University of Glasgow Terrier Team at the TREC 2020 Fair Ranking Track

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

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

ABSTRACT

In our participation to the TREC 2020 Fair Ranking Track, the University of Glasgow Terrier Team investigated a new approach for organically uncovering latent communities of authors that we wish to be fair to. Our deployed approach leverages a co-embedding model to jointly model a document's attributes, such as the document's authors, and the citation link graph of the documents in a collection, within a single embedding space. This network co-embedding is then used as input to a community detection approach that automatically updates the identified communities for each instance of a repeated query. Moreover, we experiment with two different ranking strategies to provide a fair exposure to different communities, and the authors within the communities, over time. Our first ranking strategy is inspired by the concepts of coverage and novelty from search results diversification, while our second ranking strategy leverages a data fusion approach for prioritising different communities over time.

1 INTRODUCTION

Building on last year's participation, for the TREC 2020 Fair Ranking Track re-ranking task, the University of Glasgow Terrier Team aimed to build upon their Terrier.org Information Retrieval (IR) platform [7, 10] to investigate another approach for organically uncovering the latent communities of authors that we wish to be fair to.

Our approach jointly models a document's attributes, such as co-authorship, and the document collection's citation link graph within a single embedding space, before leveraging a community detection approach to organically uncover latent communities of authors. Such latent author communities are likely to, for example, work on a particular problem or within a particular field of research. Moreover, our approach automatically updates the membership of the identified communities for each instance of a repeated query to try to ensure fairness for *unknown*, or arbitrary, protected groups.

To evaluate the effectiveness of our community detection approach for generating fair rankings, we experiment with two distinct ranking strategies. Our proposed strategies aim to provide a fair exposure to authors within a particular community, and to provide a fair exposure to each of the communities over time. The first ranking strategy that we deploy is inspired by search results diversification [12] while the second ranking strategy that we deploy leverages a well-known data fusion [3] approach to prioritise different communities over time.

In the following, we first describe our community detection fairness approach. Next we discuss the two ranking strategies that we deploy, before discussing the relevance-focused component of our approaches and providing details of our submitted runs. We then present an analysis of our obtained results.

2 COMMUNITY DETECTION FOR FAIR RANKING

Our participation in the Fair Ranking Track 2020 builds upon a novel community detection approach for identifying groups of documents that share common attributes, such as authors, and that are related to each other, for example through citations. Our intuition is that to be fair to unknown, or arbitrary, groups of authors when generating rankings we need a way of automatically uncovering the latent author groups, or communities, that exist within the collection of documents that are to be ranked.

Our deployed community detection approach for fair ranking constructs a co-embedding attribute network [9] to jointly model the documents' attributes and the links that exist between the documents in the collection. Our approach has three stages, as follows:

- **Stage 1**: Constructs a directed network multigraph representing the document collection where a node in the graph is a document in the collection, represented by a set of attributes of the document. The edges of the graph are the citation links between documents.
- **Stage 2**: Learns a co-embedding of the constructed graph, which jointly models the documents' attributes and the links between documents within a single embedding space.
- **Stage 3**: Deploys a clustering-based community detection approach to identify communities within the learned embeddings.

For our participation in the Fair Ranking Track 2020, in Stage 1 of our community detection approach, we leverage a document's authors as attributes for the nodes of the graph and citation links as the graph edges. We note however, that the proposed approach is not restricted to modelling nodes and links in such a way. For example, the document's attributes can easily be updated to also include the terms of the document.

3 FAIR RANKING STRATEGIES

We experiment with two strategies for generating fair rankings that aim to provide a fair exposure to our organically identified latent communities, and the authors within those communities, over time. Both of our approaches combine a *community fairness* score with a relevance score (we will discuss our approach for ranking the documents with respect to relevance in Section 4) for each document on a query-by-query basis to provide a fair exposure over multiple repeated queries.

3.1 Ranking Approach 1: Community Representativeness

Our first fair ranking strategy is loosely inspired by search results diversification [12]. Specifically, we take inspiration from the search results diversification concepts of *novelty* (that selects documents to include in a ranking if they discuss sub-topics of an ambiguous query that are not yet discussed by the documents that have previously been selected for the ranking) and *coverage* (where documents that represent many potential sub-topics of an ambiguous query are promoted in the ranking).

However, differently from search results diversification, we aim to promote documents in the ranking that are (a) highly *representative* of the community that they belong to and, (b) highly dis-similar to the documents that are in the other communities. Importantly, for this approach, we consider that each document is a member of a single community and we generate new communities (varying the size and membership of the communities) for each instance of a repeated query, to try to ensure a fair exposure for communities and authors over time.

Our community representativeness fair ranking strategy combines the scores of two components. The first component, namely *Community Coverage*, scores documents based on their similarity to the other documents within the document's own community. The Community Coverage component prioritises the documents that are the most representative of the community that the documents belong to. The second component, namely *Community Novelty*, scores a document based on its dis-similarity to the documents that are members of the communities that the current document does not belong to. This component promotes documents if they are different from the documents that are in the *other* communities, i.e., the communities that the candidate document does not belong to.

For our experiments, we deploy two variants of our community representativeness fair ranking strategy to combine the community-based scores with the predicted relevance of a document to generate the final ranking for a particular instance of a query. We provide details of the different variants that we deploy in Section 5.

3.2 Ranking Approach 2: Data Fusion for Community Prioritisation

For our second community-based fair ranking strategy, we leverage a data fusion (rank aggregation) technique, CombSUM [3], that has previously been shown to be an effective component for diversification [6]. To leverage data fusion to provide a fair exposure to individual communities, we assign prioritisation scores to each of the identified communities and generate multiple rankings. Each of the generated rankings prioritises the documents in a particular community, based on the community's assigned prioritisation score. The generated rankings are then aggregated to construct the final ranking.

For each instance of a repeated query, our data fusion community prioritisation ranking strategy has the following four stages:

- **Stage 1**: Define n communities, where each document in the collection of documents that is to be ranked is a member of exactly one community.
- **Stage 2**: Deploy a stochastic process to assign a priority score to each of the communities, so that the communities are ranked 1..n.
- **Stage 3**: Generate n rankings of documents where the rank score for a document is calculated as the document's relevance score multiplied by the priority score of the community that the document is a member of.
- **Stage 4**: Deploy a data fusion technique to aggregate the rank scores of each of the n rankings to generate the final ranking for the query instance.

4 INDEXING & RETRIEVAL

We indexed the semantic scholar [2] corpus using Terrier [7, 10] 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 indexing the collection. When indexing the collection, we recorded the 'TITLE', 'PAPERABSTRACT' and 'OTHER' fields, where 'OTHER' contains the text from all of the remaining fields in a TREC document. We removed stopwords using Terrier's standard stopword list and applied Terrier's implementation of Porter stemmer [11]. This index was used in all our TREC submitted runs.

We investigated a number of retrieval strategies to form a base ranking that provides the document relevance scores for each of our community detection fair ranking approaches. We found that combining the DPH [4] parameter-free document weighting model from the Divergence from Randomness (DFR) framework [1] with the PyTerrier [8] implementation of ColBERT [5] resulted in the most effective retrieval performance. We use the relevance scores from this configuration for four of our five submitted runs. As a comparison benchmark, to evaluate the relative improvements that are the result of integrating ColBERT into the retrieval process, we also submitted a run where the relevance scores are generated using DPH alone.

5 SUBMITTED RUNS

We submitted five runs to the TREC 2020 Fair Ranking Track:

- **UoGTrBComRel**: This run is a linear combination of the DPH+ColBERT relevance scores from PyTerrier and the (dis)similarity scores from each of the two components of our community representativeness fair ranking strategy, as follows: *Relevance+Community Coverage+Community Novelty*. The relevance component and both of the components of our community representativeness fair ranking strategy are weighted equally in this variant. In practice, this means that, for documents that have a relatively high relevance score, the relevance component in this variant dominates the community representativeness components. This ensures that documents that are predicted to be relevant remain close to the top of the final ranking.
- **UoGTrBComPro**: This run combines the DPH+ColBERT relevance scores from PyTerrier with the (dis)similarity scores from each of the two components of our community representativeness fair ranking strategy. Differently from *UoGTrBComRel*, this run linearly combines a document's relevance score with the product of the (dis)similarity scores from each

Table 1: Run results in terms of the Mean (per-query) difference in expected exposure from the target exposure over 200 queries. The table also shows the variance (Var) and standard deviation (Std.) of each of our approaches and the TREC Min, Mean, Median and Max. The best Mean score is highlighted in bold.

	Mean	Var	Std.
UoGTrBComRel	0.7966	0.1973	0.4442
UoGTrBComPro	0.8087	0.1971	0.4439
UoGTrBComFu	0.6078	0.1053	0.3245
UoGTrBRel	0.8251	0.2045	0.4523
UoGTrComRel	0.7966	0.1973	0.4442
TREC (per-query) Min	**0.2931**	0.0329	0.1814
TREC (per-query) Mean	0.7566	0.0653	0.2555
TREC (per-query) Median	0.7147	0.0789	0.2809
TREC (per-query) Max	1.4304	0.1649	0.4062

of the two components of our community representativeness fair ranking strategy, as follows: *Relevance*+(*Community Coverage*∗*Community Novelty*). This approach puts relatively more emphasis on the scores generated by the community representativeness ranking strategy. In particular, the more a document is dissimilar to the documents that are in other communities, the more the document's *own community representativeness* score (*Community Coverage*) is boosted. This can promote a document higher up a ranking than is likely to happen in the *UoGTrBComRel* variant. If a document is completely similar to the documents that are in other communities, then this approach reduces to a linear combination of the document's relevance score plus the document's *own community* representativeness score.

- **UoGTrBComFu**: This run is a linear combination of the DPH+ColBERT relevance scores from PyTerrier with the scores from our data fusion community prioritisation ranking strategy. The run deploys data fusion to promote different communities over time.
- **UoGTrBRel**: This run simply consists in ranking, for each instance of a query in the sequence, the documents according to their DPH+ColBERT relevance scores from PyTerrier. No fairness component is explicitly enforced.
- **UoGTrComRel**: This run is the same as *UoGTrBComRel*, except that the documents' relevance scores are generated by DPH alone (i.e., there is no ColBERT integration deployed when generating the relevance scores for this run).

6 RESULTS

In this section, we provide a concise analysis of the performances of our five submitted runs as reported by the official track metric, the difference in *expected exposure* from the *target exposure*. For the fairness ground-truth, each document is assigned to one of three fairness groups based on the International Monetary Fund (IMF) economic development status of the country of each author's affiliation, either *advanced*, *developing*, or *mixed*.

Table 2: Per-query analysis. Number of queries (out of 200 queries) for which each approach achieved the *lowest* difference in expected exposure from the target exposure (i.e. the best performing system for a particular query) (= Min_{ee}), less than the mean difference (< $Mean_{ee}$), less than the median difference (< $Median_{ee}$) and less than the max difference (< Max_{ee}) in expected exposure from the target exposure. Best values are highlighted in bold.

	$= Min_{ee}$	$< Mean_{ee}$	$< Median_{ee}$	$< Max_{ee}$
UoGTrBComRel	11	71	87	194
UoGTrBComPro	10	67	83	193
UoGTrBComFu	**15**	**140**	**155**	**200**
UoGTrBRel	8	66	80	191
UoGTrComRel	2	82	92	196

Table 1 presents the mean difference in expected exposure and target exposure for each of our submitted runs, averaged over 200 queries. Table 1 also presents the mean values of the per-query TREC min, TREC mean, TREC median and TREC max differences in expected and target exposures (lower scores are better). From Table 1, we can see that *UoGTrBComFu* achieves a smaller difference in expected and target exposures than the TREC mean and TREC Median (mean scores over the 200 queries).

Table 2 presents how each of our approaches performed on a per-query basis, in terms on the number of queries where an approach achieved the smallest difference in expected and target exposures (denoted as = Min_{ee}), less than the TREC mean difference (denoted as < $Mean_{ee}$), less than the TREC median difference (denoted as < $Median_{ee}$) or less than the TREC Max difference (denoted as < Max_{ee}). We can see from Table 2 that our data fusion community prioritisation ranking strategy (*UoGTrBComFu*) performed best from all of the runs submitted to TREC for 15 of the 200 queries (7.5%). Moreover, this approach resulted in a smaller difference between expected and target exposure than the TREC Mean for 70% of the queries (140) and a smaller difference than the TREC Median for 77.5% of the queries (155). Furthermore, our *UoGTrBComFu* performed better than at least one of the other systems submitted to TREC for all of the queries. Overall, these results are promising and we will further investigate developing our proposed approach as future work.

Turning our attention to the performance of our *community representativeness* fair ranking strategy, we can see from Table 1 that, in terms of mean performance over the 200 queries, our *UoGTrBComRel* variant (a linear combination of relevance and the (dis)similarity scores from our community representativeness fair ranking strategy), performs better than our *UoGTrBComPro* variant (which puts relatively more weight on the scores generated by the community representativeness components than the relevance component). Indeed, from Table 2, we can see that *UoGTrBComRel* performs better than *UoGTrBComPro* in terms of = Min_{ee}, < $Mean_{ee}$, < $Median_{ee}$ and < Max_{ee}. This shows that relevance is an important component for minimising the difference between expected and target exposures when deploying our community representativeness ranking strategies.

7 CONCLUSIONS

For our participation to the TREC 2020 Fair Ranking Track, we investigated a new approach for organically uncovering latent communities of authors that we wish to be fair to. Our approach firstly jointly models, within a single embedding space, a document's attributes (e.g., the document's authors) along with the citation link graph of the document collection. Secondly, our approach leverages a community detection approach for organically generating author communities. We also experimented with two different fair ranking strategies to provide a fair exposure to our identified author communities over time. Our first ranking strategy is inspired by search results diversification while our second ranking strategy leverages a data fusion technique. We found that our data fusion-based ranking strategy was particularly effective for providing a fair exposure to authors over time, resulting in the least difference between the expected and target exposures for 7.5% of queries, better than the TREC Mean for 70% of queries and better than the TREC Median for 77.5% of queries.

8 ACKNOWLEDGEMENTS

We would like to thank Alberto Ueda, Craig Macdonald and Zaiqiao Meng for their inputs during the course of our participation.

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 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] Edward A Fox and Joseph A Shaw. 1994. Combination of multiple searches. *NIST special publication SP* 243 (1994).

[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] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proc. of SIGIR*.

[6] Shangsong Liang, Zhaochun Ren, and Maarten de Rijke. 2014. Fusion Helps Diversification. In *Proc. of SIGIR*.

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

[8] Craig Macdonald and Nicola Tonellotto. 2020. Declarative experimentation in information retrieval using pyterrier. In *Proc. of ICTIR*.

[9] Zaiqiao Meng, Shangsong Liang, Hongyan Bao, and Xiangliang Zhang. 2019. Co-embedding Attributed Networks. In *Proc. WSDM*.

[10] 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*.

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

[12] Rodrygo LT Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. 2010. Explicit search result diversification through sub-queries. In *Proc. of ECIR*.