2020).

$$\text{AWRF}(\pi) = \Delta(\epsilon(\pi), \hat{\epsilon}) \tag{3}$$

where $\epsilon(\pi)$ is the system produced exposure distribution and $\hat{\epsilon}$ is the target exposure distribution. Δ is a function of one minus the Jenson-Shannon divergence [3].

$$\Delta(\hat{\epsilon}, \epsilon(\pi)) = 1 - \frac{1}{2}(D_{KL}(\epsilon|M) + D_{KL}(\hat{\epsilon}|M)) \tag{4}$$

$$M = \frac{1}{2}(\epsilon + \hat{\epsilon}) \tag{5}$$

[2] https://en.wikipedia.org/wiki/Discounted_cumulative_gain#Normalized_DCG
[3] https://en.wikipedia.org/wiki/Jensen-Shannon_divergence

and $D_{KL}(\cdot)$ is the Kullback–Leibler divergence [4].

Like the relevance metric, fairness metrics also incorporate log discounting to ensure that the metrics are attention-weighted.

3 Methods

We adopted a two-step framework with (1) a relevance retrieval model and (2) a fairness-aware re-ranker. Relevance retrieval, as a well-explored field, has numerous retrieval functions being proposed and deployed in various domains. Therefore, in this work, we mainly focus on the re-ranker step and fairly assume documents retrieved from the first step are relevant. The performance of the re-ranker will be judged by the improvement over the first step.

Here, we leverage the simple yet well-performed BM25 as our retrieval mode. To do so, we index the corpus and achieve BM25 [5] by Pyserini [6], a toolkit for information retrieval research. Since we were only given binary annotations for relevance and the fairness metrics are based on distribution, we retrieved 5000 documents for each query.

$$BM25(D,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{freq(q_i,D) \cdot (k_1+1)}{freq(q_i,D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})} \tag{6}$$

$$IDF(q_i) = \ln(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1) \tag{7}$$

3.1 Fairness-aware Re-ranker

For the fairness-aware re-ranker, we use two supervised learning-to-rank (LTR) models, LambdaMART (Burges, 2010) and multi-layer perceptron (MLP), to re-rank retrieved documents after the first step. To encode fairness, we use a pre-processing method, modifying the ground truth label such that:

$$y = \text{binary relevance} + \gamma * \text{fairness score} \tag{8}$$

where γ is a preference parameter.

To compute the fairness score, we first estimate the target distribution of each query based on all relevant documents of the same query. That is, we count the number of documents within each group and then normalize the number as the target distribution. For simplicity purposes, we focus on two fairness categories, geographic location and gender. It not only enables us to examine intersectional fairness but also enables us to test model robustness while shifting fairness categories. Since the ideal ranking should produce a similar distribution as the target distribution, we borrow AWRF to quantify the deviation between the target distribution and system-produced distribution and to construct our fairness score. However, since AWRF is a list-wise measure, we need to convert the list-wise AWRF to a point-wise fairness score based on the Algorithm 1. Given a list of documents, it selects the next document that can maximize the AWRF score. To train our learning-to-rank models, we randomly sampled 2500 relevant and 2500 non-relevant documents for each query as our initial ranking.

3.2 Features Extraction

Since we utilize neural networks to build our re-ranker and based on previous work (Qin et al., 2021), neural networks are very sensitive to features, and data augmentation could be leveraged to improve neural model performance. Therefore, as a data augmentation method, we extracted four contextual features based on the full-text field in addition to document annotations and a document-query level feature. The contextual features reflect the similarity between the full-text embeddings and fairness categories text embeddings. They are extracted for two major purposes: (1) whether we can replace

[4] https://en.wikipedia.org/wiki/Kullback-Leibler_divergence
[5] https://en.wikipedia.org/wiki/Okapi_BM25
[6] https://github.com/castorini/pyserini

Algorithm 1: Use AWRF to obtain a fairness score at each position

input : initial ranking π_0, ideal exposure distribution $\hat{\epsilon}$, length of the output ranking n
output: training ranking π_k

initialize $\pi_k = []$;
while $i \in [1 : n]$ **do**
  for $doc \in \pi_0$ **do**
    compute AWRF($\hat{\epsilon}$, $\epsilon(\pi_k + \text{doc})$);
  $\text{doc}^* = \arg\max_{\text{doc}} \text{AWRF}(\epsilon(\pi_k + \text{doc}))$;
  nextDoc in $\pi_k = \text{doc}^*$;
  assign the AWRF score to document doc^*;
  remove doc^* from π_0;

Feature Name	Description	Level
geo	The geographic location associated with the article	Document
gen	The gender of the individual of the article	Document
rev	The popularity of the article by page reviews	Document
lan	Number of languages the article has been replicated in	Document
q-d_bm25	The BM25 score between document and query	Doc-Query
q_sim_gender	The cosine similarity between query embeddings and gender embeddings	Contextual
q_sum_geo	The cosine similarity between query embeddings and geographic location embeddings	Contextual
d_sim_gender	The cosine similarity between document embeddings and gender embeddings	Contextual
d_sim_geo	The cosine similarity between document embeddings and geographic location embeddings	Contextual

Table 1: Training features X

human annotations with NLP techniques and (2) whether contextual features as a data augmentation method could improve model performance. Finally, the contextual features are extracted using Spacy[7] to split documents into sentences, and Sentence-BERT (Reimers & Gurevych, 2019) to get embeddings. We summarized all the features we use in Table 1.

4 Result and Observation

We submitted five runs listed in Table 2. *UDInfo_F_bm25* is a baseline run without any re-rank we submitted to see the performance of our re-ranker. *UDInfo_F_lgbm2* and *UDInfo_F_lgbm2* are the runs we submitted using re-ranker based on LambdaMART by re-ranking every 20/40 documents. And *UDInfo_F_mlp2* and *UDInfo_F_mlp4* are the runs we submitted using re-ranker based on multilayer perceptron by re-ranking every 20/40 documents. Their performance is reported in Table 3 and is broken down into different fairness categories as shown in Table 4.

According to the results reported, the five runs we submitted are very similar to each other. For both fairness and relevance, we observe nuance differences between different runs. Especially compared with the baseline BM25 run, our re-rankers did not significantly improve fairness or relevance. One possible reason is that re-ranking every 20/40 documents is too preservative. Re-ranking 20/40 documents is not significant enough to impact group distribution, and thus, statistically impossible to achieve higher AWRF scores. Another possible reason is that the extracted features need to be improved, given the average accuracy of 63% of Sentence-BERT pre-trained models, leading to weak in and out-of-sample predictive power. On the other hand, our pre-processing method to incorporate fairness could be replaced by in-processing methods for better performance. Another observation is that neural models failed to outperform GBDT-based models, which aligns with the previous results (Qin et al., 2021).

[7] https://spacy.io/

Runs Name	Runs Type	Description
UDInfo_F_bm25	BM25 only	No re-rank
UDInfo_F_lgbm2	BM25 + LambdaMART	Re-rank every 20 docs
UDInfo_F_lgbm4	BM25 + LambdaMART	Re-rank every 40 docs
UDInfo_F_mlp2	BM25 + MLP	Re-rank every 20 docs
UDInfo_F_mlp4	BM25 + MLP	Re-rank every 40 docs

Table 2: Submitted Runs

Runs	nDCG	AWRF	Score	95% CI
UDInfo_F_bm25	0.5666	0.4719	0.2708	(0.236, 0.302)
UDInfo_F_lgbm2	0.5655	0.4718	0.2703	(0.235, 0.302)
UDInfo_F_lgbm4	0.5631	0.4723	0.2693	(0.235, 0.302)
UDInfo_F_mlp2	0.5645	0.4719	0.2698	(0.235, 0.302)
UDInfo_F_mlp4	0.5638	0.4719	0.2695	(0.234, 0.301)

Table 3: Runs Performance

5 Conclusion

In this year's participation in the fair ranking track, we explored two learning-to-re-rank models, LambdaMART and multilayer perceptron, for fair ranking tasks using the two-step fair ranking framework. Even though our results did not significantly outperform others, we confirmed the value of contextual features and the potential of leveraging neural models in ranking tasks. In the future, we plan to extract more training features, especially those features based on the full-text field using NLP techniques. Moreover, besides pre-processing, we will encode fairness in constraints and loss functions for our learning-to-re-rank models.

Runs	Overall	Age	Alpha	Gender	Langs	Occ	Pop	Scr-geo	Sub-geo
UDInfo_F_bm25	0.2708	0.5282	0.5606	0.5320	0.5289	0.5335	0.4810	0.5026	0.4983
UDInfo_F_lgbm2	0.2698	0.5261	0.5587	0.5304	0.5272	0.5320	0.4795	0.5006	0.4972
UDInfo_F_lgbm4	0.2693	0.5249	0.5574	0.5290	0.5263	0.5307	0.4791	0.4991	0.4962
UDInfo_F_mlp2	0.2703	0.5268	0.5597	0.5311	0.5269	0.5326	0.4803	0.5015	0.4976
UDInfo_F_mlp4	0.2695	0.5251	0.5581	0.5294	0.5250	0.5309	0.4789	0.4999	0.4965

Table 4: Model Performance w.r.t. Different Fairness Categories

References

Burges, C. J. (2010). From ranknet to lambdarank to lambdamart: An overview. *Learning*, *11*(23-581), 81.

Qin, Z., Yan, L., Zhuang, H., Tay, Y., Pasumarthi, R. K., Wang, X., ... Najork, M. (2021). Are neural rankers still outperformed by gradient boosted decision trees?

Raj, A., Wood, C., Montoly, A., & Ekstrand, M. D. (2020). Comparing fair ranking metrics. *arXiv preprint arXiv:2009.01311*.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Sapiezynski, P., Zeng, W., E Robertson, R., Mislove, A., & Wilson, C. (2019). Quantifying the impact of user attentionon fair group representation in ranked lists. In *Companion proceedings of the 2019 world wide web conference* (pp. 553–562).

Zehlike, M., Yang, K., & Stoyanovich, J. (2021). Fairness in ranking: A survey. *arXiv preprint arXiv:2103.14000*.