FairER: Entity Resolution With Fairness Constraints
Vasilis Efthymiou, Kostas Stefanidis, Evaggelia Pitoura, Vassilis Christophides

To cite this version:
Vasilis Efthymiou, Kostas Stefanidis, Evaggelia Pitoura, Vassilis Christophides. FairER: Entity Resolution With Fairness Constraints. 30th ACM International Conference on Information & Knowledge Management (CIKM ’21), Nov 2021, New York, France. 10.1145/3459637.3482105 . hal-03608623

HAL Id: hal-03608623
https://hal.archives-ouvertes.fr/hal-03608623
Submitted on 15 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
ABSTRACT
There is an urgent call to detect and prevent “biased data” at the earliest possible stage of the data pipelines used to build automated decision-making systems. In this paper, we are focusing on controlling the data bias in entity resolution (ER) tasks aiming to discover and unify records/descriptions from different data sources that refer to the same real-world entity. We formally define the ER problem with fairness constraints ensuring that all groups of entities have similar chances to be resolved. Then, we introduce FairER, a greedy algorithm for solving this problem for fairness criteria based on equal matching decisions. Our experiments show that FairER achieves similar or higher accuracy against two baseline methods over 7 datasets, while guaranteeing minimal bias.

CCS CONCEPTS
• Information systems → Entity resolution.

KEYWORDS
entity resolution, algorithmic fairness

ACM Reference Format:

1 INTRODUCTION
Given the widespread adoption of data-driven systems for making decisions in real life, there is an urgent call for responsible data management [28]. As a matter of fact, downstream harms to particular individuals or groups, often blamed to “biased data”, should be detected and prevented at the earliest possible stage of data pipelines used to build automated decision-making systems [13].

In this paper, we focus on controlling the data bias in entity resolution (ER) tasks aiming to discover and unify records/descriptions from different data sources that refer to the same real-world entity. ER tasks are typically used to improve data quality by reducing data incompleteness (i.e., missing values), redundancy (i.e., duplicate values) or inconsistency (i.e., conflicting values) [7]. Failing to address bias in ER tasks may lead to systematic bias that jeopardize both accuracy and fairness of downstream data analysis [18, 27, 30].

We argue that, in order to mitigate data bias implications, ER systems should not only maximize accuracy, as in traditional ER, but also satisfy fairness constraints among the so-called protected and non-protected groups of entities in their output. Since several criteria of determining protected groups (e.g., race, gender) and fairness measures (e.g., equal representation of groups, similar error rates among groups) have been proposed in the literature [19], the problem of fairness-aware ER is a broad one. We introduce a general constraint-based formulation of the problem and then investigate in depth a specific instance of this formulation.

Recent research has investigated the explainability of entity matching decisions [2, 8, 26], but controlling and mitigating bias in ER tasks has not been studied so far. Previous work has focused exclusively on the assessment of demographic bias in word embeddings exploited in named entity recognition [21] and on string similarity functions in name matching tasks [14]. Fairness-aware ER addresses discrepancies in the size of the groups in input data (i.e., popularity bias), where the majority of resolved entities belongs to a specific group [20]. To the best of our knowledge, FairER is the first ER algorithm that satisfies fairness constraints. In summary, the contributions of this work are:

• We formalize the problem of fairness-aware ER as a submodular optimization problem with cardinality constraints, which is known to be NP-hard. This definition requires that the retrieved pairs of entities should maximize a cumulative similarity score, while ensuring fair matching decisions across different groups of interest.
• We introduce FairER, a greedy algorithm to solve the problem of fairness-aware ER for fairness criteria based on equal matching decisions. We prove that FairER provides a $1 - 1/e$ approximation of the optimal solution.
• We experimentally evaluate FairER against fairness-agnostic ER and fair ranking methods over 7 datasets, and show that it consistently achieves similar (or even higher) accuracy, while guaranteeing minimal bias, with a negligible time overhead.

2 PROBLEM STATEMENT
Next, we define the problem of ER subject to fairness constraints.

Traditional Entity Resolution. Real-world entities are usually described in one or more sources by sets of attribute-value pairs. Entity descriptions may be structured in a tabular form (i.e., records) when stored in relational databases or semi-structured in a graph format (e.g., subject-predicate-object triples) when stored in knowledge graphs [7]. The objective of an ER algorithm is to discover
Thus, in a fair ER setting, the results that are retrieved first should (e.g., DeepMatcher [23], DeepER [9], and [12]). In the remaining \( R \) joint groups based on the values of their protected attributes and fits and harms across groups by looking at the decision alone [22].

decision measures that allow us to examine the allocation of bene-

they should also satisfy a given fairness constraint.

By giving high or low importance to fairness or diversity.

glomerative clustering (e.g., SiGMa [17]), or binary classification

e and \( e' \) belong to the protected group according to the value of one, or more protected attributes. A less error-tolerant protected group membership function on pairs would consider conjunction instead of disjunction. More challenges regarding this function arise, including handling missing or conflicting values, as well as learning it on the fly from the data. We leave the exploration of the impact of this decision as future work.

**Ranked group fairness** [5, 16, 32], employed in settings where the ordering of results is important, requires that a fairness constraint should be satisfied when considering the results within a given rank position. In our example, if we want to hire 4 experts, we may not only want the first 4 results to be matches, but also require that they equally represent males and females. To introduce a fairness-aware adaptation of Definition 2.1, we use an abstract fairness criterion \( F \), and require that the provided solution \( R \) should satisfy \( F \).

**Definition 2.2 (Fairness-aware ER).** Given a set of candidate matches \( C \subseteq E \times E' \), a scoring function \( s : E \times E' \rightarrow \mathbb{R} \), and a fairness criterion \( F \), produce a ranking of matches \( R \subseteq C \) that for any given rank position \( k \), maximizes the cumulative scores:

\[
R = \arg \max_{R \subseteq C} \sum_{(e_i, e'_i) \in R} s(e_i, e'_i)
\]

\[
\text{s.t. } R[k] \text{ satisfies } F,
\]

where \( R[k] \) are the \( k \) first results of \( R \).

In the previous example, 3 out of 4 suggested pairs by traditional ER are male candidates (Figure 1). A fairness-aware ER algorithm targeting an equal representation of males and females would return a balanced ranking like the last ranking shown in the figure.

Note that the fairness-aware ER problem of Definition 2.2 is not the same as the fair top-k ranking problem [32], since in fairness-aware ER, the items for which a positive matching decision will be made, i.e., the subset of candidate matches that are returned as matches, are not known in advance, but may be dynamically updated based on previous decisions. For example, a fair ranking \( R[4] = \{(e_5, e'_5), (e_3, e'_3), (e_4, e'_4)\} \) (with two male and two female candidates) may not be a valid ER result, since returning both \( (e_5, e'_5) \) and \( (e_3, e'_3) \) violates the redundancy-free assumption.

**3 FAIRER ALGORITHM**

The fairness-aware ER problem introduced in Definition 2.2 is general enough to encompass different settings with respect to fairness criteria and matching constraints. In this section, we present a simple, yet highly efficient method, FairER, to solve an instance of this problem for a specific setting, as described next.

Inspired by the ranked group fairness constraint [32], we denote with \( |R_p|/k \) (resp. \( |R_n|/k \) the ratio of protected (resp. non-protected) group members in the first \( k \) results \( R[k] \), and require that the protected and non-protected groups are equally represented in \( R[k] \), i.e., \( |R_p|/k = |R_n|/k \). In other words, our ranked group fairness criterion \( F \) is defined as \( |(|R_p|/k) - (|R_n|/k)| = c \ast \), where
\( \varepsilon^* \) is the smallest possible ratio difference for a given \( k \). As shown later, \( \varepsilon^* = 0 \) when \( k \) is even, and \( \varepsilon^* = \frac{1}{k} \), when \( k \) is odd.

As the base matching decision, we build on top of a simple algorithm (similar to existing approaches \([4, 17]\)) that places candidate pairs in a priority queue (PQ) in descending scores and at each iteration, returns the top pair as a match. Then, it updates the queue by removing pairs containing descriptions that have been already matched, to respect the one-to-one matching constraint, and proceeds until the queue becomes empty. In FairER (Algorithm 1), we extend this method to use not only one, but two priority queues, one \( (Q_p) \) for the protected group and one \( (Q_n) \) for the non-protected group (Lines 6-10). The matching algorithm proceeds as usual, in each iteration for one of the priority queues only (Lines 13-16), matched, to respect the one-to-one matching constraint, and proceeds until the queue becomes empty. In FairER (Algorithm 1), we extend this method to use not only one, but two priority queues, one \( (Q_p) \) for the protected group and one \( (Q_n) \) for the non-protected group (Lines 6-10). The matching algorithm proceeds as usual, in each iteration for one of the priority queues only (Lines 13-16), until a match is found (Lines 17-21). Then, it proceeds to the next non-empty priority queue (Lines 22-23), until \( k \) matches have been found (early termination - optional), or all queues are empty. The time complexity of Algorithm 1 is \( O(|C| \log |C|) \).

**Algorithm 1:** FairER

**Input:** Candidate matches \( C = (E, E', s) \), \( k \) (optional)

**Output:** Ranked matches \( R[k] = \langle (e_1, e_1'), \ldots, (e_k, e_k') \rangle \)

1. \( R \leftarrow \emptyset \)
2. \( M \leftarrow \emptyset \) // matched nodes from \( E \)
3. \( M' \leftarrow \emptyset \) // matched nodes from \( E' \)
4. \( Q_p \leftarrow \emptyset \) // PQ for protected group in desc. score
5. \( Q_n \leftarrow \emptyset \) // PQ for non-protected group in desc. score
6. foreach \( e = (e, e', s(e, e')) \) \( \in C \)
7. if \( p(e) \) then // returns True if \( c \) is in protected group
8. \( Q_p, \text{put}(c) \)
9. else
10. \( Q_n, \text{put}(c) \)
11. \( \text{nextPr} \leftarrow \text{True} \) // should the next pair be protected?
12. while \( (Q_p \neq \emptyset \text{ or } Q_n \neq \emptyset) \text{ and } |R| < k \) do
13. if \( \text{nextPr} \)
14. \( c \leftarrow Q_p, \text{pop}() \) // the top protected pair
15. else
16. \( c \leftarrow Q_n, \text{pop}() \) // the top non-protected pair
17. if \( c, e \in M \text{ or } c, e' \in M' \) then // \( e \) and \( e' \) are matched
18. \( \text{continue} \)
19. \( R, \text{append}((c, c, e, e')) \)
20. \( M \leftarrow M \cup \{c, e\} \)
21. \( M' \leftarrow M' \cup \{c, e'\} \)
22. if \( \text{nextPr and } Q_n \neq \emptyset \) or \( \neg \text{nextPr and } Q_p \neq \emptyset \) then
23. \( \text{nextPr} \leftarrow \neg \text{nextPr} \) // swap queues
24. return \( R \)

**Proposition 3.1.** Algorithm 1 is a \( 1 - 1/e \) approximation to the problem of Definition 2.2, for \( F \) defined as \( |(|R_p|/|k|) - (|R_n|/|k|)| = \varepsilon^* \).

**Proof (Sketch).** Solving the problem of Definition 2.2 reduces to maximizing a monotone submodular function with cardinality constraints \( (k) \), which is an NP-hard problem \([24]\). The greedy approach of Algorithm 1 is a known \( 1 - 1/e \) approximation to this problem, noting that the next pair to be added on each iteration, is guaranteed to satisfy \( F \). We omit the detailed proof due to limited space. We focus on the satisfaction of the fairness criterion \( F \).

| Dataset          | Pr. group criterion | \( |R_p| \), \( |R_n| \) |
|------------------|---------------------|-----------------|
| BeerAdv-RateBeer (D1) | “Red” in beer name | 5, 9            |
| iTunes-Amazon (D2)   | “Dance” in genre  | 11, 16          |
| Fodors-Zagats (D3)    | type = “asian”    | 3, 19           |
| DBLP-ACM (D4)        | female last author | 39, 405         |
| DBLP-Scholar (D5)     | “vldb” in venue   | 80, 990         |
| Amazon-Google (D6)   | “Microsoft” in manufacturer | 12, 222 |
| Walmart-Amazon (D7)  | category = “printers” | 11, 182 |

Let \( R_p \subseteq R[k] \) be the protected group matches and \( R_n \subseteq R[k] \) be the non-protected group matches, where \( |R_p| + |R_n| = k \), and let \( |(|R_p|/|k|) - (|R_n|/|k|)| \geq \varepsilon^* \) for all possible solutions \( R' \). For a constant \( k \), this absolute difference is minimized when \( |(|R_p| - |R_n|)| \) is minimized. It is straightforward to show that when \( k \) is even, this equation is minimized for \( |R_p| = |R_n| = k/2 \), so \( \varepsilon^* = 0 \), while, when \( k \) is odd, it is minimized for \( |(R_p| - |(R_n)|)| = 1 \), so \( \varepsilon^* = 1/k \). This holds in the general case when \( k \leq \min(|Q_p|, |Q_n|) \). Otherwise, \( \varepsilon^* \) is bigger. By construction, Algorithm 1 produces \( R_p \) and \( R_n \), such that \( |R_p| = |R_n| = k/2 \), when \( k \) is even, and \( |R_p| = (k + 1)/2 \) = \( |R_n| + 1 \), when \( k \) is odd. In both cases, \( |(|R_p|/|k|) - (|R_n|/|k|)| = \varepsilon^* \). \( \square \)

4 EXPERIMENTS

In this section, we present results comparing FairER to two baseline methods over 7 publicly available datasets.

**Color-blind baseline.** This baseline is the classic paradigm of fairness-agnostic ER, solving Problem 2.1, in which the candidate matches are forwarded to matching and matches are returned ignoring fairness. We use it as a baseline for comparing fairness-aware to fairness-agnostic ER methods.

**Fais baseline.** In this baseline, the candidate matches are first ranked in descending score order and then re-ranked (using FAIR [32]) to respect the fairness constraints. The re-ranked list is then forwarded to matching to produce the final output of ER.

The problem with Fa’ir baseline is that it guarantees that the input to matching is fair, but the output may not be. On the other hand, it is straightforward to prove that FairER guarantees maximal fairness until no more protected group matches exist to be returned, i.e., until the priority queue for the protected group becomes empty.

**Setup.** Our open-source framework uses DeepMatcher (10 epochs and defaults for the other parameters, yielding similar scores to [23]) as the scoring function \( s \), and FAIR\(^2\) (for FA’ir baseline).

**Datasets.** All datasets are publicly available from DeepMatcher. We list them in Table 1, along with the criteria we employed for defining the protected group in each dataset, and the number of ground truth matches in the protected \( R_p \) and non-protected \( R_n \) groups. For detecting gender from authors’ first names in DBLP-ACM dataset, we used the gender-guesser library\(^3\). This was not possible in DBLP-Scholar dataset, where first names are abbreviated. We note that those criteria are orthogonal to our approach and they need not be personal data, as we show in our examples.

\(^2\)https://github.com/vefthym/fairER

\(^3\)https://github.com/anhaidgroup/deepmatcher

\(^4\)https://github.com/MikaLichtblau/FA-IR_Ranking

\(^5\)https://github.com/lead-ratings/gender-guesser
Measures. We evaluate the suggested method and baselines with respect to accuracy and fairness at the top-k positions of the returned results, for \( k \in \{5, 10, 15, 20\} \). The employed measures are the following. Accuracy@\( k \): out of \( k \) returned matches, how many are correct (i.e., in the ground truth of known matches)? Bias@\( k \): following the fairness constraint \( F \) defined earlier as \( |(|R_p|/k) - (|R_n|/k)| = \epsilon \), we report the values of \( (|R_p|/k) - (|R_n|/k) \), with negative values denoting favoring the non-protected group, zero implying no bias, and positive values favoring the protected group.

Scores of protected vs non-protected. Figure 2 shows the average scores for protected and non-protected groups, also presenting the scores only for known (from the ground truth) matches. As expected, matches have much higher average scores than non-matches. We observe that in all cases, matches belonging to the protected group (aka protected matches), shown in grey color, have lower average match scores than matches belonging to the non-protected group (aka non-protected matches), shown in blue color. The same observation holds when comparing protected (black) vs non-protected (orange) pairs (including matches and non-matches).

Accuracy. Table 2 shows the accuracy@\( k \) results for all three methods. We observe that FairER yields lower accuracy than the other methods in very few occasions, while, surprisingly, it yields better accuracy than even the color-blind method in a few cases (Walmart-Amazon \( k = 5 \), Amazon-Google \( k = 10 \), DBLP-Scholar \( k = 15 \)). Note that accuracy may change if a different matching algorithm than the one described in Section 3 is employed.

Table 3: Fairness (bias@\( k \)) results. [-1,0): favoring non-protected, 0: no bias, (0,1]: favoring protected. The minimum possible |bias@\( k \)| values \( \epsilon \) are also presented.

<table>
<thead>
<tr>
<th>Method</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>D6</th>
<th>D7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color-blind</td>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.1</td>
<td>-0.1</td>
<td>-0.1</td>
<td>-0.1</td>
<td>-0.6</td>
</tr>
<tr>
<td>FairER</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
</tr>
</tbody>
</table>

5 CONCLUSION

In this paper, we have introduced the problem of fairness-aware ER and proposed a general constraint-based formulation. We have presented FairER, an algorithm that solves an instance of this problem for the case of fairness constraints expressed in terms of the cardinalities of the protected and non-protected groups in the output. We are currently working on extending our Algorithm 1 to more complex protected group criteria, such as handling missing and conflicting values for the protected attributes. As a future work, we plan to study mitigation of bias in other ER tasks, such as clustering and fusion. We additionally plan to study the impact of alternative fairness measures on ER, given that several competing definitions have been proposed in the literature that imply different, possibly mutually exclusive, understandings of fairness [6, 15, 22, 25].

ACKNOWLEDGMENTS

This project has received funding from the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), under grant agreement No 969.
REFERENCES

record matching packages. In SIGMOD. 783–794.

456.


[7] Vassilis Christophides, Vassilis Efthymiou, Themis Palpanas, George Papadakis,

and Mourad Ouzzani. 2019. EXPLAINER: Entity Resolution Explanations. In

Ouzzani, and Nan Tang. 2018. Distributed Representations of Tuples for Entity

[10] Vassilis Efthymiou, George Papadakis, Kostas Stefanidis, and Vassilis
Christophides. 2019. MinoanER: Schema-Agnostic, Non-Iterative, Mas-

382–395.

[12] Cheng Fu, Xiampi Han, Le Sun, Bo Chen, Wei Zhang, Shuhui Wu, and Hao Kong.
2019. End-to-End Multi-Perspective Matching for Entity Resolution. In JICAF.
4961–4967.

[13] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miroslav Dudík,
and Hanna M. Wallach. 2019. Improving Fairness in Machine Learning Systems:
What Do Industry Practitioners Need?. In CHI.

Matching Tasks. In EDBT. 626–629.


[17] Simon Lacoste-Julien, Konstantina Palla, Alex Davies, Gjergji Kasneci, Thore
Graepel, and Zoubin Ghahramani. 2013. SiGMA: simple greedy matching for
aligning large knowledge bases. In KDD. 572–580.

[18] Peng Li, Xi Ran, Jennifer Blase, Yue Zhang, Xu Chu, and Ce Zhang. 2021. CleanML:
A Study for Evaluating the Impact of Data Cleaning on ML Classification Tasks.
In ICDE.


[20] Edgar Meij, Tara Safavi, Chenyan Xiong, Gianluca Demartini, Miriam Redi,
2020.


[23] Sidharth MUDGAL, Han Li, Theodors Rekatsinas, Andi Dao, Youngchoon Park,
Deep Learning for Entity Matching: A Design Space Exploration. In SIGMOD.
19–34.

analysis of approximations for maximizing submodular set functions - I. Math.


loop System for Explainable Entity Resolution. Proc. VLDB Endow. 12, 12 (2019),
1794–1797.

FairPrep: Promoting Data to a First-Class Citizen in Studies on Fairness-Enhancing

[28] Julia Stoyanovich, Bill Howe, and H. V. Jagadish. 2020. Responsible Data Man-

[29] Harini Suresh and John V. Guttag. 2019. A Framework for Understanding Uninten-

2019. Data Cleaning for Accurate, Fair, and Robust Models: A Big Data - AI

in the Wild: A Consistent and Versatile Framework to Unify Data in Industrial
Applications. In SIGMOD. 2287–2301.

[32] Meike Zehlike, Francesco Bonchi, Carlo Castillo, Sara Hajan, Mohamed Megha-
CIKM. 1569–1578.