Overview of the TREC 2020 Fair Ranking Track*

Asia J. Biega
Microsoft Research Montréal
asia.biega@acm.org

Fernando Diaz
Montreal Institute for Learning Algorithms
diazf@acm.org

Michael D. Ekstrand
Boise State University
michaelekstrand@boisestate.edu

Sergey Feldman
Allen Institute for Artificial Intelligence
sergey@allenai.org

Sebastian Kohlmeier
Allen Institute for Artificial Intelligence
sebastiank@allenai.org

For 2020, we again adopted an academic search task, where we have a corpus of academic article abstracts and queries submitted to a production academic search engine. The central goal of the Fair Ranking track is to provide *fair exposure* to different groups of authors (a *group fairness* framing). We recognize that there may be multiple group definitions (e.g. based on demographics, stature, topic) and hoped for the systems to be robust to these. We expected participants to develop systems that optimize for fairness and relevance for arbitrary group definitions, and did not reveal the exact group definitions until *after* the evaluation runs were submitted.

The track contains two tasks, *reranking* and *retrieval*, with a shared evaluation.

Rerank runs sorted a query-dependent list of documents to simultaneously provide fairness and relevance.

Retrieval runs returned 100-item rankings from the corpus in response to a query string.

The track organizers provided a sequence of queries, each accompanied by a varying-size set of documents. Both tasks used the same queries; participants were asked not to use the test queries' rerank sets as a component of their retrieval model training.

1 Protocol

For our fair ranking evaluation, we provided participants with a sequence $\mathcal{Q}$ of queries accompanied by unordered sets of documents to rank. The document sets are of varying size. For each request (query q and set of documents $\mathcal{D}_q$), participants provided a ranked list of the documents from $\mathcal{D}_q$. For the retrieval task, q is the set of all documents in our corpus and participants were asked to return a fixed set of documents. The final system output is a sequence of rankings for each query. Algorithm 1 presents a pseudocode of the evaluation protocol.

The rankings produced in response to queries in the sequence were to balance two goals: (1) be relevant to the consumers and (2) be fair to the producers.

*Data and code are available at: https://fair-trec.github.io/2020/

Algorithm 1 Evaluation protocol

$\forall q, \mathcal{D}_q \in \mathcal{Q}, \Pi_q \leftarrow \{\}$
for $q, \mathcal{D}_q \in \mathcal{Q}$ **do**
  $\pi \leftarrow$ System$(q, \mathcal{D}_q)$
  $\Pi_q \leftarrow \Pi_q \cup \{\pi\}$
end for
return $\{\Pi_q\}$

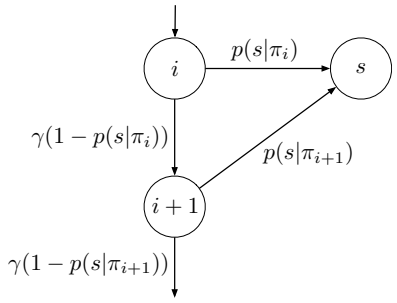

Figure 1: Attention model.

2 Evaluation

Unlike previous TREC tracks, systems were to return multiple rankings for each query, as they might in response to different impressions of the same query text. At evaluation time, we measured measure *expected exposure* of groups over rankings produced for each given query [1].

Given a sequence of queries $\mathcal{Q}$ and associated system rankings, we evaluated systems according to fair exposure of authors and relevance of documents.

2.1 Measuring Author Exposure for a Single Query

In order to measure exposure, we adopt the browsing model underlying the Expected Reciprocal Rank metric [**?**]. Given a static ranking π in response to a query impression, the exposure of author a is,

$$e_a^\pi = \sum_{i=1}^{n} \left[\gamma^{i-1} \prod_{j=1}^{i-1} (1 - p(s|\pi_j)) \right] I(\pi_i \in \mathcal{D}_a)$$

n	number of documents in ranking π
$\mathcal{D}_a$	documents including a as an author
π_i	document at position i
γ	continuation probability (fixed to 0 for the final position in the ranking)
$p(s\|d)$	probability of stopping given user examined d

We present a graphical depiction of this model in Figure 1.

We used a discounting factor $\gamma = 0.5$, and assumed $p(s|d) = f(r_d)$, where r_d is the relevance of the document d and f is a monotonic transform of that relevance into a probability of being satisfied.

In order to compute the *expected exposure* for a, we consider the set of all rankings presented by the

system for that query Π_q,

$$e_a = \sum_{\pi \in \Pi_q} e_a^\pi \tag{1}$$

The *target expected exposure* for a query is derived from Equation 1 assuming a policy that randomizes amongst all permutations whose relevance monotonically degrades with rank [1].

2.2 Measuring Group Exposure for a Single Query

Assume that each author is assigned to exactly one of $|\mathcal{G}|$ groups. Let $\mathcal{A}_g$ be the set of all authors in group g. The group expected exposure is defined as,

$$\mathcal{E}_g = \sum_{a \in \mathcal{A}_g} e_a \tag{2}$$

We define the target group expected exposure $\mathcal{E}_g^*$ as Equation 2 using the individual target expected exposure (Section 2.1).

2.3 Expected Exposure Metric

We evaluated systems using the per-query difference in system group expected exposure and target group expected exposure,

$$\Delta_{\mathcal{G}}(\mathcal{E}) = \left(\sum_{g \in \mathcal{G}} (\mathcal{E}_g - \mathcal{E}_g^*)^2 \right)^{\frac{1}{2}} \tag{3}$$

We averaged per-query metrics to compute the summary metric for the run.

3 Data

3.1 Input

Three main inputs were made available to participants: the *corpus* of articles to search, the *example group definition* file to help them develop and test their solutions, and the *queries*.

3.1.1 Paper and Author Data

The paper and author metadata CSV files provide summary information for papers and their authors in the rerank set. There are three files:

paper_metadata.csv contains basic paper information: ID, title, year, venue, and the number of citations.

author_metadata.csv contains author information: ID, name, citation count, paper count, and H-index.

authors_for_papers.csv contains the author list for each paper: paper ID, author ID, and position.

These files do *not* contain abstracts. Creating a usable index requires the corpus in the next section.

3.1.2 Corpus

The full corpus for this track was the Semantic Scholar (S2) Open Corpus from the Allen Institute for Artificial Intelligence. It can be downloaded from `http://api.semanticscholar.org/corpus/`, and consists of 186 1GB data files. Each file is compressed JSON, where each line is a JSON object describing one paper. The following data are available for most papers:

- S2 Paper ID
- DOI
- Title
- Abstract
- Authors (resolved to author IDs)
- Inbound and outbound citations (resolved to S2 paper IDs)

We provide tools for subsetting the corpus at `https://github.com/fair-trec/fair-trec-tools`. These tools were used to create the subset we released to participants.

3.1.3 Example Group Definition Data

For training, we provided the file `fair-TREC-sample-author-groups.csv` containing group ids for authors in the S2 corpus. This group definition was not our final group definition, but was intended to help groups get started on the task.

This CSV file contains two columns:

1. The `author` column has the S2 ID of the author.
2. The `gid` column has the author's group identifier.

3.1.4 Queries

Query data. The query data was obtained from searches that occurred on the Semantic Scholar[1] website between Feb 14, 2020 and April 27, 2020. The data consisted of session id, query text, result papers from the first 3 pages (30 results), and result clicks. Sessions with more than 25 unique queries were excluded, after which only sessions with at least 1 result paper click and no more than 250 result paper clicks were included.

Query-document relevance. We estimated the relevance of different documents to queries based on the click data described above. We computed the query-document relevance as a weighted average of the number of clicks on a given document over all impressions of a given query-document pairs present in the data. For weighting, we used ranking position propensity scores estimated by the Semantic Scholar from their system data. Relevance scores were converted to binary based on a manually selected threshold.

Query filtering. Because of the exhaustive annotation process that required annotating group memberships of all document authors, we then sampled a smaller number of queries to construct evaluation sequences. We released 200 training and 200 evaluation queries. For both the training and evaluation data, these queries were selected first by random sampling, and then by a number of filtering steps. More specifically,

- To help remove known-item queries, we included only queries with at least two relevant documents and excluded queries with more than 4 words.
- We further manually cleaned the sample to remove any known-item queries, queries containing people's names, and queries with offensive and sensitive keywords.

[1] `https://www.semanticscholar.org`

Query sequences. Since the evaluation this year focused on individual queries, each query sequence consisted of repetitions of a single query. We had 200 sequences, each consisting of a 100 repetitions of a query.

3.2 Output

For each query sequence, participants submitted a JSON file where each line is a JSON object (a dictionary) containing their ranking results:

- <sequence id>.<query number in sequence> ('q_num')
- <query id> (to look up in query file) ('qid')
- An ordered list of document IDs (of the documents to be re-ranked for the query) ('ranking')

3.3 Annotations

NIST assessors annotated returned papers with the country in which each author was operating (based on their affiliation data in the paper manuscript), along with institution type (academic, industry, nonprofit, government, etc.). Not all papers were able to be annotated. These are the known reasons a paper may not have annotations:

- It has a large author list (> 10). We excluded such long papers because there were not very many of them, and large-team papers require special treatment in how we consider their author lists, particularly when authors may be from different groups.
- Some papers did not have an accessible source with sufficient affiliation information to provide annotations (e.g. no available PDF file, and a paper information page that either did not contain affiliation details or was not accessible from the annotation interface).
- Some papers may not provide sufficient information to determine an author's affiliation location.

All documents in the candidate sets for the rerank tests were annotated, along with many of the documents in the corpus for the retrieval task.

	Overall	Eval Candidates
Documents	—	4,693
Annotated Documents	4,381	2,114
Have Country Data	4,160	2,008
Advanced Econ Papers	3,374	1,609
Developing Econ Papers	543	272
Mixed Econ Papers	243	127
Advanced Econ Authors	10,679	5,250
Developing Econ Authors	2,317	1,187

Table 1: Annotation outcome summary.

Table 1 shows a summary of the collected annotation data, after merging and integrating data sources. For these statistics, to aggregate each paper's authors into a single economic designation for the paper, we considered a paper to be from an advanced or developing economy if all authors' locations had the same economic designation; otherwise, we list it as a 'mixed' economy paper.

run	$\Delta_{\mathcal{G}}$
NLE_META_9_1	0.428
NLE_META_99_1	0.429
NLE_META_PKL	0.433
NLE_TEXT_9_1	0.438
NLE_TEXT_99_1	0.442
UoGTrBComFu	0.475
LM-rel-groups	0.580
LM-relevance	0.601
MacEwan-base	0.722
UoGTrComRel	0.798
LM-relev-year	0.811
UoGTrBComRel	0.832
MacEwan-norm	0.850
UoGTrBComPro	0.851
UW_bm25	0.875
UoGTrBRel	0.886
UW_Kr_r60g20c20	0.895
umd_relfair_ltr	0.907
UW_Kr_r25g25c50	0.916
UW_Kr_r0g0c100	0.948
UW_Kr_r0g100c0	0.999
LM-rel-year-100	1.046
Deltr-gammas	1.067

Table 2: Reranking results. Runs ordered in increasing expected exposure (Equation 3). *Smaller values are better.*

3.4 Group Definitions

Group definition accompanying the training data. To help participants get started, we provided a file containing group membership definitions for authors in the S2 corpus. This definition was based on author h-indices. This definition was not used in the final evaluation, but was meant as a starting point for system development. For each author, the data consisted of:

- the author's S2 ID,
- the author's group identifier.

Authors were split into 2 groups, based on the value of their h-index.

Group definitions for evaluation. Our primary evaluation was based on the NIST assessors' country annotations. We combined these annotations with economic development levels from the International Monetary Fund. With this definition, the fairness target is to ensure fair exposure for papers written in countries with more- and less-developed economies. The evaluation itself uses individual author-level annotations; the exposure a mixed-economy paper receives counts towards both developing and advanced economy exposure. Under this definition, authors are split into two groups.

run	$\Delta_{\mathcal{G}}$
UW_t_bm25	0.748
UW_Kt_r80g10c10	0.769
UW_Kt_r60g20c20	0.770
UW_Kt_r25g25c50	0.821
UW_Kt_r0g0c100	1.056

Table 3: Retrieval results. Runs ordered in increasing expected exposure (Equation 3). *Smaller values are better.*

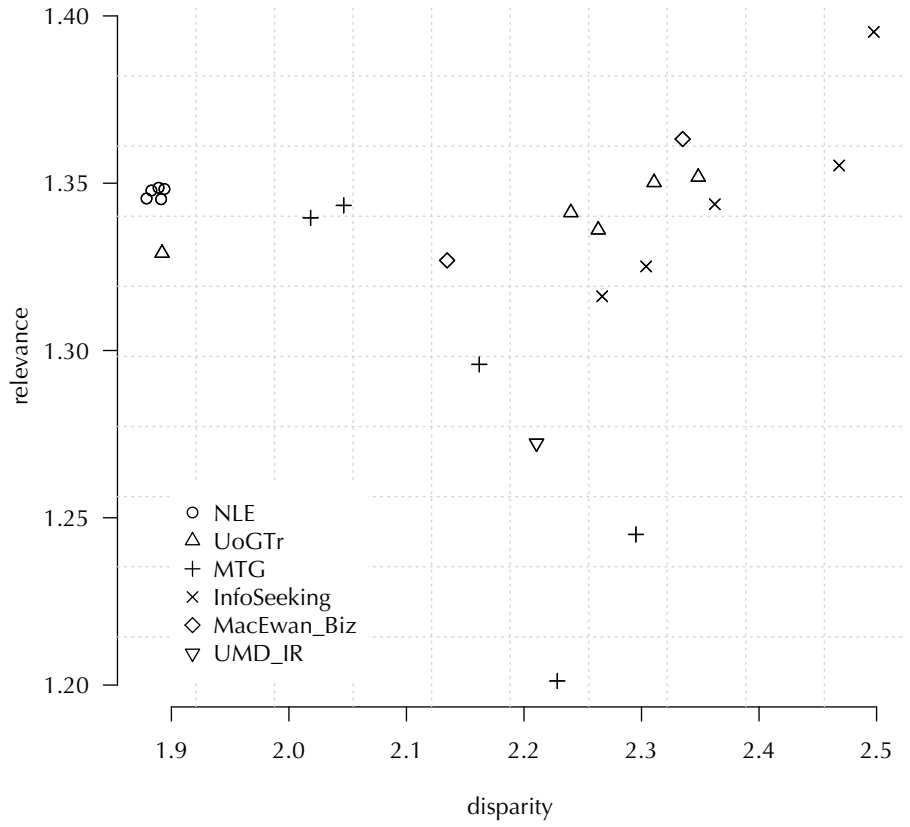

Figure 2: Disparity and relevance results for the reranking task. Lower disparity values are better. Higher relevance values are better.

4 Results

4.1 Submitted runs

This year, 6 different teams submitted a total of 28 runs, (23 runs for the reranking task and 5 runs for the retrieval task). Some of the approaches included:

- weighted reranking methods that optimized the KL-divergence of the output group distributions to target group distributions estimated using external Google Scholar data (team InfoSeeking),
- ranking fusion methods where the documents were ranked by the BM25 scores of queries matched to different document parts (title, abstract) and the individual ranking weights shift throughout a query sequence (team MacEwan),
- including authors from all groups at the top of the ranking with the groups determined by a clustering algorithm on the authorship graph; an approach that randomizes the output of a learning-to-rank algorithm based on the predicted relevance and optionally includes the publication year as a feature (team MTG),
- randomization of the outputs of rankings based on the textual content of documents and externally trained word embeddings, with an optional parametrized readjustments to match a target group exposure distribution (team NLE),
- a static method that does keep track of exposure in between rankings based on a learning-to-rank algorithm with a custom objective balancing fairness and relevance (team UMD),
- a two-stage approach where the first stage is based on standard retrieval methods, and the second stage uses reranking based on membership in authorship communities detected using graph embedding methods (team UoGTr).

Notably, novel ideas as compared to the last year's runs included estimating group membership using automatically detected authorship communities, randomization of the outputs, including the publication year as a feature, and incorporation of external resources (Google Scholar data and word embeddings trained on the Bing corpus).

4.2 Evaluation

We present the results for reranking and retrieval in Tables 2 and 3, sorted by $\Delta_{\mathcal{G}}$. Although the run descriptions were not sufficient to draw many general conclusions, the top runs all used external resources (e.g. public embeddings) and multiple permutations per query (e.g. amortization or randomization).

We can decompose $\Delta_{\mathcal{G}}$ into relevance and disparity components [1],

$$\text{disparity} = \sum_{g \in \mathcal{G}} \mathcal{E}_g^2 \tag{4}$$

$$\text{relevance} = \sum_{g \in \mathcal{G}} \mathcal{E}_g \times \mathcal{E}_g^* \tag{5}$$

This allows us to plot each run on disparity-relevance axes which often reflects a trade-off between disparity and relevance. We present results in Figure 2. In general, we would like runs to lie close to the top left corner. Although the top performing runs from NLE had relatively high relevance, the strong $\Delta_{\mathcal{G}}$ was more attributable to exhibiting less disparity.

References

[1] F. Diaz, B. Mitra, M. D. Ekstrand, A. J. Biega, and B. Carterette. Evaluating stochastic rankings with expected exposure. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 275–284, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3411962. URL https://doi.org/10.1145/3340531.3411962.