Fair ranking in academic search

Notebook for the TREC 2019 Fair Ranking Track

Malte Bonart
malte.bonart@th-koeln.de
Technische Hochschule Köln

ABSTRACT

This notebook summarizes our participation in the first Fair Ranking Track at TREC 2019 [2]. We shortly introduce the problem setting, give an overview of the software framework, and discuss the task and the results of our two submissions.

KEYWORDS

ranking, algorithmic fairness, learning-to-rank

1 INTRODUCTION

The issue of algorithmic fairness has recently gained popularity in the IR community [6, 9]. In this year's Fair Ranking Track, the goal was to re-rank rankings of scholarly articles from a productive academic search engine such that groups of authors are 'fairly' exposed in the sequence of rankings. The data for this challenge was taken from the *SemanticScholar* open research corpus that contains around 47 million articles [1].

It was assumed, that a fair exposure for a group of authors was satisfied if the group's g share of aggregated exposure $\mathcal{E}_g$ equals a group's share of aggregated relevance $\mathcal{R}_g$, such that

$$\mathcal{E}_g = \frac{\sum_{a\in\mathcal{A}_g}\sum_{\pi\in\Pi} e_a^\pi}{\sum_{a\in\mathcal{A}}\sum_{\pi\in\Pi} e_a^\pi} = \frac{\sum_{a\in\mathcal{A}_g}\sum_{\pi\in\Pi} r_a^\pi}{\sum_{a\in\mathcal{A}}\sum_{\pi\in\Pi} r_a^\pi} = \mathcal{R}_g, \qquad (1)$$

with $\mathcal{A}$ the set of all authors, $\mathcal{A}_g$ the set of authors in the group g, Π the set of all rankings, e_a^π the single exposure and r_a^π the individual relevance of author a in ranking π.

Individual exposure and relevance for a single ranking π and an author a was defined in the spirit of the Expected Reciprocal Rank metric [7] as

$$\begin{aligned} e_a^\pi &= \sum_{i=1}^{|\pi|}\left[\gamma^{i-1}\prod_{j=1}^{i-1}(1-f(r_{\pi_j}))\right]\mathbf{I}(\pi_i\in\mathcal{D}_a) \\ r_a^\pi &= \sum_{i=1}^{|\pi|}\left[f(r_{\pi_i})\right]\mathbf{I}(\pi_i\in\mathcal{D}_a), \end{aligned} \qquad (2)$$

where $0<\gamma<1$ a discounting factor, $f(r_{\pi_i})$ a monotonic transformation of the relevance r_{π_i} of document π_i into a probability value and $\mathcal{D}_a$ the set of documents from author a.

Note first, that an author's *relevance* metric is independent of the ranking. Second, an author's *exposure* metric at position i is independent of the relevance of the document at that specific position, but it depends on the relevance of the previous documents in the ranking. Exposure is high if the author's documents are at the beginning of the ranking or if the previously seen documents have a low relevance value.

Individual relevance and exposure metrics are summed up and normalized for each group. Unfairness for group g is then measured as the deviation from the ideal fairness condition given by (1). This approach resembles existing group fairness perspectives in the literature [3, 11, 12]. However, this setting is different from the usual fairness problems in IR, as we do not assume a particular 'protected' group but allow for various group definitions and sizes. Also, exposure and relevance are not measured for the ranked items themselves but the underlying authors of the items. Hence, ranked items can belong to several groups, and the effect of re-ranking a single item can have multiple opposing effects on the overall fairness of the system.

The system's overall unfairness Δ was measured as the Euclidean norm over each group's unfairness,

$$\Delta = \sqrt{\sum_{g\in\mathcal{G}}(\mathcal{E}_g - \mathcal{R}_g)^2}, \qquad (3)$$

and the system's overall relevance U for the users was measured as the average Reciprocal Rank metric for all rankings,

$$U = \frac{1}{|\Pi|}\sum_{\pi\in\Pi}\sum_{i=1}^{|\pi|}\left[\gamma^{i-1}\prod_{j=1}^{i-1}(1-f(r_{\pi_j}))\right]f(r_{\pi_i}). \qquad (4)$$

The system has two objectives: Minimizing group unfairness Δ for the authors of the ranked items while maximizing the users' relevance U. This can be formalized as a weighted sum optimization problem:

$$\min_{\pi_i} \quad \alpha\Delta - (1-\alpha)U \qquad (5)$$

As a further challenge, the authors of the Fair Ranking Track assumed that no information about author group relationships, $\mathcal{A}_g$ for $g\in\mathcal{G}$, was available during training time, e.g., before a ranking was submitted. The intention was that this restriction leads to systems that are robust against various changing group definitions. Therefore, direct optimization of (5) is not feasible without further knowledge of how authors are partitioned into groups.

In an initial attempt to study this problem and to provide a reference for more elaborated models, we asked: How well do already existing learning-to-rank retrieval systems and simple heuristic baselines perform in terms of fairness and relevance?

2 FRAMEWORK AND SUBMISSIONS

In the following, we describe the data pipeline and software framework that is publicly available and open-source [4]. Its modularity allows for the additional integration and evaluation of models without further effort. For this year's track, we created to submissions: First, a random shuffling of the documents in each ranking without considering further information and second, a ranking model based on the *LambdaMart* [5, 10] algorithm and several features that we constructed from the corpus.

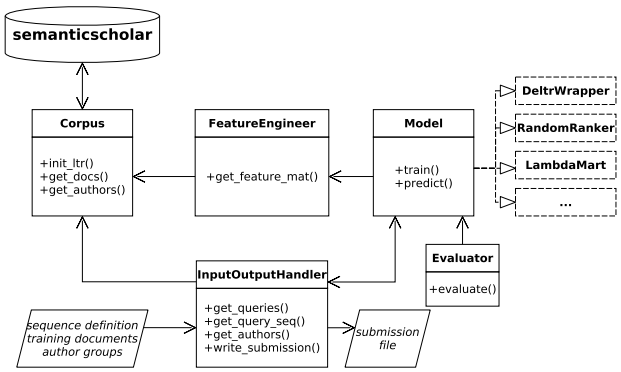

Figure 1: Class diagram that describes the software modules and their dependencies.

level	feature
query-document (BM25)	title
	abstract
	entities
	venue
	journal
	authors' names
document	year
	out-citations
	in-citations
query	query length

Table 1: Features used for the system

The *SemanticScholar* corpus has been downloaded from [1] and written into a local *Elasticsearch* database with integrated learning to rank plugin [8]. We wrote several modules that are outlined in figure 1. The *Corpus* class serves as an interface to the database. The *FeatureEngineer* class calculates a feature matrix given a query and a corresponding set of documents π. Table 1 lists the 10 features that are currently implemented.

The *Model* class serves as an interface to different model implementations, and it allows for the integration of further approaches. Given a group definition, the *Evaluator* class can be used to calculate the performance of a trained system in terms of searcher's relevance U and author's unfairness Δ. Finally, the *InputOutputHandler* class is used to handle and parse the various input files and to prepare a valid output file for submission. The system accepts a list of query sequences, a list of rankings, and a partition of authors into groups.

Some of the assessed documents were not included in the corpus and have been removed. Further, as proposed by the organizers [2], rankings with fewer than five entries were also removed. The removal resulted in a drop of 15% of the training queries and led to roughly 3800 documents with a relevance judgement.

3 RESULTS

The test files consisted of 5 sequences of query-ranking pairs. Each sequence had 25000 entries, that consist of a query and a list of corresponding unlabelled documents. The query sequences were created by sampling queries with replacement from an original test set of 635 unique queries with around 4300 corresponding documents. The specific sampling process is not known.

The submissions' performance was assessed by calculating the aggregated relevance U and the aggregated unfairness Δ based on two different author partitions: First, groups were calculated according to their productivity and impact, approximated by the h-index. And second, according to the economic level of their country of origin. Note that the group definitions were not known beforehand and were only revealed after the submission phase.

	unfairness		relevance
	Δ_{hindex}	Δ_{level}	U
min	0.0405	0.0059	0.5480
mean	0.0753	0.0435	0.6100
max	0.1110	0.0832	0.6740
random	0.0405	0.0326	0.5480
lambdamart	0.0855	0.0741	0.6600

Table 2: Aggregated test results, averaged over the five query sequences.

Table 2 summarises the aggregated performance of all submissions. For better readability, results from the five query-sequences have been averaged as we did not find any large differences between the individual sequences.

Interestingly, our randomized submission has the lowest unfairness value for the h-index partition, and a below-average unfairness for the economic level group. It also performs worst for relevance. The second submission, the *LambdaMart* algorithm, optimized users' relevance. As expected, it performed above average and achieved a relevance score close to the maximum. In terms of unfairness, this method performs above average as well but does not come close to the maximum unfairness values observed. Therefore, as this method utilizes a known learning-to-rank retrieval framework, it provides a useful baseline for both unfairness and relevance. In addition, it shows the magnitude of unfairness that current systems, which solely optimize for relevance, produce.

4 DISCUSSION

For the first Fair Ranking Track at TREC 2019, we utilized a basic learning-to-rank framework for solving this task. The main challenge was that multiple objectives (searchers' relevance and authors' group fairness) had to be optimized under the constraint of an unknown author-group-partition. This constraint seems not practicable, as one can presumably always find an arbitrary group definition that *maximizes unfairness* for a given list of rankings. Existing fairness frameworks in the literature require some group definitions during training time or, for post-processing methods, the group definitions are required for re-ranking a system's output *before* the final submission [13]. Further research should concentrate

on first, exploring the solution space of the optimization problem and analytically study the trade-offs between high fairness and high relevance. Second, robust ranking models should be implemented. Ideally, they can be trained without requiring a specific group definition, but should later handle various group definitions when making actual ranking decisions.

ACKNOWLEDGMENTS

This research was supported by the Digital Society research program funded by the Ministry of Culture and Science of the German State of North Rhine-Westphalia.

REFERENCES

[1] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the Literature Graph in Semantic Scholar. In *NAACL*. https://www.semanticscholar.org/paper/09e3cf5704bcb16e6657f6ceed70e93373a54618

[2] Asia Biega, Fernando Diaz, Michael Ekstrand, and Sebastian Kohlmeier. 2019. TREC 2019 Fair Ranking Track. https://fair-trec.github.io/

[3] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of Attention: Amortizing Individual Fairness in Rankings. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 405–414. https://doi.org/10.1145/3209978.3210063 event-place: Ann Arbor, MI, USA.

[4] Malte Bonart. 2019. irgroup/fair-trec: trec 2019 conference release. https://doi.org/10.5281/zenodo.3514668

[5] Chris J.C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview*. Technical Report MSR-TR-2010-82. https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/

[6] Carlos Castillo. 2019. Fairness and Transparency in Ranking. *ACM SIGIR Forum* 52, 2 (Jan. 2019), 64–71. https://doi.org/10.1145/3308774.3308783

[7] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*. ACM Press, Hong Kong, China, 621. https://doi.org/10.1145/1645953.1646033

[8] OpenSource Connections. 2019. o19s/elasticsearch-learning-to-rank. https://github.com/o19s/elasticsearch-learning-to-rank original-date: 2016-12-25T03:19:01Z.

[9] Michael D. Ekstrand, Robin Burke, and Fernando Diaz. 2019. Fairness and Discrimination in Retrieval and Recommendation *(SIGIR'19)*. ACM, New York, NY, USA, 1403–1404. https://doi.org/10.1145/3331184.3331380

[10] Jerry Ma. 2019. jma127/pyltr. https://github.com/jma127/pyltr original-date: 2015-08-17T05:42:11Z.

[11] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18* (2018), 2219–2228. https://doi.org/10.1145/3219819.3220088 arXiv: 1802.07281.

[12] Meike Zehlike, Gina-Theresa Diehn, and Carlos Castillo. 2018. Reducing Disparate Exposure in Ranking: A Learning To Rank Approach. *arXiv:1805.08716 [cs]* (May 2018). http://arxiv.org/abs/1805.08716

[13] Meike Zehlike, Tom Sühr, Carlos Castillo, and Ivan Kitanovski. 2019. FairSearch: A Tool For Fairness in Ranked Search Results. *arXiv:1905.13134 [cs]* (May 2019). http://arxiv.org/abs/1905.13134 arXiv: 1905.13134.