Entity Matching with AUC-Based Fairness

Soudeh Nilforoushan
The University of Western Ontario
London, Ontario, Canada
snilforo@uwo.ca

Qianfan Wu
The University of Western Ontario
London, Ontario, Canada
qwu246@uwo.ca

Mostafa Milani
The University of Western Ontario
London, Ontario, Canada
mostafa.milani@uwo.ca

***Abstract*—The research on fair *machine learning (ML)* has been growing due to the high demand for unbiased and fair ML models for objective decision-making. Most of this research has been focused on training and tuning the ML model, and less effort has been made to study biases in the processes that clean and prepare data for these models. This paper studies fairness in *entity matching (EM)*, a.k.a. *record matching* and *entity resolution*, a primary task in a data cleaning pipeline that can significantly impact ML models' performance. We introduce a new metric for measuring bias in EM based on *Area Under the Curve* (AUC) and the risk of record matching between and within subpopulations. We use this metric and real-world data to show biases in a state-of-the-art EM technique. We introduce a debiasing algorithm based on *data augmentation (DA)* to mitigate bias and conduct experiments to show the algorithm's effectiveness.**

I. Introduction

Fairness is essential in applications that rely on ML models for decision-making. Possible biases in these models, or the data they consume, can cause discrimination against specific individuals or subpopulations, particularly minorities. The existing research in fairness focuses on the last stages of developing ML models, including model training and tuning, and misses the opportunity for mitigating or reducing biases in the early stages, such as data collection, preparation, and cleaning. While several recent research studies (e.g., [1], [2], [3], [4], [5], [6], [7], [8]) have taken on fairness in these early stages, the subject has mainly remained unexplored. In this paper, we study fairness in *entity matching (EM)*, the problem of finding duplicate records in data, which is a primary task in data preparation and cleaning and is integral to downstream ML applications that rely on clean data to function.

EM is a long-standing problem in data management and has been widely studied in other areas, such as information retrieval, artificial intelligence (AI), and ML. As such, there are diverse solutions for EM that apply techniques, including textual similarity, rule-based methods, pair-wise classification, clustering approaches, and probabilistic inference (see [9] for a survey). The state-of-the-art EM solutions (e.g., [10], [11], [12], [13]) address EM as a binary classification problem, where the goal is to classify pairs of records into *matched* or *unmatched*. A classifier in these solutions, usually called a *matcher*, is accurate if it matches equivalent record pairs (i.e., record pairs that refer to the same real-world entity) and labels nonequivalent record pairs as unmatched.

A matcher often consists of a score function and a matching threshold. It works by ranking the input record pairs using the score function and matching the record pairs if their scores are higher than the threshold. The score of a pair reflects the probability that a pair matches, and the threshold controls the false positive rate vs. the false negative rate. The threshold is particularly important in applications where miss-matching records are costly, e.g., miss-matching medical records of patients that can be life-threatening [14] or wrongly pairing profiles of businesses in online review systems that can damage a business reputation [15].

Example 1. Table I shows a few patient records from a relation that is collected from multiple health institutions and has duplicate records, e.g., $\{r_2, r_3, r_7\}$ and $\{r_1, r_6\}$ are sets of equivalent records that are highlighted with the same color in the table, and r_4 and r_5, which are not highlighted, refer to different patients. Tables IIa and IIb demonstrate two lists of pairs of records from Table I. The lists consist of the same set of pairs ordered by their scores returned from the score functions s_1 and s_2. "Equiv" in the lists is the true label, i.e., whether the two records in a pair are equivalent.

A matcher consisting of s_1 and a matching threshold 0.5 (r_3, r_4) and (r_2, r_6), and labels the other pairs as unmatched. The score functions s_1 and s_2 rank one equivalent record pair lower than a nonequivalent pair, e.g., s_1 ranks the equivalent pair (r_2, r_3) lower than the nonequivalent pair (r_4, r_5). This means the scores will make matchers with error, independent of the threshold. Nonetheless, the threshold controls the risk of false positives (matching nonequivalent records) and false negatives (not matching equivalent records); e.g., a matcher with s_1 and a threshold 0.87 has no false positive but has false negative as it does not match (r_2, r_3). ■

To study fairness in EM, we adopt an AUC-based measure of bias introduced in [16].[1] *The area under the curve (AUC)* is a widely used performance measure for classifiers [22] and refers to the area under a two-dimensional curve that shows the true positive rate vs. the false positive rate for varying thresholds in $[0, 1]$. The AUC of a binary classifier has a probabilistic interpretation and refers to the probability that a positive example is ranked higher than a negative example. Since it is independent of any classification threshold, it gives a more meaningful measure of how well a classifier separates positive and negative examples compared with the

[1] Similar measures of bias using the AUC are presented in [17], [18], [19], [18], [20], [21]. See Section II for a brief review.

other performance measures, such as precision, recall, and F1 measures, that use the confusion matrix for a discrete classifier. The AUC is also proved more effective for evaluating a classifier with imbalanced data [23].

Our measure of bias is based on the new notion of subAUC that evaluates the AUC per subpopulation. To define subAUC, we use *the cross AUC (xAUC)* between two subpopulations introduced in [16]. For two subpopulations a and b, the xAUC is the probability that a positive example in the subpopulation a is ranked higher than a negative example from the subpopulation b. A substantial difference between the xAUC of a, b and the xAUC of b, a shows that one subpopulation is advantaged over the other subpopulation. We adopt the xAUC to define *subAUC* and bias for EM. In a nutshell, the subAUC of a score function for a subpopulation measures how well the function ranks record pairs for EM when at least one record from the subpopulation is in the record pairs. Example 2 explains the AUC, subAUC, and our new measure of bias in EM.

	NAME	**ETHNICITY**	**ADDRESS**
r_1	Goldie R. Chisolm	Caucasian	1195 Holly St. Athens, GA
r_2	T. S. Fatimata	Afr. Am.	109 Ralph St
r_3	Thokozani Fatimata	African American	109 Ralph St Belleville, NJ
r_4	Naomi Rodrigez	Asian	3672 Glory Road, FL
r_5	Xavier Rodrigez	Latino	3672 Glory Road, FL
r_6	G. R. Chisolm	Caucasian	1195 Holly St., GA
r_7	T Fatimata	African American	109 Ralph St Belleville

TABLE I: Patient records

Pair	Equiv	Score
(r_1, r_6)	1	0.98
(r_4, r_5)	0	0.80
(r_2, r_3)	1	0.77
(r_3, r_4)	0	0.04
(r_2, r_6)	0	0.01

(a) s_1

Pair	Equiv	Score
(r_2, r_3)	1	0.91
(r_3, r_4)	0	0.86
(r_1, r_6)	1	0.79
(r_4, r_5)	0	0.12
(r_2, r_6)	0	0.07

(b) s_2

TABLE II: Two score functions s_1 and s_2 for EM.

Example 2. For an EM score function, such as s_1 and s_2, the AUC is the probability that an equivalent record pair is correctly ranked higher than a nonequivalent record pair. This measures how well the score function ranks record pairs based on their matching probability and independent of any particular threshold. It is 1 when the function ranks all equivalent pairs higher than nonequivalent pairs and is 0.5 when the ranking is random.

Assuming that the record pairs in Table IIa are random samples from the set of possible record pairs, we can estimate the AUC of s_1. This can be done comparing the scores of the equivalent pairs (e.g., (r_1, r_6) and (r_2, r_3)) with the score of the nonequivalent pairs (e.g., (r_4, r_5), (r_3, r_4), and (r_2, r_6)), and counting the number of correct rankings. In all these 6 comparisons, the record pairs are correctly ranked by s_1, except in one where the equivalent pair (r_2, r_3) with score 0.77 is ranked lower than the nonequivalent pair (r_4, r_5) with score 0.80. Therefore, the estimation of s_1's AUC is $5/6$. Following similar steps, one can obtain the same estimate for s_2 using the record pairs in Table IIb showing that both s_1 and s_2 have the same classification quality based on their AUCs.

The ethnicity attribute in Table I defines four subpopulations of Caucasians, African Americans, Asians, and Latinos. *We define the subAUC for a subpopulation as the probability that a pair of equivalent records is ranked higher than a pair of nonequivalent records when at least one record in the two pairs is from the subpopulation.* To compute the subAUC of s_1 for Asians using Table IIa, we consider (r_4, r_5) and (r_3, r_4) that include r_4, the only record from the Asian subpopulation. Since the pairs are nonequivalent, they must be compared with the equivalent pairs (r_1, r_6) and (r_2, r_3), making 4 possible comparisons out of which s_1 only miss-ranks one as it ranks (r_4, r_5) higher than (r_2, r_3). Therefore, the subAUC for Asians is $\frac{3}{4}$. Similarly, the subAUCs for African Americans, Latinos, and Caucasians are $\frac{4}{5}$, $\frac{1}{2}$, and 1. This means the quality of s_1 differs between the subpopulations implicating biases in EM. We measure this bias by the difference between the minimum and the maximum subAUCs: $1 - \frac{1}{2} = \frac{1}{2}$. For s_2, the estimated subAUCs for Asians, African American, Caucasians and Latinos w.r.t. Table IIb are $\frac{3}{4}$, $\frac{4}{5}$, $\frac{4}{5}$, and 1, and the bias is $1 - \frac{3}{4} = \frac{1}{4}$. This shows although s_1 and s_2 have the same AUC (i.e. $\frac{5}{6}$), s_2 incurs less bias compared to s_1. ■

We conduct experiments to measure bias in DITTO, a state-of-the-art EM technique [10]. To reduce bias, we present an algorithm that augments training data with two categories of new record pairs. The first category includes record pairs that can be added by applying transitivity, reflexivity, and symmetry as the properties of the matching relationship as an equivalent relation between records. The second category consists of records from modifying the existing record pairs in the input training data. We use ROTOM [24], a general DA framework for data management tasks to generate new record pairs. ROTOM is used for data augmentation in data cleaning and entity resolution. It leverages Seq2Seq-based NLP models, and policies for selecting and weighting augmented data examples and balances diversity and quality in DA.

This work is aligned with the recent research on fairness in data preparation and cleaning, e.g., [1], [2], [3], [4], [5], [6], [7], which we review in Section II. The paper is structured as follows. Section II is a brief review of the related work. We define the fair EM problem by formulating subAUC and bias in Section III, present our debiasing algorithm in Section IV, and explain our experimental results in Section V. We discuss future work and conclude the paper in Section VI.

II. RELATED WORK

In this section, we briefly review the related work.

A. AUC-Based Fairness

The conventional bias measures, such as *demographic parity*, *equalized odds*, and *equal opportunity*, rely on the confusion matrix of a discrete classifier. The AUC-based bias measures (e.g., [17], [18], [19], [16]) are more recent and are not as popular. However, they can measure bias for classifiers that not only predict a class for an input example but also give a probability of belonging to the class.

An early work about AUC-based bias is [17], [18], where the authors introduce metrics for unintended bias in text classification. *Subgroup AUC* is the AUC for the examples from a subgroup and represents a model's separability within the subgroup. *Background Positive Subgroup Negative (BPSN)* AUC is the AUC on the positive examples from the background (all subgroups) and the negative examples from a particular subgroup. *Background Negative Subgroup Positive (BNSP)* AUC is similarly defined and specifies the AUC on the negative examples from the background and the positive examples from the subgroup. The authors use the metrics to evaluate fairness in detecting toxic comments in Wikipedia Talk. Certain identity terms, such as *Muslim* or *gay*, were often given unfairly high toxicity scores.

Beutel et al. present intra-group and inter-group pairwise AUC fairness metrics for evaluating algorithmic fairness in recommender systems [19]. The intra-group fairness expects an equal likelihood of a clicked item being ranked above another relevant unclicked item from all groups. The inter-group fairness concerns similar likelihoods, but when the clicked and unclicked items are from different groups. They compute the metrics in randomized experiments that provide an efficient way of reasoning about fairness in rankings from recommender systems. They also introduce a new regularizer to integrate these metrics with model training and improve fairness in the resulting rankings of the recommender models.

The *xAUC (cross-AUC) disparity* in [16] is a metric that assesses the disparate impact of risk scores and is defined using the xAUC measure. The xAUC is the probability of correctly ranking a positive instance of a group above a negative instance of another group. For two groups a and b, the xAUC disparity is the difference between the xAUC of a and b vs. the xAUC of b and a, where a substantial difference means one group is systematically disadvantaged compared to the other group. The authors use the xAUC disparity to evaluate recidivism prediction risk scores. We adopt the xAUC to define subAUC for the EM task.

Fong et al. define bias as a normalized distance between the min AUC and the max AUC within groups, where the AUC for a group is similar to subgroup AUC in [18]. They use feature augmentation to improve the min AUC for a disadvantaged group. They prove the improvement through feature augmentation using theoretical analysis and also empirically in real-world contexts. Vogel et al. [20] introduce general families of fairness definitions based on the AUC and show that their AUC-based constraints can be instantiated such that classifiers obtained by thresholding the scoring function satisfy classification fairness for a desired range of thresholds. They establish generalization bounds for scoring functions learned under such constraints, design practical learning algorithms, and show the relevance of our approach with numerical experiments on real and synthetic data. The work in [21] defines an AUC-based group fairness metric by combining intra-group and inter-group AUCs. The authors propose a minimax learning and bias mitigation framework that incorporates intra-group and inter-group AUCs while maintaining utility. The authors use a stochastic optimization algorithm with convergence proof to minimize group-level AUC.

B. Fairness in Data Preparation and Cleaning

Data quality has a significant impact on the performance of any downstream ML models. The existing data preparation and cleaning techniques improve data quality but ignore possible biases [25], [26], [27], [28]. The fairness impact of data preprocessing stages in an ML pipeline has been studied in [4]. The authors use a causal model to consider bias with and without each stage in a preprocessing pipeline and use the existing metrics such as statistical parity difference, equal opportunity difference, average odds difference, and error rate difference for measuring bias. They conduct a fairness evaluation of the preprocessing stages in different pipelines that demonstrates certain data transformers are causing the models to exhibit unfairness.

Fairness in data wrangling is the focus of the work in [5]. Data wrangling is identifying, extracting, cleaning, and integrating data for an exploration and analysis application [29]. Schema matching for collecting data from sources and data profiling for extracting data dependencies are two main tasks in a data wrangling pipeline. The authors in [5] use interventions to evaluate the impact of the decisions made in schema matching and data profiling on the fairness in the resulting data from a wrangling pipeline.

FairPrep is a framework based on AIF360 for evaluating ML model fairness and scikit-learn for applying data transformations. It integrates fairness with the data life cycle, including value imputation, preparation, feature selection, model selection, and hyper-parameter tunning [6], [7]. This general framework does not particularly focus on EM in the data preparation pipeline.

Two recent works study fairness in entity resolution. Karakasidis et al. study fairness in name matching and define bias by comparing the number of mismatches of individuals in groups. They use real data to showcase biases in the existing methods of name matching [2]. Their work is different than ours as they do not consider structured data. The most relevant work to ours is FairEM [1], which studies fairness in entity resolution for relational data. The main difference between their work and ours is that their entity resolution problem is limited to finding one-to-one matching between records integrated from two sources. Fairness is translated to having an equivalent number of records in top-k ranked record pairs. Our work is different as we allow records from arbitrary sources and minimize AUC-based for fair EM.

III. Preliminaries and Problem Definition

Assume a set of records R with schema $\mathcal{R}$ that consists of attributes $A_1, ..., A_m$. For a record $r \in R$, $r[A_j] \in dom(A_j)$ is A_j's value in r where $dom(A_j)$ is the domain of $A_j \in \mathcal{R}$. We use $R[A_j] \subseteq dom(A_j)$ to refer to the values of A_j in R.

A dataset D is a subset of $R \times R \times \{0,1\}$ and contains labeled record pairs from R. A pair (r, r') is labeled $y = 1$ if r and r' are *equivalent*, i.e., they refer to the same real-world entity, and is labeled $y = 0$ otherwise. We use I to refer to the set of all possible record pairs, i.e., $I = R \times R$, and denote the subsets of equivalent and nonequivalent record pairs in I with I^+ and I^- ($I = I^+ \cup I^-$).

We assume one sensitive attribute $A \in \mathcal{R}$. For each sensitive value $a \in \mathit{dom}(A)$, we define a *subpopulation* $R_{A[a]} \subseteq R$ as $\{r \in R \mid r[A] = a\}$ (e.g., $R_{\mathit{Gender[Male]}}$) refers to the male individuals. We use R_a (e.g., R_{Male}) when the sensitive attribute (e.g., *Gender*) is clear from the context. For a subpopulation R_a, we define $I_a = (R_a \times R) \cup (R \times R_a) \subseteq I$ as the set of record pairs with at least one record in R_a. Using I_a, we define $I_a^+ = I^+ \cap I_a$ and $I_a^- = I^- \cap I_a$ that contain equivalent and nonequivalent pairs with the records in R_a.

A *matcher* is a function $f : R \times R \mapsto \{0, 1\}$ that receives a record pair, e.g., (r, r'), and returns 0 or 1. If $f(r, r') = 1$, we say f matches r and r', or r and r' match when f is clear from the context. We call y the true label of (r, r'), and $f(r, r')$ the predicted label of the pair. *The problem of entity matching (EM)* is to find the best matcher w.r.t. a training dataset D where different notions of loss for binary classification can be used to define the best matcher while comparing the record pairs matched by the matcher with the equivalent record pairs.

The existing EM solutions often provide a matcher f that consists of a *score function* $s : R \times R \mapsto [0, 1]$ and a *matching threshold* $\theta \in (0, 1)$, where s receives a record pair and returns a real number that reflects the probability of matching the records. The matcher f matches record pairs r and r', i.e., $f(r, r') = 1$, if $\theta \leq s(r, r')$, and it labels them as unmatched, i.e., $f(r, r') = 0$, otherwise.

The AUC of a binary classifier measures how well the classifier separates positive and negative examples [22]. For EM, the AUC measures how well s in a matcher f separates equivalent pairs from non-equivalent pairs independent of any threshold θ. We assume a probability distribution *Pr* with the following random variables to formally define the AUC in EM. X and X' are random variables referring to random records in R, Y is a binary random variable that is 1 if X and X' are equivalent and 0 otherwise, and S is a random variable representing the score of X and X' returned from s. TPR and FPR for a classifier f with s and θ are $\mathit{TPR}_s(\theta) = \mathit{Pr}(S \geq \theta | Y = 1)$ and $\mathit{FPR}_s(\theta) = \mathit{Pr}(S \geq \theta | Y = 0)$. The AUC of s is defined as follows:

$$\mathit{AUC}_s = \int_0^1 \mathit{TPR}_s(\mathit{FPR}_s^{-1}(r))\, dr. \tag{1}$$

where FPR_s^{-1} is the inverse of FPR and gets a rate in $[0, 1]$ and returns the threshold θ for s that gives the rate as FPR.

To formalize bias in EM, we define subAUC, which measures the performance of a score function in subpopulations. To this end, we assume a sensitive attribute A, such as gender or race, with a discrete domain $\mathit{dom}(A)$. For a subpopulation $R_{A,a}$ with $a \in \mathit{dom}(A)$, TPR and FPR are defined as follows:

$$\mathit{TPR}^a(\theta) = \mathit{Pr}(S \geq \theta | Y = 1, (X, X') \in I_a), \tag{2}$$

$$\mathit{FPR}^a(\theta) = \mathit{Pr}(S \geq \theta | Y = 0, (X, X') \in I_a). \tag{3}$$

where $(X, X') \in I_a$ means either X or X' (or both) are from the subpopulation R_a. Using TPR and FPR per subpopulation, we define pairwise AUC that provides a basis for defining subAUC.

Definition 1 (Pairwuse and Background AUCs)**.** Consider a score function s for the pairs of records in R. The *pairwise AUC* of two subpopulations R_a and R_b in R is defines as

$$\mathit{AUC}_s^{a,b} = \int_0^1 \mathit{TPR}_s^b((\mathit{FPR}^a)^{-1}(x))\, dx \tag{4}$$

The *background AUC* of R_a replaces R_b with R:

$$\mathit{AUC}_s^{a,A} = \int_0^1 \mathit{TPR}_s((\mathit{FPR}^a)^{-1}(x))\, dx \tag{5}$$

We use A in the superscript instead of b to represent the entire population R rather than R_b. ■

A probabilistic interpretation of the pairwise AUC in Equation 4 is the probability that s ranks a random pair in I_a^+ higher than a random pair in I_b^-. The pairwise AUC is an extension of the xAUC in [16] to the EM task. The background AUC of R_a in Equation 5 also has a clear probabilistic meaning and represents the probability that s ranks a random pair in I_a^+ higher than a random pair in I^-. The pairwise and background AUCs measure the quality of record matching using s between subpopulations. To measure the quality of record matching within a subpopulation, we can use $\mathit{AUC}_s^{a,a}$ that is obtained from Equation 4 by replacing R_b with R_a. The three AUCs of $\mathit{AUC}_s^{a,A}$, $\mathit{AUC}_s^{A,a}$, and $\mathit{AUC}_s^{a,a}$ measure the quality of s for matching records related to the subpopulation R_a.

To define bias, we seek a single performance measure for each subpopulation that combines the three AUCs for a subpopulation R_a. Defining such a measure is not trivial because the two background AUCs ($\mathit{AUC}_s^{a,A}$ and $\mathit{AUC}_s^{A,a}$) may overlap, and both may consider the rankings of some shared record pairs. Another challenge in combining the three AUCs is having different weights because they can consider different numbers of comparisons between record pairs. We integrate them in Definition 2 to define *subAUC* as a single performance measure for s and the subpopulation R_a.

Definition 2 (subAUC)**.** Consider a score function s for the record pairs in I. The *subAUC* of a subpopulations R_a is defined as follows:

$$\begin{aligned}\mathit{subAUC}_s^a = w^{a,A} \times \mathit{AUC}_s^{a,A} + w^{A,a} \times \mathit{AUC}_s^{A,a} - \\ w^{a,a} \times \mathit{AUC}_s^{a,a},\end{aligned} \tag{6}$$

where the weights are $w^{a,A} = \frac{c^{a,A}(I)}{c^a(I)}$, $w^{A,a} = \frac{c^{A,a}(I)}{c^a(I)}$, and $w^{a,a} = \frac{c^{a,a}(I)}{c^a(I)}$ with $c^{a,b}(I) = |I_a^+| \times |I_b^-|$ and $c^a(I) = c^{a,A}(I) + c^{A,a}(I) - c^{a,a}(I)$. ■

The subAUC in Equation 6 is the probability that s ranks an equivalent record pair higher than a nonequivalent record pair when at least one record in the two pairs is from R_a. The case when the record from R_a is one of the records in the equivalent pair is considered in $\mathit{subAUC}_s^{a,A}$ and the case when the record is in the nonequivalent pair is considered in $\mathit{subAUC}_s^{A,a}$. The overlap between these two cases, when records from R_a appear in both equivalent and nonequivalent pairs, is considered in $\mathit{subAUC}_s^{a,a}$ and is subtracted from the sum of the first two cases because it is considered twice in the sum. The weights $w^{a,A}$, $w^{A,a}$, and $w^{a,a}$ depend on the numbers of comparisons between the record pairs that are counted in $c^{A,a}(I)$, $c^{a,A}(I)$, and $c^{a,a}(I)$. In general, $c^{a,b}(I)$ is the number of possible comparisons between the record pairs in I_a^+ and the record pairs in I_b^-. Note that these counts and the weights depend on I, not s or D.

The subAUC represents the performance of s in separating equivalent and nonequivalent record pairs when the pairs include at least one record from the subpopulation R_a. The notion of subAUC gives us a meaningful way to compare the performance of s in different subpopulations. Using subAUC, we formalize bias as the normalized gap between the best and worst performance of s in subpopulations.

Definition 3 (Fair EM and Bias)**.** Given a score function s for record pairs in $R \times R$ with a sensitive attribute A, the bias of EM using s is $\mathit{bias}(s) = 1 - \frac{\mathit{subAUC}_s^{\min}}{\mathit{subAUC}_s^{\max}}$ where $\mathit{subAUC}_s^{\min}$ and $\mathit{subAUC}_s^{\max}$ are the maximum and the minimum (or the best and the worst) subAUCs between the subpopulations defined by the sensitive attribute A.

Given a dataset D, the problem of fair EM is finding the best matcher with a score function with a bias less than a user-specified threshold β. ■

The bias is 0 when s has the same performance in all subpopulations, where the subAUCs measure performance. The bias is significant when there is a large gap between the subAUCs showing disparate impact between the subpopulations. Note that the bias in Definition 3 is a normalized version of the bias we explained in Example 2.

IV. Debiasing Using Data Augmentation

The bias defined in Definition III can be caused by the lack of enough training pairs for some subpopulations or the high complexity of record matching in the subpopulations. To mitigate biases in EM, we apply *data augmentation (DA)*, a common technique in computer vision and NLP that augments training data with new samples to improve the accuracy of ML models (see [30] for a survey). DA is recently applied in data management to improve data preparation, transformation, and cleaning processes [31], [24].

Our debiasing algorithm augments the training dataset with new record pairs generated from the existing record pairs in the dataset. To generate new record pairs, we use Rotom, an existing data augmentation framework that uses NLP and deep learning [24]. Rotom applies operators on the existing examples in a dataset that generate natural and diverse new examples for a supervised ML task. A key feature of Rotom is that it learns which newly generated examples improve the ML model accuracy for the task. The main idea in Rotom is to convert an input example to a sequence of tokens and use Seq2Seq models to generate new sequences.

To incorporate the generate and use new record pairs, we iteratively run the following steps:

1. Divide the input dataset into training and validation datasets.
2. Train the score functions s using the training dataset.
3. Compute the subAUCs for all subpopulations w.r.t. s using the validation data, find the maximum and minimum subAUCs, and compute bias.
4. Generate new records in the k subpopulations with the minimum subAUCs using Rotom. Add the new records to the training dataset and continue with the second step.

The iterations stop when the validation bias is less than or equal to a user-specified threshold or when we augment data for all the subpopulations. In the last step, we only use the worst k subpopulations for which we have not augmented data in the previous iterations.

```
Algorithm 1: $\mathit{Debias}(D, A, \beta, k)$
  Input: A dataset $D$, a sensitive attribute $A$, the bias
         threshold $\beta$, and the number of
         subpopulations $k$.
  Output: A score function $s$.
1 $S \leftarrow \mathit{SensitiveValues}(D, A)$;
2 $S_{aug} \leftarrow \emptyset$;
3 $(D_T, D_V) \leftarrow \mathit{Divide}(D)$
4 $s \leftarrow \mathit{Train}(D_T)$;
5 while $\mathit{bias}(s, D_V) > \beta$ and $S_{aug} \subset S$ do
6     $S_{min} \leftarrow \mathit{FindMinAUCs}(s, D_V, k, S_{aug}))$;
7     $D_\triangle \leftarrow \mathit{AugmentData}(D_T, S_{min})$;
8     $s \leftarrow \mathit{Retrain}(s, D_\triangle)$;
9     $S_{aug} \leftarrow S_{aug} \cup S_{min}$;
10 return $s$;
```

Algorithm 1 shows our debiasing algorithm, *Debias*, that applies Rotom for DA to reduce the bias. The algorithm receives a dataset D, a sensitive feature A that specifies subpopulation, the bias threshold β, and the number of subpopulations k. The output is a score function s. The algorithm starts in Line 1 by initializing S with the sensitive values in D representing the subpopulations. S_{aug}, initialized with $\emptyset$ in Line 2, keeps the subpopulations considered in the DA during the previous iterations. Line 3 divides the dataset into training and validation. The algorithm trains an initial score function s (Line 4), and iteratively applies DA (Line 7) for the subpopulations in S_{min} with the lowest subAUCs. The algorithm finds these subpopulations in Line 6, where the procedure *FindMinAUCs* computes the AUCs for all subpopulations and returns the worst k subpopulations that are not in S_{aug}. Note that computing subAUCs is done using the validation data. The algorithm then retrains s using the augmented data D_{Δ}, and adds the subpopulations S_{min} that are considered in the current iteration to S_{aug}. The iterations continue until the bias is reduced to the acceptable threshold β or all the subpopulations are considered. The algorithm returns the latest trained function s in Line 10.

Computing subAUCs using the validation data is not a trivial task. In Definition 2, the weights and the background AUCs depend on I, which is unknown. We present an extension of *the normalized Mann-Whitney U-Statistic* in Equation 7 to estimate subAUCs using the pairs in D.

$$\widehat{subAUC}_s^a = \frac{c_s^{a,A}(D) + c_s^{A,a}(D) - c_s^{a,a}(D)}{c^a(D)} \tag{7}$$

where $c_s^{a,b}(D)$ is defined as follows:

$$c_s^{a,b}(D) = \sum_{p^+ \in D_a^+} \sum_{p^- \in D_b^-} \mathbb{1}_{s(p^+) \geq s(p^-)} \tag{8}$$

In these estimations, D_a is the subset of the record pairs in D with at least one pair in R_a. D_a^+ and D_a^- are the subsets of equivalent and nonequivalent record pairs in D^a. $\mathbb{1}_{condition}$ is an indicator function that returns 1 if *condition* holds and 0 otherwise. The count value $c^a(D)$ is $c^{a,A}(D) + c^{A,a}(D) - c^{a,a}(D)$ where the function $c^{a,b}$ is the same function defined in Definition 2. Equation 8 counts the number of times s ranks an equivalent record pair in D higher than a nonequivalent record pair in D when at least a record in the equivalent pair is from R_a and at least a record in the nonequivalent record pair is from R_b.

V. Experimental Evaluations

In our experiments, we seek three main objectives. We tune the debiasing algorithm by finding the optimal k for reducing bias in Section V-C1, then we demonstrate the effectiveness of the algorithm in reducing bias for unseen test data in Section V-C2, and we study the impact of data quality on bias in EM in Section V-C3. We start this section by explaining our experimental setting in Section V-A and finish the section with a discussion about the results in Section V-D.

A. Experimental Setup

We implemented the debiasing algorithm with Python 3.7. We used two Python libraries *ethnicolr* [2] and *gender-guesser 0.4.0* [3] for finding ethnicity and gender based on given name and family name and adding these sensitive attributes to the datasets where the sensitive attributes are missing. We also used open-source implementations of Ditto and Rotom.[4][5] The training process runs a fixed number of epochs (10, 15, or 40 depending on the dataset). We randomly split each dataset into training, test, and validation sets with a ratio of 3:1:1. We conducted all experiments on a *ComputeCanada* [6] server with 4 GPUs.

B. Datasets

We use three datasets from the popular ER benchmark in [32] with their statistics in Table III. The numbers of attributes in these datasets range from 4 to 15. In the following, we briefly explain these three datasets and the specific data preparations for the experiments in this work.

Dataset	Records	Split	#Pairs	%Pos
DBLP-GoogleScholar	23,752	Train	17,223	0.229
	9,249	Validation	5,742	
	5,600	Test		
Amazon-Google	2,853	Train	6,874	0.112
	1,838	Validation	2,292	
	2,059	Test		
FEBRL	4,744	Train	2,400	0.333
	1,422	Validation	800	
	1,420	Test		

TABLE III: Datasets statistics

1) DBLP-GoogleScholar: This dataset contains data about papers and their authors [32]. Each paper has five attributes: title, author, venue, year, and ethnicity. Ethnicity is a sensitive attribute and refers to the majority of the authors' ethnicity. We use ethnicolor to predict 13 values for ethnicity.

2) Amazon-Google: This dataset contains records of products from Amazon and Google with three features: title, manufacturer, and price. We use the manufacturer as the sensitive feature to study the quality of record matching for products from different manufacturers. In total, 283 manufacturers define many subpopulations, unlike the other two datasets.

3) FEBRL (Patients): The FEBRL (freely extensible biomedical record linkage) dataset contains patient records obtained from an epidemiological cancer study in Germany.[7] The dataset contains patient records from four sources. It is generated by the Institute for Medical Bio-statistics, Epidemiology, and Informatics (IMBEI). We randomly selected record pairs to generate record pairs for EM. The FEBRL dataset comprises fifteen attributes, including name, address, postal code, and other demographic features. We remove identifiers, such as social security numbers, to make EM nontrivial. The sensitive attribute is ethnicity which we added using ethnicolor.

Table III shows the number of unique records, the number of record pairs, and the percentage of equivalent pairs (i.e., %Pos) for train, validation, and test dataset separately.

C. Experimental Results

We present the experimental results in Sections V-C1-V-C3 and discuss the takeaways in Section V-D.

1) Tuning the debiasing algorithm: To study the tuning parameter k in the algorithm, we use Amazon-Google to run

[2] https://pypi.org/project/ethnicolr/

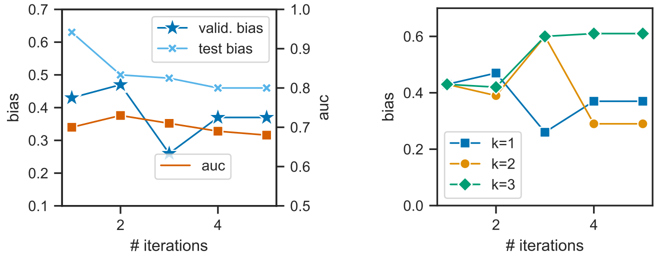
[3] https://pypi.org/project/gender-guesser/

[4] https://github.com/megagonlabs/ditto

[5] https://github.com/megagonlabs/rotom

[6] https://www.computecanada.ca/

[7] https://github.com/J535D165/FEBRL-fork-v0.4.2

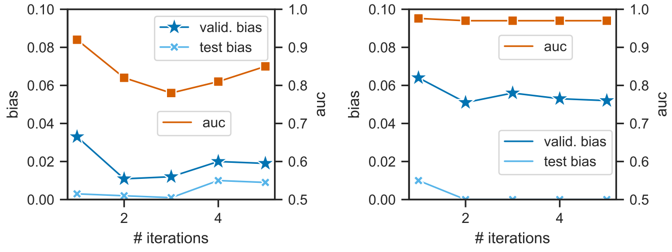


(a) Bias in test and validation

(b) Bias for varying k

Fig. 1: Debiasing in Amazon-Google

the debiasing algorithm with $k = 1, 2, 3$. Figure 1b shows the bias change in the validation data in different iterations of the algorithm. For $k = 2, 3$, the bias fluctuates without any clear downtrend as a sign of a consistent bias reduction. For $k = 1$, the bias consistently decreases in each iteration. The reason for the fluctuation of bias with $k = 2, 3$ is that although DA for one subpopulation improves its subAUC, it might decrease other subpopulations' subAUCs. Therefore, augmentation for multiple subpopulations might cause a drop in subAUCs of several other subpopulations. This does not happen with $k = 1$ as the algorithm retrains after each augmentation and the next subpopulation is decided using the retrained model. Note that running the algorithm with $k = 2, 3$ is faster as more subpopulations are treated in each iteration. This means there is a trade-off between the runtime and the consistency in reducing bias for different values of k. In the other experiments, we use $k = 1$ to obtain consistent results.



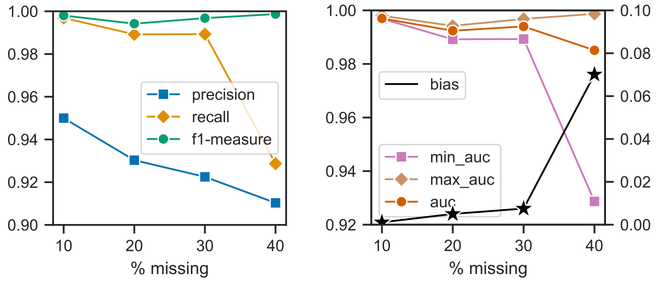
(a) DBLP-GoogleScholar

(b) Patients

Fig. 2: Bias in test and validation

2) Reducing bias in validation and test: Figures 1a, 2b, and 2a show how the bias changes in the validation and the test datasets in 5 iterations of the debiasing algorithm. Overall, the figures show a consistent decrease in bias for both validation and test datasets. The same figures also show the AUCs of EM that reflect the accuracy of EM. The AUCs slightly decrease in each iteration. This is expected because there is often a trade-off between accuracy and fairness, and debiasing is expected to cause a slight decrease in accuracy.

3) Impact of data quality on bias: To study the impact of data quality on EM, we train DITTO with the patient dataset with varying levels of missing values. It is known that low data quality impacts the performance of EM solutions, which we also confirm in our results in this section. Figure 3a plots precision, recall, and F1-measure for DITTO models trained with datasets that have varying percentages of missing values. The figure shows a decrease in these accuracy measures with more missing values. Figure 3b shows AUC, minimum subAUC, maximum subAUC, and bias for the same models. This figure also shows decreasing AUC for the models when there are more missing values. Additionally, there is a sharp increase in bias when missing values are increased. This proves that low data quality and increasing missing values have a negative impact on fairness and increase bias.



(a) Accuracy

(b) Bias and subAUCs

Fig. 3: DITTO's performance with missing values

D. Discussion

The experiments in Section V-C1 and V-C2 proves that the debiasing algorithm can effectively reduce bias in both test and validation dataset if it is properly tuned. There is a trade-off between the amount of DA in each iteration in terms of the number of subpopulations for which DA is applied vs. the consistency in reducing bias. The main takeaway from the experiments in Section V-C3 is that DITTO is resistant to missing values as its overall accuracy slightly decreases with more missing values. However, it is sensitive w.r.t. its bias as more missing values introduces a significant bias. This means low data quality has a different impact on subpopulations and can have a more severe negative effect on specific subpopulations.

VI. CONCLUSION

In this paper, we studied the problem of fair EM to accurately match records within and across subpopulations. We formalized a new bias in EM based on the AUC and the risk of record matching. We used this measure to evaluate the fairness of DITTO, a state-of-the-art EM solution using three datasets. We also presented a debiasing algorithm and show it effectively reduces the bias of DITTO for record matching in real data. A possible future research direction is to formalize and evaluate bias using other notions of bias in ML, e.g., equalized odds and equal opportunity. We used the gap between the best and work subAUCs to define bias. An alternative is to use all the subAUCs. Our debiasing algorithm uses DA which is considered a preprocessing technique. We will study in-processing (e.g., integrating fairness constraints with ML models' loss function) or post-processing (e.g., using clustering techniques, such as correlation clustering). Another future research idea is to study bias in EM solutions other than DITTO.

REFERENCES

[1] V. Efthymiou, K. Stefanidis, E. Pitoura, and V. Christophides, "FairER: entity resolution with fairness constraints," in *CIKM*, 2021, pp. 3004–3008.

[2] A. Karakasidis and E. Pitoura, "Identifying bias in name matching tasks." in *EDBT*, 2019, pp. 626–629.

[3] S. Mishra, S. He, and L. Belli, "Assessing demographic bias in named entity recognition," *arXiv preprint arXiv:2008.03415*, 2020.

[4] S. Biswas and H. Rajan, "Fair preprocessing: Towards understanding compositional fairness of data transformers in machine learning pipeline," in *ESEC/FSE 2021*, 2021, p. 981–993.

[5] L. Mazilu, N. W. Paton, N. Konstantinou, and A. A. Fernandes, "Fairness in data wrangling," in *IRI*, 2020, pp. 341–348.

[6] S. Schelter, Y. He, J. Khilnani, and J. Stoyanovich, "Fairprep: Promoting data to a first-class citizen in studies on fairness-enhancing interventions," in *EDBT*, 2020.

[7] K. Yang, B. Huang, J. Stoyanovich, and S. Schelter, "Fairness-aware instrumentation of preprocessing pipelines for machine learning," in *HILDA*, 2020.

[8] B. Salimi, L. Rodriguez, B. Howe, and D. Suciu, "Interventional fairness: Causal database repair for algorithmic fairness," in *SIGMOD*, 2019, p. 793–810.

[9] G. Papadakis, E. Ioannou, E. Thanos, and T. Palpanas, "The four generations of entity resolution," *Synthesis Lectures on Data Management*, vol. 16, no. 2, pp. 1–170, 2021.

[10] Y. Li, J. Li, Y. Suhara, A. Doan, and W.-C. Tan, "Deep entity matching with pre-trained language models," *PVLDB*, vol. 14, no. 1, p. 50–60, 2020.

[11] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, and V. Raghavendra, "Deep learning for entity matching: A design space exploration," in *SIGMOD*, 2018, pp. 19–34.

[12] M. Ebraheem, S. Thirumuruganathan, S. Joty, M. Ouzzani, and N. Tang, "Distributed representations of tuples for entity resolution," *PVLDB*, vol. 11, no. 11, pp. 1454–1467, 2018.

[13] C. Zhao and Y. He, "Auto-em: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning," in *WWW*, 2019, pp. 2413–2424.

[14] T. Gupta and V. Deshpande, "Entity resolution for maintaining electronic medical record using oyster," in *EAI*. Springer, 2020, pp. 41–50.

[15] E. Krivosheev, M. Atzeni, K. Mirylenka, P. Scotton, C. Miksovic, and A. Zorin, "Business entity matching with siamese graph convolutional networks," 2021.

[16] N. Kallus and A. Zhou, "The fairness of risk scores beyond classification: Bipartite ranking and the xauc metric," *NeurIPS*, vol. 32, 2019.

[17] L. Dixon, J. Li, J. Sorensen, N. Thain, and L. Vasserman, "Measuring and mitigating unintended bias in text classification," in *AIES*, 2018, pp. 67–73.

[18] D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman, "Nuanced metrics for measuring unintended bias with real data for text classification," in *WWW*, 2019, pp. 491–500.

[19] A. Beutel, J. Chen, T. Doshi, H. Qian, L. Wei, Y. Wu, L. Heldt, Z. Zhao, L. Hong, E. H. Chi *et al.*, "Fairness in recommendation ranking through pairwise comparisons," in *SIGKDD*, 2019, pp. 2212–2220.

[20] R. Vogel, A. Bellet, and S. Clémençon, "Learning fair scoring functions: Bipartite ranking under roc-based fairness constraints," in *AISTATS*. PMLR, 2021, pp. 784–792.

[21] Z. Yang, Y. L. Ko, K. R. Varshney, and Y. Ying, "Minimax auc fairness: Efficient algorithm with provable convergence," *arXiv preprint arXiv:2208.10451*, 2022.

[22] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[23] J. Huang and C. X. Ling, "Using auc and accuracy in evaluating learning algorithms," *TKDE*, vol. 17, no. 3, pp. 299–310, 2005.

[24] Z. Miao, Y. Li, and X. Wang, "Rotom: A meta-learned data augmentation framework for entity matching, data cleaning, text classification, and beyond," in *SIGMOD*, 2021, pp. 1303–1316.

[25] M. Milani, Z. Zheng, and F. Chiang, "Currentclean: Spatio-temporal cleaning of stale data," in *ICDE*, 2019, pp. 172–183.

[26] Z. Zheng, T. M. Quach, Z. Jin, F. Chiang, and M. Milani, "Currentclean: interactive change exploration and cleaning of stale data," in *CIKM*, 2019, pp. 2917–2920.

[27] Y. Huang, M. Milani, and F. Chiang, "Pacas: Privacy-aware, data cleaning-as-a-service," in *BigData*, 2018, pp. 1023–1030.

[28] Y. Huang, M. Milani, and F. Chiang, "Privacy-aware data cleaning-as-a-service," *Information Systems*, vol. 94, p. 101608, 2020.

[29] T. Furche, G. Gottlob, L. Libkin, G. Orsi, and N. Paton, "Data wrangling for big data: Challenges and opportunities," in *EDBT*, 2016, pp. 473–478.

[30] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[31] Y. Li, X. Wang, Z. Miao, and W.-C. Tan, "Data augmentation for ml-driven data preparation and integration," *PVLDB*, vol. 14, no. 12, pp. 3182–3185, 2021.

[32] P. Konda, S. Das, A. Doan, A. Ardalan, J. R. Ballard, H. Li, F. Panahi, H. Zhang, J. Naughton, S. Prasad *et al.*, "Magellan: toward building entity matching management systems over data science stacks," *PVLDB*, vol. 9, no. 13, pp. 1581–1584, 2016.