The University of Amsterdam at the TREC 2021 Fair Ranking Track

Ali Vardasbi, Gabriel Bénédict, Shashank Gupta, Maria Heuss
Pooya Khandel, Ming Li, Fatemeh Sarvi
University of Amsterdam
Amsterdam, The Netherlands
{a.vardasbi,g.benedict,s.gupta2,m.c.heuss,p.khandel,m.li,f.sarvi}@uva.nl

ABSTRACT

TREC 2021 fair ranking track is composed of two tasks, intended to help WikiProject coordinators, with a static ranking of 1000 pages (Task1), as well as WikiPedia editors, with a stochastic ranking of 50 pages (Task2). The rankings should be fair with respect to geographical locations as well as an undisclosed demographic attribute. We have used a lexical matching method to detect two demographic attributes, namely gender and sexuality. For the static ranker of Task1, we tried a method based on swapping the items that result in largest marginal gain for the relevance-fairness compound objective. We have seen that the greedy pairwise swapping method leads to better results compared to other variants. For the probabilistic ranker of Task2, we used two sampling approaches. The first one is based on an existing control-theory inspired algorithm, and it leads to the best overall performance among our submissions. The second approach uses a two-stage Plackett-Luce and achieves very good disparity performance (in terms of EE-D), but overall, its performance is dominated by the first approach due to its low ranking performance.

1 INTRODUCTION

The TREC 2021 Fair Ranking Track aims at helping Wikipedia editors involved in the WikiProject to find articles that need editing via fair recommendations: provide exposure to Wikipedia articles related to protected groups, otherwise underrepresented in the Wikipedia corpus.

The corpus consists of a subset of Wikipedia pages and a set of queries, and there are two main tasks defined for this year. The goal of each task is to rank documents for a given query (a WikiProject) so that the ranking is: (1) relevant to the query, and (2) fair to articles that represent a protected demographics or attributes.

Our methods for the TREC 2021 Fair Ranking Track are:

- Enriching fairness features by adding some demographic attributes representing gender and sexuality of each Wikipedia page.
- A pairwise and a listwise approach for maximizing the objective function for Task1, based on swapping items that result in largest marginal gain for the objective.
- Developing a probabilistic ranker that samples rankings from a probability distribution for Task2 based on (1) control-theory inspired from [4]; and (2) a two-stage Plackett-Luce approach.

Among all our submissions, for the Task1, we achieve our best results using the greedy pairwise swapping approach. For Task2 we achieve the best disparity score (measured by EE-D) with one of the two stage Plackett-Luce approaches, but the best relevance score (measured by EE-R) and overall score (EE-L) with one of the control-theory inspired rankers.

2 BUILDING BLOCKS

2.1 Ranking Features

We extract two types of features from the dataset: Term-based features and BERT-based features.

We consider *Corpus* and *Topics* that contain document and query information to extract term-based features. Specifically, we found that using *key_words* column from topics as representative of query information along with *text* column from corpus as representative of document information for extracting term-based features would be effective for our tasks. To this end, we extract all features based on term and document frequency in addition to BM25 score similar to feature extraction of MSLR [7] dataset.

For the BERT-based features, we use the pre-trained BERT model[1] to generate embeddings for query and document separately. After that, we compute the semantic relevance feature between the query and document via simple dot product between the query embedding vector and the document embedding vector.

2.2 Relevance Estimation

Our submissions are based on either post-processing an unfair ranker, or using the relevance probability of each page. For both cases, we need a calibrated ranker which does not consider the sensitive features. We have tested the following learning to rank (LTR) algorithms:

(1) LambdaMART [1]: A well-known pair-wise approach that is usually among the top performing LTR methods. It directly optimizes the ranking evaluation metric, i.e., normalized discounted cumulative gain (NDCG), by including the difference of NDCG caused by swapping each pair of items in a list, while leaving other items fixed at their position. We used the LightGBM implementation [3].

TREC 2021, November 17–19, 2021, Gaithersburg, Md. USA

[1] https://huggingface.co/bert-base-uncased

(2) ListNet [2]: A high performing list-wise approach that uses softmax to compute the distribution of a permutation.
(3) Pointwise RMSE: A 5-layer neural network (NN) with root mean-square error (RMSE) loss.
(4) Logistic regression: we use Scikit-Learn's [6] logistic regression module with default settings and a sag solver.

Using 5-fold validation on the training data, we select the LambdaMART model and calibrate its output scores using sigmoid calibration.

2.3 Demographic Attributes

Wikipedia is used as a reference to discover demographic attributes that are observable in the population. Each demographic attribute (e.g. gender) has its corresponding groups (male, female, non-binary). We considered the following demographic attributes: gender, sexuality, ethnicity, religion.

Ethnicity and religion were sparse in the data and ethnicity can take a lot of forms in Wikipedia and revolve around an American point of view. We thus only kept gender and sexuality.

The textual content of the articles in the given Wikipedia corpus is used to determine allocation to these groups. The first step is to determine whether that person is or has been a living human (as opposed to fictional characters). The former are characterized by having a reported birth and eventually death year. We thus filtered out any data not containing a sentence of the sort : "1954 births", "1999 deaths". Any article that is referred to bellow, thus refers to an article about a *real* person.

Next, gender is determined by searching for women ["female", "woman", "women"], men ["male", "man", "men", "pharaoh", "military", "duke", "king", "prince "] and non-binaries [" trans ", "queer", "non-binary", "transgender"]. The Wikipedia page on non-binary genders is used as a reference[2]. If terms of different groups were present, we attributed a unique group using the following order of precedence: non-binary, female, male. If none of the terms above could be found, it is considered an unknown gender.

Regarding sexuality, the reductive binary classification heterosexual VS LGBTQ+ is used. The reference to LGBTQ+ terms are rare in Wikipedia (only 13437 in our corpus of $4M$+), it seemed reasonable to aggregate them: ["LGBT", "homosexual", "bisexual", "queer", "lesbian", "gay "]. By opposition, any article that does not contain these words is assigned the heterosexual category.

3 TASK1: WIKIPROJECT COORDINATORS

In the first task, the ranker should produce a static ranked list of 1000 pages. The ranked list should be fair in terms of attention-weighted rank fairness (AWRF) [8] and the submissions are ranked based on their nDCG × AWRF. Since we do not have access to the relevance scores, nor to all the fairness attributes, we have to estimate both nDCG and AWRF for the test set. As a proxy for nDCG, we compute the DCG of the list using the calibrated outputs of our (unfair) ranker (Sec. 2.2). For the AWRF computation, we use the geographic tags together with the extracted demographic attributes (Sec. 2.3). We set the objective of our re-ranker to maximize the multiplication of these two proxy values.

To maximize the objective, we tried two approaches: pairwise and listwise.

3.1 Pairwise Swaps

Starting from the unfair ranking which maximizes the DCG but does not consider AWRF, we iteratively swap items which result in the largest marginal gain for the objective. We continue until the largest marginal gain falls below a threshold. This approach is inspired by the greedy maximization of submodular functions.

3.2 Listwise Updates

We also tried listwise score updates for maximizing the DCG × AWRF. In this approach, at each iteration, the score of each item is updated with respect to the aggregated gain in the objective caused by swapping that item with other items in the list. In other words, for each item i, we measure the objective gain caused by swapping i with items $j \neq i$ and use the average of these values as the aggregated gain of item i. Then, we update the score of item i with a constant fraction of this aggregated gain (similar to updates in gradient descent using a constant learning rate). The scores are initialized by the calibrated output of our (unfair) ranker.

4 TASK2: WIKIPEDIA EDITORS

In the second task, a stochastic ranker should produce a sequence of lists of 50 pages. The ranker should consider both relevance and work-needed for each page, while being fair with respect to geographic tags and an undisclosed demographic attribute. We tried two different approaches for this task: (1) Control-theory inspired probabilistic ranker; and (2) Two-stage Plackett-Luce ranker. In this section we explain these two approaches.

4.1 Control-Theory Inspired Probabilistic Ranker

For this ranker we took inspiration from [4], where the authors use a control-theory inspired ranker. In each sampling step, the over/under representation of item groups is measured based on a weighted sum of the predicted relevance score. This orders the items into the next ranking. We use a Plackett-Luce model [5] which draws from a probability distribution given by this linear combination of score and advantage.

Let D be the collection of items, that we want to rank for a given query q. We use a position based user model that assumes that the expected exposure of a document d in a ranking π is purely dependent on it's rank $\mathcal{E}(d \mid \pi) = \frac{1}{log_2(1+rank(d|\pi))}$. In this approach we use only the geographical attribute to define groups, using the Plackett-Luce sampling to achieve some level of individual fairness as well. The expected exposure of the items of each geographic group G can be determined by the sum of the expected exposure of all it's items:

$$\mathcal{E}(G \mid \pi) = \sum_{d \in G} \mathcal{E}(d \mid \pi) \tag{1}$$

Given a sequence $\Pi = \{\pi_i\}$ of sampled rankings that we show to the users, the total expected exposure of G can be calculated by aggregating the expected exposure of this group in each individual

[2] https://en.wikipedia.org/wiki/Non-binary_gender

ranking:

$$\mathcal{E}(G \mid \Pi) = \sum_{\pi_i \in \Pi} \mathcal{E}(G \mid \pi_i) \tag{2}$$

To determine the target exposure we work with a mix of disparate treatment and demographic parity. The total exposure that will be (in expectation) available for one top-k ranking is given by $A = \sum_{i=1}^{k} \frac{1}{log_2(i)}$. We first calculate the merit $M(G)$ of group G as the sum of the relevance scores r_d of the items in the group and get:

$$\mathcal{E}^*_{\text{disp}}(G) = \frac{\sum_{d \in G} r_d}{\sum_{d \in D} r_d} \cdot A \tag{3}$$

As relevance scores r_d we use the product of the work-needed feature and the relevance score predicted with LambdaMART 2.2. To get the final target exposure, we average this with share of the total exposure that the group should get based on the proportion of group G in the total population, pop_G, to get as final target exposure for group G:

$$\mathcal{E}^*(G) = \frac{\mathcal{E}^*_{\text{disp}}(G) + \text{pop}_G \cdot A}{2} \tag{4}$$

The target exposure for a sequence of t rankings is given by $\mathcal{E}^*_t(G) = t \cdot \mathcal{E}^*(G)$. Note that the target exposure is calculated based on relevance labels that have previously been estimated, which means that what we call target exposure here is merely an estimation of the actual target exposure that is unknown to us.

Now for sampling a ranking, we start with calculating the aggregated expected exposure that each item group has collected over the course of all previously sampled rankings. Based on the estimated target exposure $\mathcal{E}^*_{t-1}(G)$ and the aggregated exposure $\mathcal{E}_{t-1}(G)$ of each group G we now calculate the advantage as

$$A_t(G) = \left(\mathcal{E}_{t-1}(G) - \mathcal{E}^*_{t-1}(G)\right)^2 \operatorname{sign}\left(\mathcal{E}_{t-1}(G) - \mathcal{E}^*_{t-1}(G)\right). \tag{5}$$

Combining the Advantage $A_t(G)$ with the originally estimated relevance score r_d, an adjusted score is calculated as

$$h_{d,t} = \min(0, \theta r_d - (1 - \theta) A_t(G)). \tag{6}$$

for some $\theta \in [0, 1]$. These adjusted scores are now used to define a Plackett-Luce model from which we iteratively sample items to put in our ranked list. The goal of this method is that, by adding some randomization, our method will provide a more fair representation for groups that we do not consider explicitly.

4.2 Two-stage Plackett-Luce Ranker

Inspired by the Plackett-Luce used in individual fairness, we use a simple two-stage Plackett-Luce based ranker to take both group fairness and work-needed into consideration.

For relevance evaluation, we trained a logistic regression model to get the relevance probability of available documents. Given a query q, we calculate the relevance probability p of every document in the collection via the model. We only focus on the documents with relatively high relevance probability, since the number of candidate documents is large and the non-relevant documents would not contribute the relevance evaluation. We consider the document with a probability higher than a threshold v as relevant, v is a hyper-parameter.

Table 1: Performance of our submissions for Task1.

	nDCG	AWRF	Score
1step_pair_list	0.082	0.691	0.062
1step_pair	**0.084**	**0.694**	**0.065**
2step_pair_list	0.079	0.691	0.061
2step_pair	0.082	0.694	0.064
average median			0.111
average min			0.002
average max			0.199

The first ranking stage ensures group fairness. Given the number of possible relevant items C_g of each geographical group and the world geographical population information, we then generate the target distribution $\mathcal{D}^*$ across groups as follows:

$$D\,(g)_{relevance} = \frac{C_g}{\sum_{g \in G} C_g} \tag{7}$$

$$\mathcal{D}^*\,(g) = \frac{D\,(g)_{relevance} + D\,(g)_{population}}{2} \tag{8}$$

where $D_{relevance}$ is the relevant documents distribution over different groups; $D_{population}$ is the population distribution over different groups.

We aim to add a constraint at this stage to help with group fairness. Specifically, we generate a candidate pool which is associated with the target group distribution. Given the target distribution, we can derive the desired number of documents for each group:

$$c_g = L \cdot \mathcal{D}^*(g)$$

where L is the required list length.

Within each group, we use PL sampling to select c_g candidate documents according to the relevance probability. Since 100 ranking are desired, we repeat this generation process to get 100 candidate pools to avoid that the 100 rankings only consist of the same group of documents. Note that this repeated process naturally helps with the individual fairness. The candidate pools will be used in the second stage.

The second stage accounts for the work needed factor. For each candidate pool, we multiply the work-needed factor w and the relevance probability of documents in this candidate pool and get the score:

$$s_{wp} = p \cdot w$$

These scores s_{wp} will be used to define the second level PL model and generate a rank list.

5 RESULTS AND DISCUSSION

5.1 Results

Table 1 contains the average performance of our submissions for Task1. There are four submissions for Task1 as follows:

(1) 1step_pair_list: For each test query, we re-ranked by both pairwise (Sec. 3.1) and (Sec. 3.2) approaches and selected the output with the highest objective.
(2) 1step_pair: Only the pairwise approach.
(3) 2step_pair_list: Similar to 1step_pair_list, but on the output of 1step_pair_list, instead of the unfair initial ranker.

Table 2: Performance of our submissions for Task2.

	EE-D	EE-R	EE-L
PL_control_0.6	3.273	**8.809**	**15.501**
PL_control_0.8	3.254	8.665	15.77
PL_control_0.92	3.148	8.48	16.034
PL_IRlab_0.5	**1.402**	4.933	21.383
PL_IRlab_0.7	1.532	5.278	20.821
average median			20.635
average min			13.072
average max			30.086

(4) 2step_pair: Similar to 1step_pair, but on the output of 1step_pair_list, instead of the unfair initial ranker.

Among the above four submissions, the 1step_pair reaches the highest nDCG and AWRF score, though the difference between different submissions seems to be insignificant.

Table 2 shows the results for Task2. For the control theory inspired probabilistic ranker we experiment with different values for the hyperparameter θ. The results of the experiments with values $\theta = 0.6, 0.8, 0.92$ can be found in the rows with name PL_control_θ. Among these submissions PL_control_0.92 reaches the best (lowest) EE-D score, and PL_control_0.6 reaches the highest and hence best value for EE-R, and the total score, EE-L. For two-stage Plackett-Luce Ranker, we experiment with different values of relevance threshold $v = 0.5, 0.7$, the corresponding results can be found in rows with name PL_IRLab_v.

5.2 Discussion

Looking at the Task1 results in Table 1, we observe that the nDCG of all the four submissions is below the average median nDCG × AWRF. Since AWRF is always between $[0, 1]$, this means the main weak point of our submissions for Task1 is its initial ranker. We hypothesize the problem with our initial ranker could be on the non-calibrated scores of the tree-based methods. To elaborate, we choose the ranker based on its ranking performance, but for Task1 we relied on our proxy estimations of DCG from the output scores of the ranker. An accurate estimation of DCG is only possible if the output of the ranker is calibrated. But as the output of our initial ranker is not calibrated, our DCG estimations are far from being accurate. For example, we observe the range of scores for each query lies in a small range, causing our re-rank approaches to treat liberally in swapping relevant and non-relevant items.

The results of the second task show that for the control theory inspired approach the lowest value of θ gives the best results in terms of EE-R, while the highest value of θ gives the best results for EE-D. This seems counter intuitive to us since we expected to see a trade off between EE-D and EE-R in the opposite direction. According to our assumptions, lower values for θ should favor groups with little exposure so far, leading to lower EE-R and higher fairness measured by EE-D. On the other hand all these scores lie fairly close together, when comparing them to the range of scores in the other submissions and might not be significant.

Compared with the controlled PL method, the two-stage Plackett-Luce method is able to generate several rankings in parallel. However, we identify two limitations of this method. (1) The logistic regression might be too weak to find the most relevant items, when comparing to modern LTR methods. (2) Because of the randomness of PL sampling, this method might perform better with a large number of ranking settings, however only 100 rankings are needed in Task2.

6 CONCLUSION

In this TREC track we have submitted nine runs, four for Task1 and five for Task2. We used LambdaMART as the initial (unfair) ranker and built our fair rankers on top of that. Though LambdaMART was chosen by cross-validation on the training queries, detailed results on the test queries show that for some queries a simple lexical matching method (such as BM25) would have performed better. It seems in the extremely low data regime of this track, one should use an ensemble of rankers and assign each query to one ranker based on the confidence or other similar measures.

Among our four runs for Task1, 1step_pair reaches the highest nDCG and AWRF score. The difference between different runs is not noticeable. The main weak point of our runs is its low ranking performance, as the average nDCG scores of all four runs falls below the average median nDCG × AWRF scores.

For Task2, our control-theory inspired method achieves the best overall result ($EE_L = 15.50$), very close to the average minimum score of 13.07. It has to be noted that the average minimum score is a lower bound for the best performance and may differ from it, as the minimum score may come from different runs for different queries. At the time of writing this report we do not have access to the results of other teams. Our two-stage Plackett-Luce method achieves very good disparity scores (EE-D), but due to its bad relevance score, the overall performance is worse than the average median. One reason for this could be the use of logistic regression-based ranker instead of LambdaMART.

REFERENCES

[1] Christopher J.C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview.* Technical Report MSR-TR-2010-82. Microsoft.

[2] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning.* 129–136.

[3] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems.* 3146–3154.

[4] Till Kletti and Jean-Michel Renders. [n.d.]. Naver Labs Europe at TREC 2020 Fair Ranking Track. ([n. d.]).

[5] Harrie Oosterhuis. 2021. Computationally Efficient Optimization of Plackett-Luce Ranking Models for Relevance and Fairness. *arXiv preprint arXiv:2105.00855* (2021).

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[7] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. *CoRR* (2013). http://arxiv.org/abs/1306.2597

[8] Piotr Sapiezynski, Wesley Zeng, Ronald E Robertson, Alan Mislove, and Christo Wilson. 2019. Quantifying the Impact of User Attentionon Fair Group Representation in Ranked Lists. In *Companion Proceedings of The 2019 World Wide Web Conference.* 553–562.