University of Glasgow Terrier Team and Naver Labs Europe at TREC 2019 Fair Ranking Track

Graham McDonald
University of Glasgow
Glasgow, Scotland, UK
graham.mcdonald@glasgow.ac.uk

Thibaut Thonet
NAVER LABS Europe
Meylan, France
thibaut.thonet@naverlabs.com

Iadh Ounis
University of Glasgow
Glasgow, Scotland, UK
iadh.ounis@glasgow.ac.uk

Jean-Michel Renders
NAVER LABS Europe
Meylan, France
jean-michel.renders@naverlabs.com

Craig Macdonald
University of Glasgow
Glasgow, Scotland, UK
craig.macdonald@glasgow.ac.uk

ABSTRACT

In our participation to the TREC 2019 Fair Ranking Track, the University of Glasgow Terrier Team and Naver Labs Europe joined forces to investigate (1) a novel probabilistic retrieval strategy that maximises the utility of the ranking, (2) two greedy brute-force re-ranking approaches that build on our novel probabilistic retrieval strategy and enforce individual fairness before adopting a particular trade-off between the utility and the fairness of the ranking, and (3) two approaches that deploy search results diversification as a fairness component to diversify over multiple possible dimensions of the task's unknown author groupings.

1 INTRODUCTION

For the TREC 2019 Fair Ranking Track, the University of Glasgow Terrier Team and Naver Labs Europe collaborated in making a joint submission. In particular, the Terrier Team aimed to investigate how to build upon their Terrier.org Information Retrieval (IR) platform [6, 7] and how to tailor search results diversification [10, 11] into the fairness component of the fair ranking task. Naver Labs Europe experimented with an approach for fair ranking based on individual fairness so as to alleviate the non-disclosure of protected groups. The approach is inspired by the method introduced in [3] by accounting for the group exposure in an amortised fashion.

Overall, in this year's TREC Fair Ranking Track, the main ideas we wanted to test and evaluate in a realistic setting were:

- We postulated that an efficient and robust strategy to obtain fair rankings across any potential definition of the protected groups is to ensure fairness at the individual (author) level.
- Since achieving an optimal relevance-fairness trade-off in the rankings involves finding a solution in a huge combinatorial space, we introduce an on-line scalable approximation of this problem, mixing brute-force enumeration and fine-tuned prefiltering.
- We postulated that it could be possible to cast the fair ranking problem as a search results diversification approach.
- We experimented with two possible search results diversification approaches tailored to the collection used by the Fair Ranking Track. In particular, we investigated diversifying across the authors' influence (estimated by citations) and/or the journals' preponderance in the corpus.

In the following, we describe our document indexing process. Next, we present our proposed approaches in the TREC Fair Ranking Track, followed by an analysis of our obtained results.

2 INDEXING & RETRIEVAL

We indexed the semantic scholar [2] corpus using Terrier v5.2. We transformed the JSON representation of the corpus into traditional TREC documents, where each JSON attribute is represented as a separate field in the TREC document, before creating multiple indexes of the collection, one for each of the used combinations of configurations: Stopword removal On or Off and; Stemming On or Off. We used Terrier's standard stopword list for stopword removal and we used Terrier's implementation of Porter stemmer [8]. When indexing the collection, we recorded positional information and 'TITLE', 'PAPERABSTRACT' and 'OTHER' fields, where 'OTHER' contains the text from all of the remaining fields in a TREC document. This index was used in all our submitted runs.

After indexing the collection, we investigated a number of retrieval strategies to form an initial ranking that serves as a base ranking before the application of the fairness components that we evaluate. For our base ranking, we evaluated two retrieval models from the literature, namely BM25 [9] and the DPH [4] parameter free document weighting model from the Divergence from Randomness (DFR) framework [1]. We also evaluated the effectiveness of both of these models for our initial ranking with various query expansion (QE) settings using Terrier's Bo1 Divergence from Randomness approach.

We found that DPH performed better than BM25 in our initial experiments. Moreover, in general, we found that QE only improved the retrieval performance for a small minority of QE settings. Therefore, we opted for using DPH (without QE) as our base ranking for all our runs described in Section 4. On the other hand, for the runs described in Section 3, the retrieval strategy builds on the DPH base ranking (with QE) by introducing a greedy component that is optimised for either maximising utility or minimising unfairness.

3 NAVER LABS EUROPE APPROACHES FOR FAIRNESS

The Naver Labs Europe approach to fairness is based on the two following assumptions:

- Given that the protected groups are undisclosed and that the final objective is to reach a good utility-fairness trade-off over all potential group definitions, we enforce fairness at the individual level (i.e., every single author is considered as a singleton group) so as to be robust with respect to any possible grouping. This is in line with some work in "fair classification", asking for equalised fairness across any sub-population that one might define [5].
- As we observed that the overlap of authors across queries is limited, we consider amortisation (used to compute the cumulative utility and unfairness) for each unique query independently, i.e., over the repeated occurrences of the same query. In other words, we can solve the problem at the query level and do not have to consider the whole set of queries.

Consequently, we propose an on-line single-query greedy brute-force re-ranking (SGBR) approach that enforces individual fairness. We derive two runs from SGBR, namely `uognleSgbrFair` and `uognleSgbrUtil`, that emphasise fairness and utility respectively.

Our proposed SGBR approach operates in two steps: First, we pre-order the documents in order to position the most promising documents (i.e., the documents which would yield the largest gain in both utility and fairness based on previous rankings) at the top. Second, we apply a brute-force strategy to the pre-ordered documents by scoring all permutations of the top documents while keeping the tail documents fixed. These two steps are further detailed in the remainder of this section.

3.1 Pre-ordering

For each instance of a query in the sequence, the documents are initially pre-ordered based on a scoring function that linearly combines (1) the accumulated (estimated) relevance of the documents' authors and (2) the accumulated discrepancy in the deserved exposure for these authors. The deserved exposure of an author with respect to a query corresponds to the exposure the author would have merited based on their relevance to the query. Such discrepancy results from the rankings that were output for the previous occurrences of the same query, which might not have been able to provide the exact exposure the authors deserved. Given that the rankings are computed in a greedy/online way (i.e., the rankings that are output for the previous query instances in the sequence are kept fixed) this pre-ordering procedure helps to compensate for the exposure that was granted in previous query instances and alleviates under/overexposure.

3.2 Brute-force Re-ranking

Next, SGBR operates in a brute-force fashion on the top pre-ordered documents (obtained as detailed in Section 3.1): it scores every permutation of the top n documents from the pre-ordered list while keeping the remaining documents in their position. The scoring is based on an estimate of the track's official measure, which linearly combines utility U (here, derived from the estimated relevance scores[1]) and unfairness Δ (here, considered at the individual/author level) as $U - \lambda\,\Delta$, where λ is a hyperparameter. The ranking that obtains the best combined score is retained and is output for the query; the ranking's computed utility and unfairness are stored for amortisation in the future instances of this query in the sequence.

[1]Note that the track's metrics rely on a probabilistic relevance score. For that reason, we calibrated the relevance scores prior to using them in utility and unfairness.

4 GLASGOW DIVERSITY APPROACHES FOR FAIRNESS

Building on the Divergence from Randomness (DFR) rankings that we discussed in Section 2, we investigate whether it is possible to cast the fair ranking problem as a search results diversification task. The first approach that we investigate is to diversify the ranking to cover authors at different stages of their careers: e.g., early career researchers vs. highly experienced researchers, whose influence will be different. Our second approach aims to give a fair exposure to different paper sources across the corpus (e.g., different journals and proceedings will have a different preponderance across the corpus). Our submitted runs to the TREC 2019 Fair Ranking Track involve instantiating the author's influence and/or the source exposure as the aspects on which to diversify the ranking of documents to increase the fairness of the ranking.

In the remainder of this section, we describe the two specific approaches that we use for enhancing fairness. We leverage Glasgow's xQuAD [12] approach to implement the diversification.

4.1 Author Influence

Intuitively this approach aims to reduce the preponderance of a given set of authors at the top of the ranking for a given query (e.g., reducing Bruce Croft's occurrence as an author of the top ranked papers for the query "Information Retrieval"). Hence, prolific and perhaps experienced authors will not overwhelm other authors in the ranking. To operationalise this idea, we adapt the xQuAD approach by considering single authors as the aspects that we aim to optimise. In this respect, xQuAD aims to cover as many authors as possible in the top of the ranking (coverage), while penalising authors who already appeared in the top ranks (novelty). For scoring the documents to a given author (aspect) in xQuAD, we use the author's citation count as the relevance score. We denote this fairness diversification approach as `uognleDivAAsp` in Section 5.

4.2 Author and Journal Exposure

Our second diversification approach aims to enhance fairness by combining two ideas: (1) Papers that are written by early career authors should have the same chance of being ranked in the top rank positions as those by more experienced authors with high citation counts. (2) Papers from the popular venues (journals, proceedings, etc.) should not overwhelm the ranking; instead, we aim to give a fair exposure to all papers regardless of their publication venue.

To operationalise this idea, we leverage again xQuAD assuming that the papers have two aspects and their scores are generated as follows : (i) for a given paper, we calculate the listed authors' average citation count; next we bin the papers into 15 bins depending on the paper's mean authors citation count (ii) for the same paper, we also calculate the journal's exposure within the collection, which is estimated by the count of papers from that venue within the collection. We denote this approach in Section 5 as `uognleDivAJc`.

5 SUBMITTED RUNS

We submitted five runs to the TREC Fair Ranking Track:

- `uognleMaxUtil`: This run linearly combines the DPH ranking (with Bo1 query expansion) with a simple LM model (with Dirichlet smoothing), restricted on the TITLE field. No fairness is explicitly targeted. All hyperparameters (Dirichlet smoothing constant and combination weight) were fine-tuned by cross-validation on the training set.
- `uognleSgbrFair`: This run extends the `uognleMaxUtil` run by enforcing fairness through the application of the greedy brute-force re-ranking approach described in Section 3. Overall, this run balances fairness and utility in its ranking scoring scheme by setting the utility/unfairness weighting hyperparameter $\lambda = 1$.
- `uognleSgbrUtil`: This run extends the `uognleMaxUtil` run by enforcing fairness through the application of the greedy brute-force re-ranking approach described in Section 3. This run puts more emphasis on utility than `uognleSgbrFair` by setting $\lambda = 0.3$. Note that due to an error in the submission, the JSON file corresponding to `uognleSgbrFair` was submitted in lieu of the correct `uognleSgbrUtil` run. For that reason, we report in Section 6 the correct results for this run (obtained through the test evaluation scripts provided by the organizers) in addition to the official, incorrect ones.
- `uognleDivAAsp`: This run builds upon the DFR base ranking approach described in Section 2, i.e., no calibration or query expansion is applied. The diversification approach described in Section 4.1 is deployed as the fairness component for the final ranking. The relevance / diversification (fairness) trade-off is varied as queries are repeated.
- `uognleDivAJc`: This run builds upon the DFR base ranking approach described in Section 2, i.e., no calibration or query expansion is applied. The diversification approach described in Section 4.2 is deployed as the fairness component for the final ranking. The relevance / diversification (fairness) trade-off is varied as queries are repeated.

6 RESULTS

In this section, we provide a concise analysis of the performance of our 5 submitted runs as reported by the official track metrics, *utility* and *unfairness*, over the two group membership definitions that the TREC 2019 Fair Ranking Track evaluated: *h-index*, which assigns the authors to four groups based on an author's h-index; and *Level*, which assigns authors to two groups based on the IMF economic development status of an author's country of affiliation.

Table 1 presents the performance of each of our runs in terms of utility (higher is better) for each of the 5 query sequences, compared to the *best*, *mean* and *worst* scores achieved by all of the TREC 2019 participating groups. As can be seen from Table 1, `uognleMaxUtil` was our best performing approach in terms of utility for each of the five query sequences. Moreover, this approach achieved the highest utility score achieved by any of the TREC Fair Ranking Track participating teams for all of the query sequences.

Table 2 presents the results of our submitted runs in terms of minimising unfairness, with respect to the *h-index* author grouping (lower scores are better). In terms of unfairness, for this author grouping, our diversification approach `uognleDivAJc` from Section 4.2 achieved the lowest unfairness scores (the best performing run on this measure) for all of the query sequences, compared to our four other submitted runs. Note that our diversification approach from Section 4.1 was less effective than `uognleDivAJc` at minimising unfairness with respect to the *h-index* author grouping.

Table 3 presents the results of our submitted runs in terms of minimising unfairness, but this time with respect to the *level* author grouping (again, lower scores are better). For this author grouping, our diversification approach `uognleDivAAsp` actually achieved the best unfairness scores for all of the query sequences, compared to our other four submitted runs. Moreover, `uognleDivAAsp` was the best performing run, in terms of unfairness, compared to all of the runs submitted to the TREC 2019 Fair Ranking track.

In practice, we have to decide for a particular compromise between the utility and fairness criteria. Assuming that this trade-off can be materialised by a weighted combination of both dimensions (Utility - λ Unfairness), Figures 1a and 1b show how our different methods compare to each other (performance measures were averaged over the 5 query sequences). For the *h-index* author grouping, three methods dominate in the Pareto sense: `uognleMaxUtil` for relatively low values of λ ([0,3.4]), `uognleSgbrFair` for medium values of λ ([3.4,18]) and `uognleDivAJc` for very high values of λ ([18,∞]). For the *level* author grouping, only two methods dominate: `uognleMaxUtil` for low values of λ ([0,1.5]) and `uognleDivAAsp` for medium to high values of λ ([1.5,∞]). Of course, it is hard to generalize to any possible definition of the protected groups, but we can already draw some insights from this first wave of experiments:

- At this stage, the choice of the optimal strategy is still very sensitive to the grouping definition and to the choice of the trade-off point between relevance and fairness; no single method clearly outperforms the others.
- With methods that try to explicitly optimize amortized exposure (such as `uognleSgbrFair` and `uognleSgbrUtil`), it is clear from the definitions of exposure and fairness that we need to have a relatively precise measure of relevance probability to be sure that the amortised exposures are distributed in a fair way. In other words, having an accurate calibrated relevance score is a prerequisite for ensuring fairness (in its current definition). This explains why `uognleMaxUtil` (which does not take fairness into account) might still be a good choice. Relying on a poor (or biased) estimate of relevance to derive a measure of unfairness and to compensate for it could actually deteriorate the fairness, instead of enhancing it.

7 CONCLUSIONS

In our participation to the TREC 2019 Fair Ranking Track, we initiated a collaboration between the University of Glasgow and Naver Labs Europe. Our participation revolved around two new ideas: (1) an approach for fair ranking based on individual fairness so as to ensure robustness across any protected group that one might define; (2) the casting of the fair ranking task as a diversification problem, by tailoring Glasgow's xQuAD diversification approach to create fairer rankings towards authors and publication venues. We

Table 1: Run results in terms of utility (higher is better). The corrected run is indicated with a star. Best values are highlighted in bold.

run	query sequence					mean	std
	0	1	2	3	4		
uognleDivAAsp	0.556890	0.563662	0.559796	0.562195	0.563329	0.561174	0.002834
uognleDivAJc	0.551952	0.555744	0.554115	0.553229	0.557159	0.554440	0.002053
uognleMaxUtil	**0.675142**	**0.673084**	**0.673399**	**0.673256**	**0.675538**	**0.674084**	0.001161
uognleSgbrFair	0.614921	0.614873	0.615033	0.614744	0.615791	0.615072	0.000415
uognleSgbrUtil	0.614921	0.614873	0.615033	0.614744	0.615791	0.615072	0.000415
uognleSgbrUtil*	0.626195	0.626420	0.626958	0.626134	0.627279	0.626597	0.000501
TREC Best	0.675142	0.673084	0.673399	0.673256	0.675538	0.674084	0.001161
TREC Mean	0.608679	0.609166	0.609620	0.609211	0.611033	0.609542	0.000898
TREC Worst	0.545806	0.545560	0.550262	0.545997	0.550128	0.547551	0.002419

Table 2: Run results in terms of unfairness, with respect to the *h-index* grouping (lower is better). The corrected run is indicated with a star. Best values are highlighted in bold.

run	query sequence					mean	std
	0	1	2	3	4		
uognleDivAAsp	0.060567	0.057001	0.058163	0.059342	0.057216	0.058458	0.001498
uognleDivAJc	**0.045589**	**0.044121**	**0.045479**	**0.044977**	**0.044462**	**0.044926**	0.000635
uognleMaxUtil	0.067043	0.068092	0.061793	0.065033	0.066209	0.065634	0.002422
uognleSgbrFair	0.047388	0.049657	0.049119	0.045755	0.049058	0.048195	0.001608
uognleSgbrUtil	0.047388	0.049657	0.049119	0.045755	0.049058	0.048195	0.001608
uognleSgbrUtil*	0.053779	0.055183	0.053906	0.051620	0.054589	0.053815	0.001351
TREC Best	0.038984	0.039868	0.043182	0.039830	0.040659	0.040505	0.001610
TREC Mean	0.074779	0.076113	0.074923	0.075145	0.075360	0.075264	0.000523
TREC Worst	0.109511	0.113056	0.110704	0.111900	0.110680	0.111170	0.001351

Table 3: Run results in terms of unfairness, with respect to the *level* grouping (lower is better). The corrected run is indicated with a star. Best values are highlighted in bold.

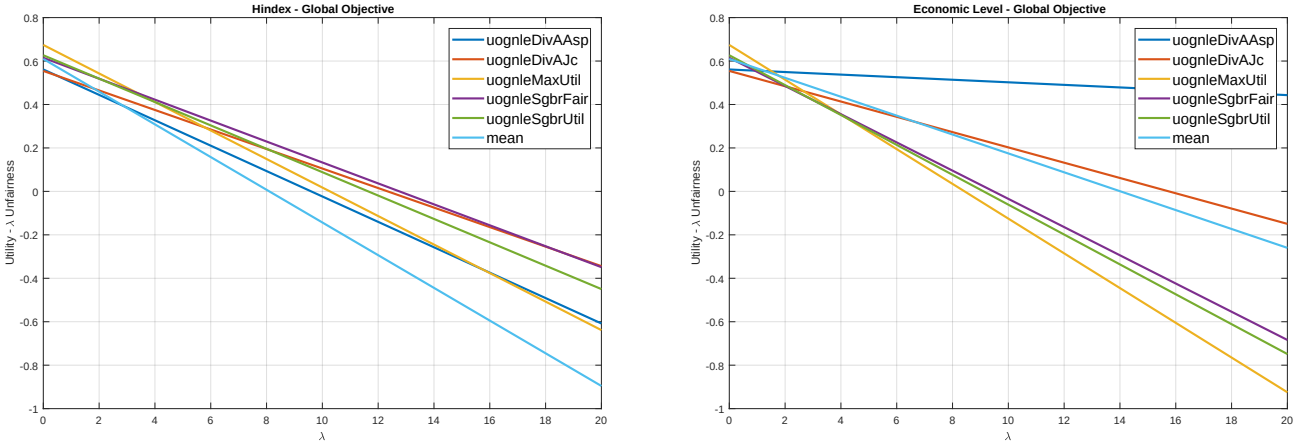
run	query sequence					mean	std
	0	1	2	3	4		
uognleDivAAsp	**0.005296**	**0.005004**	**0.005872**	**0.006809**	**0.006536**	**0.005903**	0.000774
uognleDivAJc	0.033390	0.034778	0.032304	0.038458	0.037053	0.035197	0.002544
uognleMaxUtil	0.077489	0.077477	0.080568	0.077632	0.086557	0.079945	0.003924
uognleSgbrFair	0.061774	0.065647	0.063986	0.063208	0.070127	0.064948	0.003215
uognleSgbrUtil	0.061774	0.065647	0.063986	0.063208	0.070127	0.064948	0.003215
uognleSgbrUtil*	0.066040	0.068790	0.067522	0.066020	0.075460	0.068766	0.003916
TREC Best	0.005296	0.005004	0.005872	0.006809	0.006536	0.005903	0.000774
TREC Mean	0.041605	0.043898	0.044432	0.042603	0.044838	0.043475	0.001343
TREC Worst	0.077489	0.088616	0.085493	0.077632	0.086557	0.083157	0.005231

showed that our diversification approaches were particularly effective at minimising unfairness, while our amortisation approaches were effective at maximising utility. In particular, our participation produced the runs with the least unfairness and the maximum utility among all TREC Fair Ranking Track participants.

REFERENCES

[1] Gianni Amati. 2003. *Probabilistic Models for Information Retrieval based on Divergence from Randomness.* Ph.D. Dissertation. Department of Computing Science, University of Glasgow.

[2] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu



(a) h-index grouping **(b) Level grouping**

Figure 1: Run results for different combinations of utility and unfairness, by varying the weighting hyperparameter λ. The results for `uognleSgbrUtil` correspond here to the corrected run.

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 *Proc. of NAACL*.

[3] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of Attention: Amortizing Individual Fairness in Rankings. In *Proc. of SIGIR*.

[4] Ben He, Craig Macdonald, Iadh Ounis, Jie Peng, and Rodrygo L Santos. 2008. University of Glasgow at TREC 2008: Experiments in blog, enterprise, and relevance feedback tracks with Terrier. In *Proc. TREC 2008*.

[5] Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. 2018. Meritocratic Fairness for Infinite and Contextual Bandits. In *In Proc. AIES*.

[6] Craig Macdonald, Richard McCreadie, RL Santos, and Iadh Ounis. 2012. From puppy to maturity: Experiences in developing terrier. In *Proc. of OSIR at SIGIR*.

[7] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Christina Lioma. 2006. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proc. OSIR at SIGIR*.

[8] Martin F Porter et al. 1980. An algorithm for suffix stripping. *Program* 14, 3 (1980), 130–137.

[9] Stephen Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proc. SIGIR*.

[10] Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *Proc. of WWW*.

[11] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2015. Search Result Diversification. *Found. Trends Inf. Retr.* 9, 1 (2015), 1–90.

[12] Rodrygo L. T. Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. 2010. Explicit Search Result Diversification through Sub-queries. In *Proc. ECIR*.