Through the Fairness Lens: Experimental Analysis and Evaluation of Entity Matching [Experiment, Analysis & Benchmark]

Nima Shahbazi
University of Illinois Chicago
nshahb3@uic.edu

Nikola Danevski
University of Rochester
ndanevsk@u.rochester.edu

Fatemeh Nargesian
University of Rochester
fnargesian@rochester.edu

Abolfazl Asudeh
University of Illinois Chicago
asudeh@uic.edu

Divesh Srivastava
AT&T Chief Data Office
divesh@research.att.com

ABSTRACT

Entity matching (EM) is a challenging problem studied by different communities for over half a century. Algorithmic fairness has also become a timely topic to address machine bias and its societal impacts. Despite extensive research on these two topics, little attention has been paid to the fairness of entity matching.

Towards addressing this gap, we perform an extensive experimental evaluation of a variety of EM techniques in this paper. We generated two social datasets from publicly available datasets for the purpose of auditing EM through the lens of fairness. Our findings underscore potential unfairness under two common conditions in real-world societies: (i) when some demographic groups are over-represented, and (ii) when names are more similar in some groups compared to others. Among our many findings, it is noteworthy to mention that

while various fairness definitions are valuable for different settings, due to EM's class imbalance nature, measures such as positive predictive value parity and true positive rate parity are, in general, more capable of revealing EM unfairness.

PVLDB Reference Format:
Nima Shahbazi, Nikola Danevski, Fatemeh Nargesian, Abolfazl Asudeh, and Divesh Srivastava. Through the Fairness Lens: Experimental Analysis and Evaluation of Entity Matching [Experiment, Analysis & Benchmark]. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at github.com/DataIntelligenceCrew/FairEM.

1 INTRODUCTION

Entity matching (EM) seeks to match pairs of entity records from (the same or different) data sources that refer to the same real-world object. EM is very useful in many applications domains, including (a) healthcare, where matching of patient records from different healthcare facilities (e.g., emergency rooms, hospitals, etc.) can be used to determine if they refer to the same real-world person; (b) airline security, where airline passenger records are matched against no-fly list records to identify people who should be prevented from boarding flights or should undergo additional screening; (c) e-commerce, where product records from different retailers' websites can be matched to identify popular products and fraudulent knockoffs; and so on.

EM is a challenging problem that has been extensively investigated for over half a century by different communities, e.g., statistics, databases (DB), natural language processing (NLP), and machine learning (ML), resulting in a variety of techniques proposed in the literature for addressing this problem. These challenges arise because entities in autonomous data sources can be represented in a variety of ways (e.g., highly structured records versus textual descriptions), using different conventions (e.g., the many ways in which person names and postal addresses are represented), data quality issues (e.g., misspellings, missing values), and so on. A consequence is that, despite significant advances in recent years (especially with recent neural techniques like Ditto [30]), EM techniques still result in both false positives (non-matching entity record pairs that are declared as matches) and false negatives (matching record pairs that are declared as non-matches).

These errors can have serious consequences in practice. As a motivating example throughout this paper, consider the airline security application, which aims to identify passengers that are likely to be dangerous (e.g. terrorists) for screening and potential prevention from boarding the flights. Using a dataset of criminal records called the no-fly list, passenger names (and other information) are matched against the no-fly list for this purpose. False positives in airline security can lead to significantly inconveniencing passengers. On the other hand, false negatives can result in known terrorists being permitted to board flights with undesirable consequences. Due to historical biases, the no-fly list datasets could be biased, over-representing some minority groups in comparison to society's population distribution. This, as we shall evaluate in our experiments, can result in higher false positive rates for those demographic groups. Another potential issue is that some countries and demographic (sub-)groups have more similar names. As a result, passengers from those groups may have higher chance of having the same or similar information to those of known terrorists, which in turn will cause higher false positive rates for them.

When such disparities (e.g., false positives) occur in a systematic way for some demographic (sub-)groups thereby disadvantaging them over others, concerns about the *fairness* of EM techniques arise. While the fairness of ML models has been the topic of much

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

recent work in the literature [8, 17, 19, 19, 22, 28, 48, 49, 51], not much attention has been paid to the fairness of EM techniques.

In this paper, we seek to address this gap in the literature and perform an extensive experimental evaluation and analysis of a variety of EM techniques on a range of datasets through the *fairness lens*. Traditionally, a blocking algorithm may precede a matching algorithm aiming to reduce the space of possible matching candidates from quadratic to sub-quadratic, e.g., linear. A rich body of research focuses on blocking algorithms [13, 29, 37, 38]. In this paper, our goal is to audit off-the-shelf entity matching systems used in practice. As such, our evaluation and analysis are performed on end-to-end matching systems which may include their own built-in blocking algorithms.

Summary of Contributions: In summary, we make the following technical contributions in this paper:

- Given the pairwise nature of EM, we propose the use of *single fairness* and *pairwise fairness* to evaluate entity matchers. We adopt 11 popular fairness measures from the literature for this task and analyze their suitability for EM.
- We select a suite of 13 EM techniques (including 1 declarative rule-based technique, 7 non-neural ML techniques, and 5 neural ML techniques), 2 semi-synthetic datasets and 6 benchmark datasets that have been used in prior work on entity matching (including 2 structured datasets, 2 textual datasets and 2 dirty datasets) for fairness evaluation.
- Using publicly available individual-level data, we created two semi-synthetic matching *social datasets*. The datasets are shared publicly as benchmarks for auditing the fairness of matchers.
- We evaluated all the combinations of EM techniques, benchmark datasets, and fairness measures and analyzed the outcomes to obtain generalizable results. We classified the results into four classes of configurations based on whether an (EM technique, benchmark dataset, fairness measure) yielded (i) accurate or inaccurate matching results, and (ii) fair or unfair matching results. Interestingly, all four classes contained configurations involving ML-based classifiers.

Some of our findings in this study are as follows:

- Our results on social data confirm matching unfairness when (i) there are higher similarities among tuples of a certain group, (ii) there is a biased representation of demographic groups in data.
- Our results underscore that responsible EM requires unbiased training data that covers different possibilities from various (demographic) groups.
- While different fairness measures are valuable for different settings, due to the class-imbalance property of EM, measures such as *positive predictive value parity* and *true positive rate parity* are more capable of revealing EM unfairness.
- Significantly relying on proxy attributes such as name, salary, etc., can cause unfairness in non-neural models. On the other hand, relying on pre-trained language models and embeddings, or not fully considering the dataset structures can cause unfairness in neural matchers.

2 RELATED WORKS

Fairness in entity resolution (ER) has briefly been studied in the literature. In [18], a constraint-based formulation for fairness is proposed to mitigate bias in ER tasks by ensuring that all subgroups have the same opportunity to be resolved. Furthermore, [32] proposes a subgroup-based training for different ethnicities in order to increase both accuracy and fairness in SVM-based ER which is consistent with our suggestion to use ensemble learning for EM. Finally, in a parallel work [36], the authors propose an AUC-based fairness definition for EM and ER tasks and try to resolve the bias issues through a data augmentation solution. To the best of our knowledge, we are the first to comprehensively audit entity matching models for fairness and propose proper measures, datasets, and comparison angles fitting the problem settings given the inherent differences with typical machine learning tasks that have so far been studied.

3 FAIRNESS EVALUATION FRAMEWORK

3.1 Background

Given two sets of entities $A \in S_A$ and $B \in S_B$ from data sources S_A and S_B, the EM problem is to identify all correspondences between entities in $A \times B$ that correspond to the same real-world object. A correspondence $c = (e_i, e_j, s)$ interrelates two entities e_i and e_j with a confidence value $s \in [0, 1]$ that indicates the similarity of e_i and e_j or the confidence of a matcher about e_i and e_j referring to the same object. [26]. To decide whether the entity pair of $c = (e_i, e_j, s)$ is a *match* or *non-match*, matchers often apply a threshold on s [7, 47]. We decouple the choice of a threshold from the outcome of the matching and consider the outcome of an EM task as pairs of matching and non-matching entities. Formally, we consider the following EM problem in this paper:

Definition 1 (Entity Matching Problem). *Consider two sets of entities $A \in S_A$ and $B \in S_B$ from data sources S_A and S_B. For every pair of entities $(e_i, e_j) \in A \times B$, let y_{ij} be the ground truth label indicating if e_i and e_j refer to the same object. Given all pairs $(e_i, e_j) \in A \times B$, the EM problem is to predict y_{ij} with a label h_{ij}. That is, h_{ij} refers to the decision of the matcher about the label of e_i and e_j (match or non-match).*

In a fairness-sensitive setting, entities are accompanied with sensitive attributes (e.g. `gender`, `country`, `race`, etc.). Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ be the sensitive attributes, $dom(A_i)$ be the domain of A_i, and $\mathcal{G} = \{g_1, \ldots, g_m\}$ be the set of all groups of interest, i.e. $\mathcal{G} = \bigcup_{A_i \in \mathcal{A}} dom(A_i)$. The mapping $L(e_i)$ relates an entity to its associated groups $G_i \subseteq \mathcal{G}$. In other words, G_i is the group that e_i belongs to.

Given two sets of entities $A \in S_A$ and $B \in S_B$ from data sources S_A and S_B, and the set $[(e_i, e_j, G_i, G_j, h_{ij}, y_{ij})]_{\forall (e_i, e_j) \in A \times B}$, we would like to audit the fairness of a matcher with respect to groups and the combination of groups (subgroups).

3.2 Single and Pairwise Lens

3.2.1 Group Selection. The first step in auditing an entity matcher for fairness is identifying meaningful groups/subgroups in sensitive attributes. An input dataset to a matcher $\mathcal{M}$ includes entity ids, the value (group/subgroup) of each entity for sensitive attributes, the decisions of $\mathcal{M}$, as well as true labels for the entity pairs. Depending on the type, cardinality, and the number of sensitive attributes, multiple fairness cases may happen that are presented Table 1.

The space of groups for a single attribute with binary or multiple values is the domain of the corresponding attribute. In multiple-attributes settings, we can define intersectional subgroups. The

Table 1: Fairness types based on the number and cardinality of sensitive attributes.

Type	Description	Example
Single Attribute w/ Binary Values	Each entity belongs to one of two groups in the attribute domain.	attribute: `gender={male, female}` `group(e) = {female}`
Single attribute w/ multiple exclusive values	Each entity belongs to exactly one group in the attribute domain.	attribute: `gender={male, female, transgender, non-binary, other}` `group(e) = {non-binary}`
Single setwise attribute	Each entity belongs to a subset of values in the attribute domain.	attribute: `genre={Pop, Rock, Jazz}` `group(e) ={Pop, Rock}`
Multiple attributes	Groups could be either one or a combination of the three cases above.	attributes: `genre` and `gender` `group(e) = {male-Pop, male-Rock, male-Jazz}`

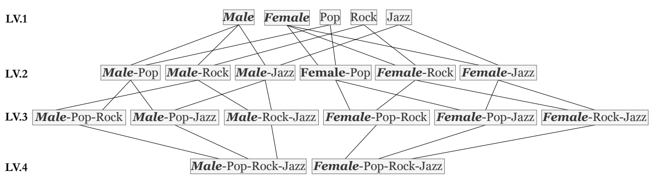

Figure 1: Intersectional subgroup hierarchy for single setwise and multiple attributes

subgroups can be presented in a hierarchical data structure, where the first level includes all groups (of all attributes), while the k-th level includes the set of non-overlapping groups created by combining groups from k different attributes. When one attribute is of the single setwise type, level k includes $k-1$ groups from the setwise attribute with one group from a binary or multi-value attribute.

Example 1: Figure 1 shows the intersectional subgroup hierarchy of sensitive attributes `gender` and `genre` for a dataset that matches songs of different artists. Note that `gender` is a binary attribute and `genre` is a setwise attribute. The level-2 of this hierarchy includes all combinations of groups from `gender` and `genre` in level-1. Level-3 enumerates 2-combinations of the domain of `genre` with groups from `gender`. □

Note that a subgroup hierarchy represents the space of groups and does not mean a data set must or does contain all these groups. In addition to enabling fairness audit on a particular group selected by a user, we allow batch auditing subgroups of each level. That is, a matcher's fairness is evaluated for all subgroups of a particular level selected by a user.

3.2.2 Single and Pairwise Fairness Evaluation. Given the pairwise nature of EM tasks, there are two ways to audit entity matchers:

- *Single Fairness:* The performance of a matcher is evaluated for one subgroup s against either entity in a pair. Given a correspondence $c = (e_i, e_j, h, y)$ and a subgroup s of interest, c is legitimate, if either e_i or e_j belong to subgroups s.
- *Pairwise Fairness:* The performance of a matcher is evaluated for a pair of subgroups s, s' against both entities in a pair. Given a correspondence $c = (e_i, e_j, h, y)$ and a pair of sub-groups (s, s') of interest, c is legitimate, if e_i belongs to s and e_j belongs to s', or vice versa. From an encoding perspective, we concatenate the encodings of subgroups s and s' into a vector c and the encodings (explained in the supplimentary materials) of e_i and e_j into a vector e and validates vector e belongs to c with both directions of $\langle s, s' \rangle$ and $\langle s', s \rangle$.

We consider the EM task to be symmetric in single and pairwise fairness definitions. We remark that these definitions can be extended to ordered single and ordered pairwise fairness where the subgroups are defined on left or right entities. In this paper, we focus on non-directional single and pairwise fairness.

3.3 Fairness Measures

At a high level, fairness definitions can be viewed from three perspectives: group, subgroup, and individual fairness [8]. The most granular notion of fairness is individual fairness that requires similar outcomes for similar individuals [17]. The more popular perspective of fairness, group/subgroup fairness, requires similar treatment for different groups/subgroups. A model/algorithm satisfies some fairness constraints if it has equal or similar performance (according to some fairness measure) on different subgroups. *The focus of this paper is on group/subgroup fairness.* Most of the group fairness measures belong to one of the four categories [5, 8]. (1) *Independence* requires independence of analysis outcome from demographic groups. (2) *Separation* requires independence of the outcome from demographic groups conditioned on the target variable. (3) *Sufficiency* requires independence of the target variable from demographic groups conditioned on the outcome. (4) *Causation* requires that in a counterfactual world, the decision would not change had the individual belonged to a different demographic group. Since we only assume access to a matcher's decisions and true labels, we will not consider Causal fairness in our audit.

In Table 2, we present a suite of fairness measures, based on notions of fairness in classification [8], repurposed for auditing an entity matcher $\mathcal{M}$ for a set $\mathcal{G}$ of (sub)groups. We note that some of the measures cannot be applied in pairwise fairness scenarios where conceptually, the equality of groups restricts matching results. In some scenarios, two entities with different groups can never be considered *match* in the ground-truth. For instance, in a matching task defined between *DBLP* and *ACM* publications, two entities with different venues (after standardization) and years are never a true *match*. More concretely, when pairwise fairness is evaluated on subgroups with non-overlapping groups, TPs and FNs are always zero; therefore, fairness measures that rely on TPs and FNs become inapplicable.

Table 2: **Fairness measures.** $h(e,e')$ **is the output of a matcher** $\mathcal{M}$ **(*match* ('M') or *non-match* ('N')) and** y **is the ground-truth on entities** e **and** e'.

Name	Description	Equation ($\forall g_i \in \mathcal{G}$)
Accuracy Parity (AP)	requires the independence of matchers's accuracy from groups	$Pr(h(e,e') = y \mid g_i) \simeq Pr(h(e,e') = y)$
Statistical Parity (SP)	requires the independence of the matcher from groups	$Pr(h(e,e') = \text{'M'} \mid g_i) \simeq Pr(h(e,e') = \text{'M'})$
[1]**True Positive Rate Parity (TPRP)**	aka *Equal Opportunity*; in the group of true matches requires the independence of match predictions from groups	$Pr(h(e,e') = \text{'M'} \mid g_i, y = \text{'M'}) \simeq Pr(h(e,e') = \text{'M'} \mid y = \text{'M'})$
False Positive Rate Parity (FPRP)	in the group of true non-matches, requires the independence of match predictions from groups	$Pr(h(e,e') = \text{'M'} \mid g_i, y = \text{'N'}) \simeq Pr(h(e,e') = \text{'M'} \mid y = \text{'N'})$
[1]**False Negative Rate Parity (FNRP)**	in the group of true matches, requires the independence of non-match predictions from groups	$Pr(h(e,e') = \text{'N'} \mid g_i, y = \text{'M'}) \simeq Pr(h(e,e') = \text{'N'} \mid y = \text{'M'})$
True Negative Rate Parity (TNRP)	in the group of true non-matches, requires the independence of non-match predictions from groups	$Pr(h(e,e') = \text{'N'} \mid g_i, y = \text{'N'}) \simeq Pr(h(e,e') = \text{'N'} \mid y = \text{'N'})$
[1]**Equalized Odds (EO)**	in both groups of true matches and true non-matches requires the independence of match predictions from groups	$Pr(h(e,e') = \text{'M'} \mid g_i, y = \text{'M'}) \simeq Pr(h(e,e') = \text{'M'} \mid y = \text{'M'})$ $Pr(h(e,e') = \text{'M'} \mid g_i, y = \text{'N'}) \simeq Pr(h(e,e') = \text{'M'} \mid y = \text{'N'})$
[1]**Positive Predictive Value Parity (PPVP)**	among the pairs predicted as match requires the independence of true matches from groups	$Pr(y = \text{'M'} \mid h(e,e') = \text{'M'}, g_i) \simeq Pr(y = \text{'M'} \mid h(e,e') = \text{'M'})$
[1]**Negative Predictive Value Parity (NPVP)**	among the pairs predicted as non-match, requires the independence of true non-matches from groups	$Pr(y = \text{'N'} \mid h(e,e') = \text{'N'}, g_i) \simeq Pr(y = \text{'N'} \mid h(e,e') = \text{'N'})$
[1]**False Discovery Rate Parity (FDRP)**	among the pairs predicted as match, requires the independence of true non-matches from groups	$Pr(y = \text{'N'} \mid g_i, h(e,e') = \text{'M'}) \simeq Pr(y = \text{'N'} \mid h(e,e') = \text{'M'})$
[1]**False Omission Rate Parity (FORP)**	among the pairs predicted as non-match, requires the independence of true matches from groups	$Pr(y = \text{'M'} \mid g_i, h(e,e') = \text{'N'}) \simeq Pr(y = \text{'M'} \mid h(e,e') = \text{'N'})$

3.4 Insights for Selecting Fairness Measures in the Context of Entity Matching

Depending on the context of an EM task at hand, proper fairness measures should be employed. Besides, a major difference between EM and regular classification tasks is that the input to EM tasks is a pair of entities. In the following, we provide insights for selecting fairness measures for EM.

3.4.1 Apriori Insights. Which fairness measures to choose depends on the importance of TPs, FPs, FNs, and TNs in the problem context and how forgiving we can be towards each. Among the fairness measures, *statistical parity* does not consider the ground-truth labels, and requires the independence of the matching prediction from the groups. In simple words, it requires equal match ratios from different groups, independent of whether they really are a match or not. As a result, this measure (and other measures in the independence category) is not reasonable fairness for deduplication tasks using EM. However, it may be useful for EM in table joins to ensure equal representation of different groups in the results. *True* (resp. *false*) *positive rate parity* is useful when correctly predicting the matches is crucial, while false match predictions (resp. correct match predictions) are not costly. Similarly, *true* (resp. *false*) *negative rate parity* is useful when predicting the non-matches correctly is crucial, while false non-match predictions (resp. correct match predictions) are not costly. *Equalized odds*, also known as *positive rate parity*, is a good choice when correctly predicting matches and minimizing false match predictions are both highly important. *Positive* (resp. *negative*) *predictive value parity* is useful when guaranteeing equal chance of correct predictions when predicting the match (resp. non-match) is important. Finally, *false discovery* (resp. *omission*) *rate parity* is a good choice when guaranteeing an equal chance of making a mistake when predicting the match (resp. non-match) is important.

3.4.2 Aposteriori Insights. Due to its pairwise matching nature, *class imbalance* is a distinguishing property of EM, compared to regular classification tasks. To better explain this, let us consider a toy example, where two data sources D and D' contain exactly the same set of n entities. Each pair of entities $e \in D$ and $e' \in D'$ is passed as an input to an entity matcher. In this setting, only n of the n^2 pairs are a match, and the others are non-match. In other words, the probability a random pair is a match is as low as $\frac{1}{n}$. Indeed, blocking techniques [37] can help in reducing the extreme class imbalance. However, even after blocking, class imbalance is expected for EM tasks. Besides, blocking is an engineering step that may or/may-not be applied, independent of the choice of EM technique, while our objective is to evaluate the fairness of EM techniques. Nevertheless, when the input to the EM task is imbalanced and most of the pairs are non-match, some measures are more capable of revealing the unfairness of matchers – as we shall explain in the following. First, note that even a matcher that marks all pairs as non-match has high accuracy in this setting. Subsequently, accuracy parity may not reveal the unfairnesses. Similarly,

[1]This measure is only meaningful for *single* fairness and only extends to *pairwise* fairness cases where sensitive attributes are either setwise or left and right groups are identical.

measures such as FPRP and TNRP may fail to reveal unfairnesses in detecting true matches. In these settings where true matches are considered as *rare events*, a matcher's goal is to successfully discover the matches. Therefore, as explained in § 3.4.1, the fairness measure for successfully discovering these events is **Positive Predictive Value Parity (PPVP)**. Another important measure in this context is **True Positive Predictive Rate Parity (TPRP)**, aka **Equal Opportunity**, which focuses on correct match predictions among the (rare) true matches. This also is consistent with our comprehensive experiments on several data sets, where PPVP and TPRP were the two measures that could reveal the unfairnesses of the matchers. We will further explain this in § 5.

3.5 Measuring Unfairness

Consider a fairness notion and a subgroup $g_i \in \mathcal{G}$. In a perfect situation, the matcher should satisfy the parity (equality) between two probabilities in the following form:

$$\forall g_i \in \mathcal{G}, Pr(\alpha \mid \beta, g_i) = Pr(\alpha \mid \beta) \tag{1}$$

where α and β are specified by the fairness measure. For example, for Positive Predictive Parity, α is $y =$ 'M' and β is $h(e, e') =$ 'M'.

On the other hand, due to the trade-offs [24] between different fairness notions and the impossibilities theorems [12], it is often not possible to satisfy complete parity on all fairness measures. As a result, considering a threshold value (e.g., the 20% rule [19] suggests the threshold as 0.2), the objective is to make sure that *disparity* (as known as *unfairness*) is less than the threshold. Given a fairness notion and a subgroup $g_i \in \mathcal{G}$, the disparity can be computed using subtraction [9], as follows:

$$F^{(s)}_{\alpha,\beta}(g_i) = \max\Big(0\,,\ Pr(\alpha \mid \beta) - Pr(\alpha \mid \beta, g_i)\Big) \tag{2}$$

For example, for accuracy parity (α is $h(e, e') = y$ and β is null) the disparity can be computed as

$$F^{(s)}_{\text{AP}}(g_i) = \max\Big(0\,,\ Pr(h(e, e') = y) - Pr(h(e, e') = y \mid g_i)\Big)$$

Note that if the accuracy for the subgroup g_i is higher than the average accuracy of the model, it is not considered as unfairness. Also, note that Equation 2 considers the higher the probability, the better. Depending on fairness measures (and application), the direction may be as the lower the probability, the better. For example, for FNRP, a lower probability of the false negative is preferred. For such cases, one should consider $Pr(h(e, e') = y \mid g_i) - Pr(h(e, e') = y)$ when computing disparity. As a result, for false negative rate (α is $h(e, e') =$ 'N' and β is $y =$ 'M') the disparity can be computed as

$$\begin{aligned} F^{(s)}_{\text{FNRP}}(g_i) = \max\Big(0\,, & \,Pr(h(e, e') = 0 \mid y = \text{'}M\text{'}, g_i) \\ & - Pr(h(e, e') = 0 \mid y = \text{'}M\text{'})\Big) \end{aligned} \tag{3}$$

Alternatively, given a fairness notion and a subgroup $g_i \in \mathcal{G}$, the disparity can be computed using division [19] , as following:

$$F^{(d)}_{\alpha,\beta}(g_i) = \max\Big(0\,, 1 - \frac{Pr(\alpha \mid \beta, g_i)}{Pr(\alpha \mid \beta)}\Big) \tag{4}$$

Similar to Equation 2, Equation 4 also considers the higher the probabilities the better. For the cases (such as FNRP or FDRP) where the lower probabilities are better, one should swap the nominator and the denominator in the equation. Therefore, for false discovery rate (α is $y = 0$ and β is $h(x) = 1$) the disparity can be computed as

$$F^{(d)}_{\text{FDRP}}(g_i) = \max\Big(0\,, 1 - \frac{Pr(y = \text{'}N\text{'} \mid h(e, e') = \text{'}M\text{'})}{Pr(y = \text{'}N\text{'} \mid h(e, e') = \text{'}M\text{'}, g_i)}\Big)$$

Our proposal in this paper is agnostic to the choice of operation for computing the disparities. Still, in our experiments, without any preference, we use subtraction to compute the disparities.

4 ENTITY MATCHING APPROACHES

The existing techniques for EM fall into one of the following three categories: 1) declarative rule-based, 2) ML-based, and 3) crowd-sourcing-based approaches. The last class of techniques relies on crowd-worker knowledge for EM tasks and we do not include them in our analysis. From each of the remaining categories, we select a few important matchers to be assessed for fairness. The specifications of the evaluated matchers are presented in Table 3.

4.1 Rule-based Matchers

Rule-based approaches perform EM based on the conjunction/disjunction of a few logical predicates, each specifying a matching condition. Each matching condition consists of a similarity measure (e.g., Hamming, cosine, Levenshtein, Affine, Jaccard, etc.) computed between entity pair columns, a comparison operator (e.g., <, =, >), and a threshold value specifying the similarity value. Rule-based matchers are scalable to large settings and provide results that are explainable. However, they highly depend on human experts with relevant domain knowledge to assist with rule specification.

4.2 ML-based Matchers

A crucial part of rule-based matching that affects the overall correctness of the task is the selection and configuration of the rules used for comparison. This task is difficult and laborious even for domain experts. ML-based supervised EM approaches reduce the associated manual labor by benefiting from the training data at hand. They significantly reduce the rule discovery efforts by extracting fitting parameters (e.g., model weights) from the data. However, preparing the training data itself imposes an additional cost. Furthermore, such techniques are computationally expensive (demanding a blocking phase to reduce the search space) and are less explainable on account of using black-box classification methods. Depending on the employed classification technique, ML-based matchers belong to one of the *non-neural* or *neural* categories.

4.2.1 Non-neural Matchers. This category of matchers uses traditional ML algorithms such as decision tree, SVM, etc., to decide whether or not a pair of entities is a match. Since the number of meaningful insights that can be extracted from data and fed as features to the learning algorithm is limited to word-level similarity metrics and TF-IDF scores, non-neural matchers may not perform well for cases where datasets are less structured, and column values are more in a textual format consisting of long spans of text.

4.2.2 Neural Matchers. Deep learning techniques have recently shown promising results in NLP applications. Due to the growing demand for matching textual data instances, it only makes sense to adopt such techniques where the other approaches usually fall short. Deep learning methods transform text into numerical values using character/word embeddings often through pre-trained embedding models such as word2vec [33], GloVe [39], fastText [10]. Due to the sequential nature of text, to better capture the semantics of the data, sequence models such as RNN and its variants (e.g., LSTM, GRU, etc.), where prior sequences of inputs can affect the current input and output, are utilized [7]. Further improvement mechanisms such as attention [44], pre-trained language models [15], domain knowledge injection, data augmentation, summarization, etc., deliver further insights into the models to make better matching decisions. The superiority of neural matchers for textual and dirty data sets has been pointed out in the existing research [34]. However, there are associated challenges, such as high computation costs and large training data requirements, making them not suitable for every EM scenario.

5 EVALUATION AND ANALYSIS

5.1 Evaluation Plan

To evaluate the matchers for fairness, we investigate the performance of matchers in terms of single and pairwise fairness for all valid subgroups in the datasets w.r.t. a variety of fairness definitions. To present a side-by-side comparison and visualization, we aggregate the results based on the dataset and the type of fairness (i.e., single and pairwise). Next, we look into some of the identified discriminated subgroups from different settings and investigate the reasoning behind the unfair behavior of matchers.

5.1.1 Experimental Settings. We conducted the experiments on a 3.5 GHz Intel Core i9 processor, 128 GB memory, running Ubuntu. The evaluation framework was implemented in Python. We accessed the source code of the entity matchers either through the authors' public GitHub or by directly contacting the authors.

5.1.2 Social Datasets. The concept of fairness holds significant societal implications and carries more significance when studied on the *individual records*. Unfortunately, public access to such data, especially the demographic information, is restricted owing to privacy concerns. Therefore, we construct semi-synthetic datasets based on two publicly available real-world datasets CSRankings and Compas [2]. We selected these datasets based on our airline security example discussed in the introduction. Particularly, we want to evaluate the fairness of the matchers under two conditions: (a) *when two demographic groups have different degree of similarities in their names*, and (b) *when there is an over-representation of some groups in the data.* CSRankings[2] is a global ranking system that evaluates computer science departments based on the scholarly research activities of their faculty members from universities across the world. CSRankings dataset is publicly available [3]. For each faculty, in addition to their names, the dataset contains other information such as affiliation country. Having observed various name similarities between different geographical regions, we found this dataset as a good candidate for evaluating (a). Compas, on the other hand, is a public dataset of criminal records that has been widely used in Fair ML research. In addition to names and other information, the dataset contains demographic information for each individual. The dataset over-represents `Black/African-Americans`, which makes it a good candidate for evaluating (b).

[2]csrankings.org

To create our first EM dataset FacultyMatch based on CSRankings, we do the following steps:

- Using `fullName` and `country` for matching, we only focus on two demographic groups of faculties working in Germany `de` and China `cn`.
- We perform a Cartesian product on the sample and label each pair as a match if left and right entities have identical `scholarIDs`.
- We perturb the values of `fullName` column for the right-side entities which involve randomly adding, removing, or replacing a random character in the cell.

Next, following our motivating example in the introduction, we create NoFlyCompas, a no-fly list scenario based on Compas:

- Using `firstName`, `lastName`, and `race` for matching, we focus only on individuals that are either `Caucasian` or `Black/African-American`.
- We create the no-fly list by taking a uniform sample from Compas comprising of 48% `Caucasian` records and 52% `Black/African-American` (the distribution of the two groups in the Compas dataset).
- In accordance with the racial distribution of the U.S. population as reported by the Census Bureau Data [1], we create a passenger table by taking a sample from Compas that includes 80% `Caucasian` and 20% `Black/African-American` individuals.
- We perform a Cartesian product on the two tables and label each pair as a match if left and right entities have identical `personIDs`.
- Similar to the aforementioned process for FacultyMatch, the right entities (that correspond to the No Fly List table) undergo perturbation in the `firstName` and `lastName` columns.

5.1.3 Complimentary Datasets. Data in the context of EM tasks usually fall into one of the following categories:

- *Structured:* In this category of datasets, attribute values are atomic, meaning that they can not be broken into multiple values. Furthermore, there are no missing values in the data.
- *Dirty:* This category of datasets is similar to structured datasets; however, they include far too many random missing values in their columns. Therefore an attribute value may appear for an entity while it does not exist for another one.
- *Textual:* Textual datasets are made of a single attribute per entity containing a textual description.

For the completeness of our experiments, we select several datasets from each category on which we evaluate the matchers. The complementary datasets are chosen from WDC [40] and Magellan [34] repositories which are the standard benchmark corpora used in EM literature. Aside from the dataset type, we carefully handpicked the datasets w.r.t. domain, sensitive attribute type, and ground-truth class balance to cover a variety of possible settings. For the textual datasets Shoes and Cameras, we extract the manufacturer of the corresponding product from the description as the sensitive attribute. Table 4 shows the details of the selected datasets.

5.1.4 Entity Matchers. To cover the breadth of existing methods in our experiments, we picked 13 EM tools from each of the discussed approaches (1 rule-based, 7 non-neural, and 5 neural). The selection criteria included the public availability and error-free execution of the source codes. To ensure the satisfactory performance of the entity matchers, we took the following steps:

Table 3: List of EM approaches evaluated for fairness

Name	Type	Description
BooleanRuleMatcher [25]	Rule-based	Conjunction of rules defined using a similarity measure, a comparison operator, and a threshold value between the entity pair columns, part of Magellan framework
Dedupe [21]	Non-neural	Uses regularized logistic regression for agglomerative hierarchical clustering of entities
DTMatcher [25]	Non-neural	Uses decision tree classifier for matching, part of Magellan framework
SVMMatcher [25]	Non-neural	Uses SVM classifier for matching, part of Magellan framework
RFMatcher [25]	Non-neural	Uses random forest classifier for matching, part of Magellan framework
LogRegMatcher [25]	Non-neural	Uses logistic regression classifier for matching, part of Magellan framework
LinRegMatcher [25]	Non-neural	Uses linear regression classifier for matching, part of Magellan framework
NBMatcher [25]	Non-neural	Uses naive bayes classifier for matching, part of Magellan framework
DeepMatcher [34]	Neural	Provides a variety of deep learning approaches such as aggregation-based, RNN-based, attention-based and, hybrid (RNN+attention) to learn latent semantic features for a pair of entities
Ditto [30]	Neural	Deep learning approach utilizing pre-trained transformer-based language models and optimizing performance using domain knowledge injection, text summarization, and data augmentation techniques
GNEM [11]	Neural	One-to-set neural framework (in contrast to the remaining pairwise solutions) benefiting from graph neural networks
HierMatcher [20]	Neural	Deep learning approach based on RNN, attribute-aware attention mechanism and cross attribute token alignment, built on top of DeepMatcher framework
MCAN [50]	Neural	Deep learning approach based on RNN and multi-context attention mechanisms such as self-attention, pair-attention, global-attention, and gating mechanism, built on top of DeepMatcher framework

Table 4: overview of the datasets used in our analysis

Name	Repository	Domain	Type	Train	Test	% Pos.	# Attr.	Sens. Attr.	Sens. Attr. Type
FacultyMatch		Population	Structured	271108	1084432	0.21%	2	`country`	Single attr. w/ binary values
NoFlyCompas		Population	Structured	20122	75459	0.63%	3	`race`	Single attr. w/ binary values
iTunes-Amazon	Magellan	Music	Structured	321	109	24.7%	8	`genre`	Single setwise attr.
Dblp-Acm	Magellan	Publications	Structured	7417	2473	17.9%	4	`venue`	Single attr. w/ multiple exclusive values
Dblp-Scholar	Magellan	Publications	Dirty	225	100	19%	10	`entry type`	Single attr. w/ multiple exclusive values
Cricket	Magellan	Sports	Dirty	2277	1013	96.5%	20	`batting style`	Single attr. w/ binary values
Shoes	WDC	Products	Textual	24111	10717	10.3%	1	`company`	Single attr. w/ multiple exclusive values
Cameras	WDC	Products	Textual	5476	2434	17.2%	1	`company`	Single attr. w/ multiple exclusive values

BooleanRuleMatcher. We used the automatic feature generation tool provided in the Magellan library to extract features based on the similarity of the columns in the input table w.r.t. multiple distance measures. Next, we handpick some of the generated features based on which we declare matching conditions. Depending on the attribute involved in the generated features, we either use the exact match of the attribute values (for attributes with short and atomic values, e.g., *year*) or similarity of greater than 0.5 (for attributes with longer values, e.g., *paper title*).

Non-neural Matchers. For all non-neural matchers except for Dedupe, we used the automatic feature generation tool in the Magellan library. Next, all of the generated features are fed to the models for training. Dedupe's active learning component requires manual labeling of difficult entity pairs, which is an uphill task. To bypass this step, we converted the training data into Dedupe's generated cache file format and utilized the entire training samples to keep the experiment consistent with the other matchers. Finally, Dedupe did not scale for FacultyMatch, NoFlyCompas, Shoes and Cameras.

Neural Matchers. We tuned the hyper-parameters of all the matchers according to their results on the validation set. For DeepMatcher, HierMatcher, and Mcan we trained the models for 10 epochs with a batch size of 16 and used *fastText* [10] pre-trained word embeddings. We used the *hybrid* model of DeepMatcher that reportedly performs superior compared to the other models. For HierMatcher, we used the attribute-aware attention mechanism. For Mcan, we utilized self-attention, pair-attention, global-attention,

Table 5: NoFlyCompas **results**

Matcher	TPR Afr.	TPR Cauc.	Disparity sub	Disparity div	FDR Afr.	FDR Cauc.	Disparity sub	Disparity div
DeepMatcher	0.89	0.86	-0.03	-0.03	0.20	0.18	0.02	0.11
Ditto	0.76	0.82	0.06	0.08	0.31	0.22	0.09	0.41
GNEM	0.84	0.84	0.00	0.00	0.17	0.09	0.08	0.88
HierMatcher	0.72	0.74	0.02	0.10	0.22	0.16	0.06	0.38
MCAN	0.54	0.57	0.03	0.05	0.19	0.05	0.14	2.8

and gating mechanisms that reportedly would achieve the best results. For Gnem, we trained the GCN models for 10 epochs with a batch size of 2 and 768 nodes at each layer. For Ditto, we trained the models for 40 epochs with a batch size of 64 while using the DistilBERT language model and optimizations such as data augmentation, sequence summarization, and domain knowledge injection.

For all datasets except Cricket, we declare a pair of entities as a "match" if the similarity between the two is greater than 0.5. For the Cricket dataset, due to the high similarity of all pairs, we had to choose a higher similarity threshold of 0.9 because otherwise, all of the models would predict all pairs as "match", which would affect the models' correctness.

As for the fairness threshold, we follow EEOC's 80% rule [14], that only 20% disparity is tolerated.

5.2 Experiment Results on Social Cases

5.2.1 NoFlyCompas. We begin our experiments by evaluating marchers fairness on our NoFlyCompas dataset. Recall that NoFlyCompas dataset is the matching between the no fly list and the passengers list, where the two lists have different distributions of the demographic groups. In particular, while in the U.S. population (passenger list) the White population (75%) is significantly higher than Black (13%), in the no fly list Blacks are over represented and the White and Black ratios are almost the same. It is common for a no-fly list to suffer from sampling bias, as such lists are known to have more individuals from certain regions linked to terrorism.

The experiment results for the case of single fairness are reported in Table 5 and Figure 2. Due to the disjoint nature of the binary-sensitive attribute `race` in our comparison, single and pairwise fairness results are identical and therefore we only report the single fairness results. The first observation we make is the *superiority of non-neural matchers* over neural matchers for this task. This has previously been shown in [34] that for structured datasets, the majority of non-neural matchers perform on par with or outperform the neural matchers. Next, by looking into the neural matchers in Table 5, we see a significant disparity against `African-American` group. More specifically, in terms of **FDR** the `African-American` to `Caucasian` ratio is between 1.11 to **3.8** (280% larger) across different matchers. This translates to *a significantly higher chance of preventing an* `African-American` *person to board a flight* or enter a country compared to a `Caucasian` person.

To better illustrate the root cause behind the disparity, let us highlight the following case that is falsely labeled as "match" by Ditto:

(left entity) **firstName:** `James` **lastName:** `Brown` **race:** `African-American`
(right entity) **firstName:** `Samanthai` **lastName:** `Browne` **race:** `African-American`

Some names are more common within certain demographic groups than others. For example, last names that are very common among black people include Brown, Jackson, Williams, Johnson, etc. Since the no-fly list in our NoFlyCompas dataset over-represents the Black group, for an individual in this group there is a higher chance of getting falsely labeled as a match.

5.2.2 FacultyMatch. Having seen the impact of disproportionate representation of records in data w.r.t. the sensitive attributes, we now turn our attention to another type of discrimination attributable to EM systems. Consider the FacultyMatch dataset described in section 5.1.2. There are 2,061 Chinese `cn` faculty members in this dataset compared to 1,595 German `de` ones. Therefore, when we create the EM dataset by performing the Cartesian product, the group of Chinese faculty members has the larger population in the dataset. To increase the population gap even wider, we remove 80% of the non-match pairs that have a German faculty member either on the right side or the left side. As a result, the number of Chinese pairs becomes more than 6 times the number of German pairs in the final sample ensuring proper representation. Next, using a variety of matchers, we conduct the matching task on the data and audit the matchers for fairness. The results [3] are presented in Table 6 and Figure 3, which imply that neural matchers are consistently unfair toward Chinese `cn` faculties. By looking into the results, we observe that matchers are somewhere between 11% to 22% more prone to make an erroneous positive prediction (match) for the Chinese `cn` group. By manually investigating the false-positive cases, we observed that they mostly include names that are very similar in the English transcription. An example of such cases (FP by Ditto) is brought in the following:

(left entity) **fullName:** `Qingming Huang` **country:** `cn`
(right entity) **fullName:** `Qing-Hu Huang` **country:** `cn`

Furthermore, the models make somewhere between 44% to 75% more mistakes in terms of false-negative predictions for the Chinese `cn` group. Due to the higher degrees of similarities in Chinese names, models in general become more sensitive to minor differences and tend to mismatch. An example of such cases consider the following:

(left entity) **fullName:** `LinLin Shen` **country:** `cn`
(right entity) **fullName:** `Linlin phen` **country:** `cn`

5.3 Comprehensive Evaluation Results

Having discussed the experiment results for social cases, in this section we provide our comprehensive evaluation of the matchers' fairness and correctness using the complimentary benchmark datasets. Due to space limitations, the results for some of the datasets such as iTunes-Amazon (pairwise fairness), Cameras (pairwise fairness), and Shoes (single and pairwise fairness) have been moved to the technical report [42].

5.3.1 Correctness. In summary, through our extensive experiments, we observed that *neural matchers are more accurate than*

[3] Non-neural matchers and rule-based matcher results are not reported due to either perfect (fair) or poor (faulty) overall performance.

Table 6: FacultyMatch results

	TPR		Disparity		PPV		Disparity	
Matcher	cn	de	**sub**	**div**	cn	de	**sub**	**div**
DeepMatcher	0.48	0.72	0.23	0.50	0.79	0.87	0.08	0.11
Ditto	0.59	0.85	0.26	0.44	0.77	0.94	0.17	0.22
GNEM	0.78	0.90	0.12	0.15	0.83	0.92	0.08	0.11
HierMatcher	0.47	0.78	0.31	0.66	0.78	0.89	0.11	0.14
MCAN	0.40	0.70	0.30	0.75	0.86	0.94	0.08	0.09

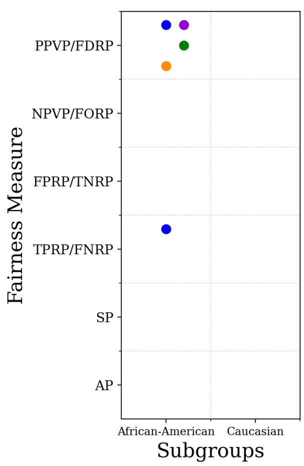

Figure 2: NoFlyCompas: Single Fairness; plot markers provided in Figure 4.

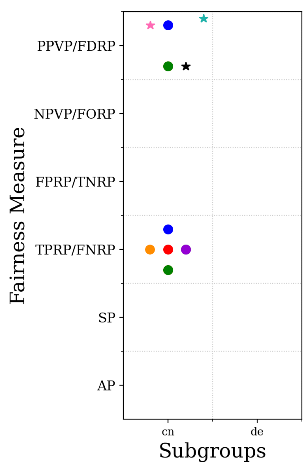

Figure 3: FacultyMatch: Single Fairness

non-neural matchers on textual and dirty data. Modern neural matchers draw on external knowledge by incorporating language models, which helps a matcher to learn the relevance of entities despite the lack of structure and syntactic similarity in text entities. This result is consistent with what is reported by the state-of-art matchers. On the other hand, our results corroborate that *non-neural matchers are more accurate than neural matchers on structured data.* Finally, the flip side of correctness and fairness also exists in EM as some matchers have a low accuracy and F-1 score, while no unfairness issue is observed. This can be explained by the low accuracy of these matchers for all groups across the board which makes the disparity a low value. For more comprehensive results on the overall performance of the matchers across the datasets, fairness and accuracy synergies, and detailed discussions on the correctness of the matchers, please see our technical report [42].

[2]For the evaluation of ML-based matchers, we used random train/test splits from the datasets published by Magellan [25]. To be consistent, all matchers are evaluated in a standard framework against the same datasets. We acknowledge that these results may not exactly match the accuracy results reported by matchers' papers.

[3]Across all plots, Equalized Odds (EO) is the union of FPRP and TPRP rows. That is a matcher that is appears either in row 3 or row 4 of any column is unfair from EO perspective.

5.3.2 Fairness: Measure Types. Some fairness measures are not applicable to EM, and some are more capable of revealing the unfairness of matcher. We have extensively discussed this point in § 3.4. Here, we bring some empirical evidence of such scenarios.

First, as we observe in the majority of experiments, PPVP and TPRP are the measures that discover unfairness the most across all datasets and matchers. Second, it is not the case that one measure fits all. When data has match/non-match negative imbalance, i.e., the number of matching pairs is much higher than non-matching pairs in the ground truth, NPVP and FPRP are the most appropriate measures. This is because while the majority of pairs are positive instances, the failure of a matcher in identifying non-matches makes it unfair to certain groups. Consider the Cricket dataset that contains a larger number of pairs of matching cricket batters than non-matching batters. As we observe in Figure 10, NPVP allows us to detect the unfairness of a matcher such as LogRegMatcher to left-handed batters due to the large number of FNs generated by this matcher. Third, SP could potentially identify false unfairness when the underlying data has label bias. Recall SP does not consider the ground-truth labels and requires the independence of the matching prediction from the groups. In other words, SP requires equal match ratios from different groups, independent of whether they really are match or not. Then, when the ground truth has match/non-match imbalance for a group, that is, the ratio of matched pairs to unmatched ones, is low, the SP measure falsely identifies a matcher as unfair for that group. An example of this phenomenon can be observed in Figure 7, for `French-Pop` group in the iTunes-Amazon dataset, where SP unfairness is indeed due to the fact that the ground truth only contains TNs.

Some measures can be explained by others. For example, let us consider the AP unfairness of Gnem on iTunes-Amazon for the group of country genres, including `Country`, `Cont. Country`, and `Honky Tonk`, reported in Figure 7. This matcher has low accuracy for this group of genres because it identifies a small number of true matches (i.e., has a low number of TPs, thus, suffers from TPRP). Instead, the matchers falsely identify many pairs as non-match (i.e., has a high number of FPs, thus, suffers from NPVP). Similarly, we observe that HierMatcher demonstrates AP unfairness on iTunes-Amazon for the group of country genres because it incurs a large number of FPs, thus, suffers from FPRP unfairness.

Single unfairness can potentially propagate to pairwise fairness. In Figure 10 and 11, we observe that the unfairness of LogRegMatcher for the single `Left Handed` group incurs its unfairness for the pairwise `Left Handed-Left Handed` groups because, in fact, most likely only a left-handed batter can be matched with another left-handed batter.

5.3.3 Fairness: Matcher Types. **Neural Matchers:** *Neural matchers demonstrate more unfairness on structured datasets than non-neural matchers*, as shown in Figure 7 and 8. One reason is that matchers such as Ditto merge the content of different attributes as a single block and use token similarity as a signal for matching. However, for structured data, this technique may lose the important information specified by the structure. In particular, in the following example from Dblp-Acm dataset, the two entities have similar titles and are predicted as match despite the fact that they are (i) written by different authors, (ii) published in different venues, and (iii) published in different years.

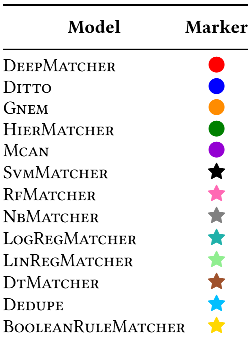

Figure 4: plot markers for the entity matchers

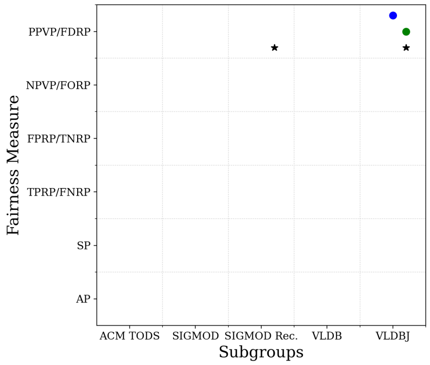

Figure 5: DBLP-ACM: Single Fairness[4]

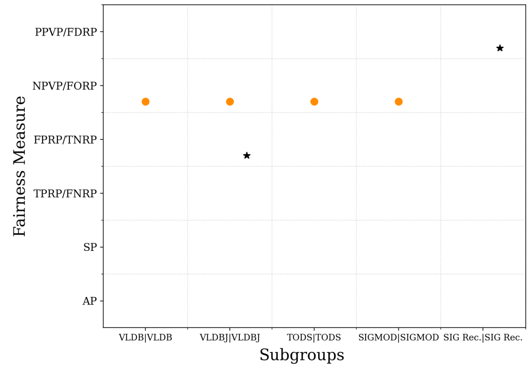

Figure 6: DBLP-ACM: Pairwise Fairness

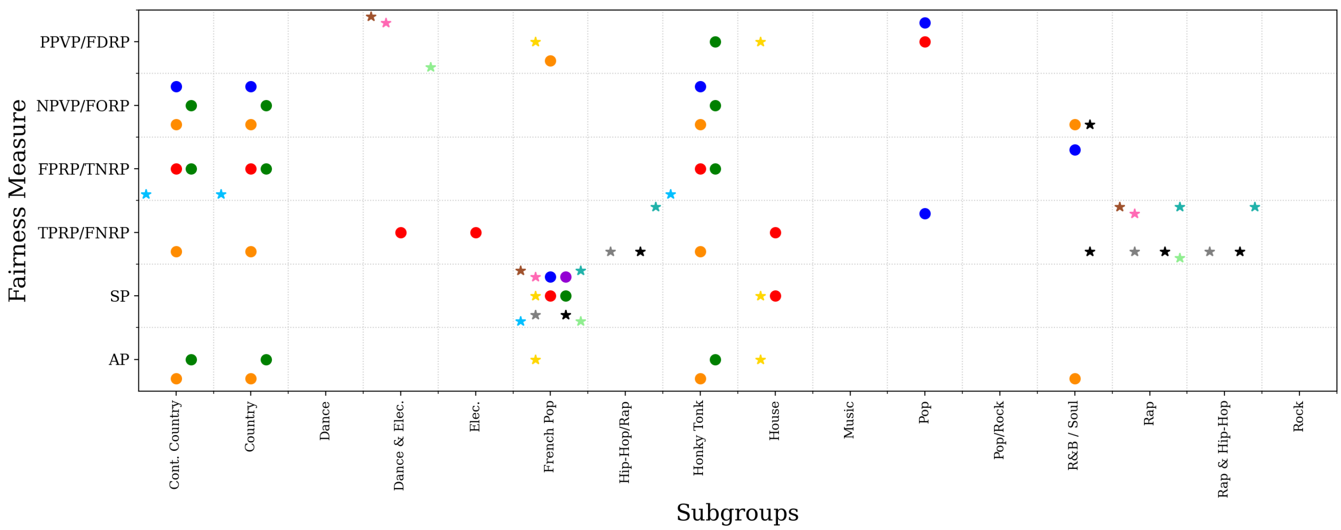

Figure 7: iTunes-Amazon: Single Fairness

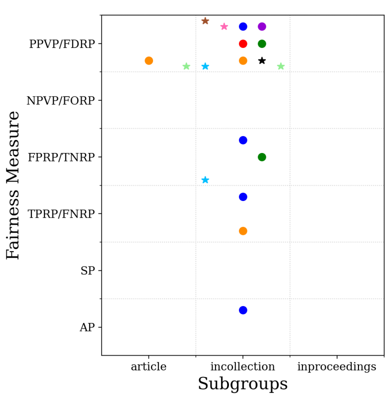

Figure 8: DBLP-Scholar: Single

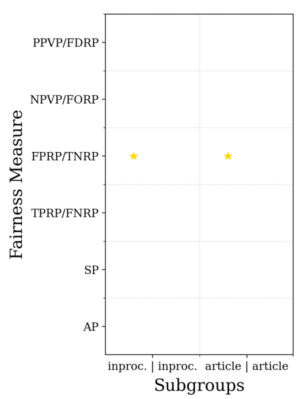

Figure 9: DBLP-Scholar: Pairwise

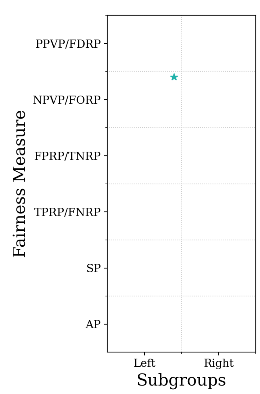

Figure 10: Cricket: Single

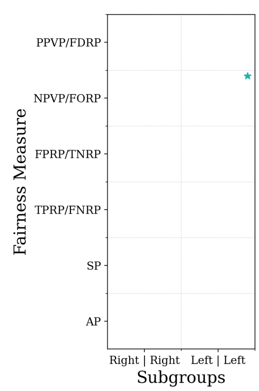

Figure 11: Cricket: Pairwise

(left entity) **title:** `lineage tracing for general data warehouse transformations`; **author:** `jennifer widom , yingwei cui`; **venue:** `VLDBJ`; **year:** `2003`
(right entity) **title:** `data extraction and transformation for the data warehouse`; **author:** `case squire`; **venue:** `SIGMOD`; **year:** `1995`

One of the reasons Ditto was unfair for `VLDBJ` is that, similar to the following example, it is common to publish extended versions of previously published papers in this venue. As a result, after merging different attributes as a block of text for each entity, similar titles

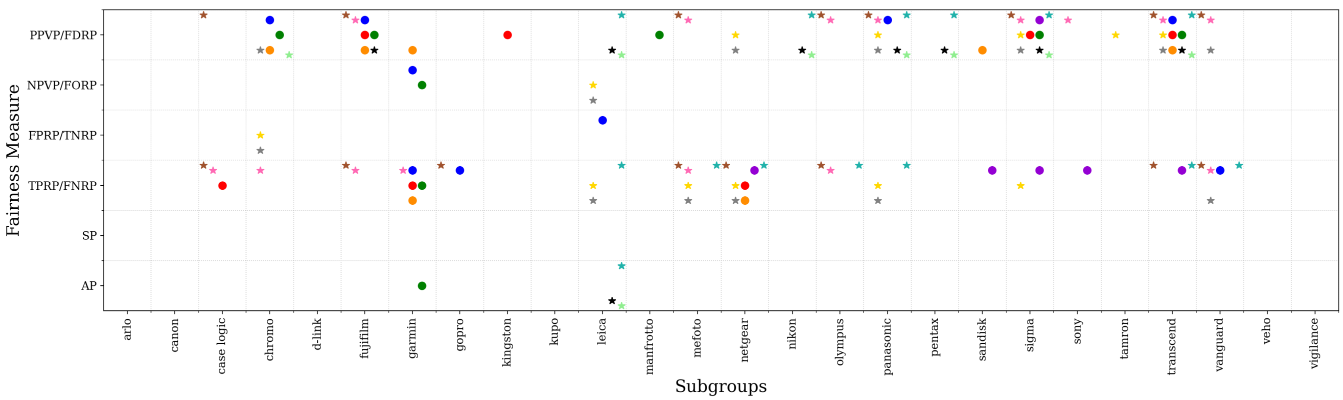

Figure 12: CAMERAS: Single Fairness

and authors may cause enough similarity between the two phrases that the DITTO mistakenly predicts them as a match.

(left entity) **title:** efficient schemes for managing multiversionxml documents; **author:** shu-yao chien , carlo zaniolo , vassilis j. tsotras; **venue:** VLDBJ; **year:** 2002
(right entity) **title:** efficient management of multiversion documents by object referencing; **author:** shu-yao chien , vassilis j. tsotras , carlo zaniolo; **venue:** VLDB; **year:** 2001

External bias could be injected into neural through the use of language models and word embeddings.

For example, HIERMATCHER uses language models and *word embeddings* to compare the attribute similarities of entities. As a result, it may mistakenly match articles with similar titles. Below is an FP example for HIERMATCHER. Both articles are published in the same year. But they appear in different venues and are written by different authors. Still, language models find sufficient similarity between titles to persuade the matcher to label the entities as a match. Perhaps this is because of the similarity of words like "efficient" and "effective" in the embedding space.

(left entity) **title:** efficient and cost-effective techniques for browsing and indexing large video databases; **author:** kien a. hua , jung-hwan oh; **venue:** SIGMOD; **year:** 2000
(right entity) **title:** effective timestamping in databases; **author:** kristian torp , christian s. jensen , richard thomas snodgrass; **venue:** VLDBJ; **year:** 2000

Another example we bring is from ITUNES-AMAZON dataset. The following pair entities is an FP by DITTO. First, both songs are by Kenny Chesney. But more importantly, using a pre-trained language model, `Likes Me` and `Loves Me` are considered (almost) identical. As a result, the model mistakenly labeled the left and right songs as a match. Interestingly, such cases happen to be more frequent in genres such `Country`, resulting in FPRP unfairness for those groups, as shown in Figure 7.

(left entity) **song:** Tequila Loves Me; **artist:** K. Chesney
(right entity) **song:** Likes Me; **artist:** K. Chesney

Our fourth example is from the CAMERAS dataset. In this dataset, camera entities are matched based on their descriptions. A successful matcher on a dataset that includes descriptions in many languages requires extensive coverage of language models on various languages. For example, MCAN returns the following pair of entities as an FN, although the model and the brand match, and *Prijzen* is the Dutch translation of word *Prices*. We suspect that this is due to the poor coverage of word embeddings on the Dutch language.

(left entity) **title:** Sony Cyber-shot RX100@en RX100 Prices - CNET@en
(right entity) **tile:** Sony Cyber-shot RX100 Zwart - Prijzen @NL Tweakers@NL

One model does not fit all. In ITUNES-AMAZON dataset, an interesting observation is that neural matchers perform poorly for the class of country (because a neural matcher creates a curvy decision boundary for all groups and fails for easy groups), while non-neural matchers perform poorly for the class of rap (because non-neural matchers make simple decision boundaries which may not work for a difficult group such as the class of rap genres).

For setwise attributes, matchers demonstrate similar unfair behavior on groups with overlapping semantics. In practice, we observe that, in single setwise sensitive attributes, different sets of groups highly overlap. This is sometimes due to the existence of a semantic hierarchy of groups. For example, in the ITUNES-AMAZON dataset, `Honky Tonk` and `Cont. Country` are a subclass of `Country` in the semantic taxonomy of Wikipedia. As a result, we observe similar behavior of matchers across these groups. For instance, Figure 7 shows extensive unfair behavior of neural matchers on country music groups: `Honky Tonk`, `Cont. Country`, and `Country`. Following the same trend, non-neural matchers perform poorly on groups `Hip-hop/Rap` and `Rap` and `Rap & Hip-Hop`, suggesting these matchers are unfair to rap and hip-hop singers.

Non-neural Matchers: The non-neural matchers universally failed for the textual datasets (CAMERA and SHOES), with F-1 measures as low as zero in several cases. This underscores that these matchers are not fit for unstructured data. Still, in some settings these matchers were both inaccurate and unfair for different groups, as shown in Figures 12 and 15. Note that a matcher being fair in these cases simply means that it equally *failed* for all groups, not that it is a good choice. For example, LINREGMATCHER was fair for the SHOES dataset. However, looking at its overall performance, it turns out it did not correctly find *any* of the true matches for any of the groups, hence was equally bad for all groups.

On the other hand, non-neural matchers performed well for the structured datasets. Still, similar to the neural matchers, all of them showed unfairnesses in multiple cases. Further investigating these unfairnesses, we realized that by minimizing the overall error, these models put high weights on attributes that often indicate a match. In other words, overall, those attributes are good *proxies* for the ground-truth labels. However, when it comes to certain groups, they may not be as good proxies, causing the model to underperform for those groups. For example, let us consider SvmMatcher for the Dblp-Acm dataset, which was unfair for SIGMOD Rec. and VLDBJ. First, we realized that both these groups have frequently published reports or editorial articles with the same title but different years and authors. Being trained to perform for all groups, the SvmMatcher model assigned a high weight to the title, assuming that different articles should not have identical titles. Therefore, for examples like the one below, it matched them, although different authors wrote those in different years. This caused a higher ratio of false match detection (FP) compared to the other groups resulting in PPVP unfairness.

`(left entity)` **title:** `guest editorial;` **author:** `alon y. halevy;` **venue:** `VLDBJ;` **year:** `2002`
`(right entity)` **title:** `guest editorial ;` **author:** `vijay atluri , anupam joshi , yelena yesha;` **venue:** `VLDBJ;` **year:** `2003`

Besides, looking at Figure 9 the unfairness due to the high FP for SIGMOD Rec. and VLDBJ, caused pairwise unfairness for these two groups as well. Note that this issue is not necessarily limited to the non-neural matchers. For example, [16] also reports that an RNN-based matcher heavily relied on the “time” attribute when matching songs in the iTunes-Amazon dataset.

Lack of proper coverage [6, 43] in the data to sufficiently include different combinations is the reason the models do not get well-trained for those. For example, in the Dblp-Acm case, the training data did not include enough non-match cases with (almost) identical titles to reduce the correlation of the title with the ground-truth label.

6 LESSONS AND DISCUSSION

Some of the lessons learned in this study include:

(i) *Call for action to collect entity matching benchmarks on societal applications:* Perhaps the most challenging burden when auditing EM techniques from the fairness perspective is *lack of proper benchmark* datasets. Fairness is a societal issue and is meaningful when the EM task is on *individual records.* In real-world, EM is frequently used for tasks such as terrorist watch list screening [27], record linkage for data integration on medical data [4, 35], individual records deduplication [41], and many more. On the other hand, due to reasons such as privacy, such data are not publicly available. Although the EM community already has some benchmarks [25, 45], a thorough audit of existing and future EM techniques requires benchmark entity-matching data for societal applications.

In this paper, we took the first steps towards addressing this need by creating and publishing two semi-synthetic social datasets using publicly available real datasets. NoFlyCompas and FacultyMatch are generated for auditing the fairness of EM techniques when some groups are over-represented in data and when two demographic groups have different degrees of similarity in their names.

(ii) *Over-representation and name similarity in social data:* Group over-representation and higher (name) similarity degrees for specific groups are common in social data. Experiment results on our social datasets confirm the general unfairness of entity matchers under these conditions. Interestingly, under the group over-representation, we observed the superiority of non-neural matchers in terms of model performance and accuracy. Over-representation in general can increase the chance of finding similar non-matches for an entity, which can be falsely labeled as a match. Likewise, when names in one group are more similar, there is a higher chance of mistakenly labeling non-matching tuples from that group as a match.

(iii) *Unbiased and coverage-aware data:* Responsible training of EM techniques requires access to unbiased data with proper coverage of different groups and possible cases. In particular, equal ground-truth label (match, non-match) ratios for different groups are required, or it is not possible to satisfy statistical parity while maintaining high model performance which is fair from other definitions' perspectives. Insufficient coverage of different groups can bias the models in favor of some of the groups, making the model unfair.

(iv) *Proper fairness measures for entity-matching:* Different fairness definitions are valuable for different settings. Still, due to its pairwise matching nature, class imbalance, with most of the records being non-match, is a distinguishing property of EM. In this setting, *positive predictive value parity* and *true positive rate parity* aka equal opportunity is more capable of revealing the matchers' unfairnesses. Finally, some of the unfairnesses of a matcher, such as AP, could be explained using other measures, such as TPRP.

(v) *Proper Matching techniques for different settings:* Different matching techniques perform differently for different dataset types. At a high level, non-neural matchers fail for textual datasets while performing well for structured data. Lack of proper coverage in training data can bias these models to significantly rely on attributes (such as name, salary, etc.) that are highly correlated with the ground-truth label but may bias their performance for the minority groups. Neural matchers, on the other hand, generally perform well for different dataset types. Still, (a) using pre-trained language models and embeddings, (b) relying less on the structure of data caused these matchers to be unfair for different settings.

(vi) *Ensemble Learning for Fair Entity Matching:* We observed that, in a fixed dataset, some groups needed matchers with more complex decision boundaries, while others required matchers with simpler decision boundaries. As a result, adapting either of the neural/non-neural matchers would show unfairness for some groups. This observation underscores the need for techniques such as *ensemble learning* to consider a range of matchers with different properties to assure similar performances across different groups. Fortunately, the importance of ensemble-learning based approaches for EM [23, 46] and data problems in general [31] been recognized.

ACKNOWLEDGEMENT

The authors would like to thank Dongxiang Zhang and Zepeng Li for providing the code for Mcan matcher.

REFERENCES

[1] [n.d.]. u.s. census bureau quickfacts: united states. https://www.census.gov/quickfacts/fact/table/US/PST045221

[2] 2015. COMPAS Recidivism Risk Score Data and Analysis. www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis.

[3] [visited: 2023]. CSRankings GitHub Repository. https://github.com/emeryberger/CSRankings.

[4] Ernest Donald Acheson et al. 1967. Medical record linkage. *Medical record linkage.* (1967).

[5] Abolfazl Asudeh and H. V. Jagadish. 2020. Fairly evaluating and scoring items in a data set. *PVLDB* 13, 12 (2020), 3445–3448.

[6] Abolfazl Asudeh, Zhongjun Jin, and HV Jagadish. 2019. Assessing and remedying coverage for a given dataset. In *ICDE*. IEEE, 554–565.

[7] Nils Barlaug and Jon Atle Gulla. 2021. Neural networks for entity matching: A survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 3 (2021), 1–37.

[8] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. Fairness and machine learning: Limitations and opportunities. fairmlbook.org.

[9] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. 2018. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943* (2018).

[10] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606* (2016).

[11] Runjin Chen, Yanyan Shen, and Dongxiang Zhang. 2021. GNEM: a generic one-to-set neural entity matching framework. In *Proceedings of the Web Conference 2021*. 1686–1694.

[12] Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5, 2 (2017), 153–163.

[13] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. 2020. An overview of end-to-end entity resolution for big data. *ACM Computing Surveys (CSUR)* 53, 6 (2020), 1–42.

[14] Equal Employment Opportunity Commission. 1979. The U.S. Uniform guidelines on employee selection procedures.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[16] Vincenzo Di Cicco, Donatella Firmani, Nick Koudas, Paolo Merialdo, and Divesh Srivastava. 2019. Interpreting deep learning models for entity resolution: an experience report using LIME. In *Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*. 1–4.

[17] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[18] Vasilis Efthymiou, Kostas Stefanidis, Evaggelia Pitoura, and Vassilis Christophides. 2021. FairER: Entity Resolution With Fairness Constraints. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3004–3008.

[19] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 259–268.

[20] Cheng Fu, Xianpei Han, Jiaming He, and Le Sun. 2021. Hierarchical matching network for heterogeneous entity resolution. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 3665–3671.

[21] Forest Gregg and Derek Eder. 2022. Dedupe. https://github.com/dedupeio/dedupe.

[22] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016).

[23] Anna Jurek, Jun Hong, Yuan Chi, and Weiru Liu. 2017. A novel ensemble learning approach to unsupervised record linkage. *Information Systems* 71 (2017), 40–54.

[24] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2016. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807* (2016).

[25] Pradap Venkatramanan Konda. 2018. *Magellan: Toward building entity matching management systems.* The University of Wisconsin-Madison.

[26] Hanna Köpcke and Erhard Rahm. 2010. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering* 69, 2 (2010), 197–210.

[27] William J Krouse and Bart Elias. 2009. Terrorist watchlist checks and air passenger prescreening. LIBRARY OF CONGRESS WASHINGTON DC CONGRESSIONAL RESEARCH SERVICE.

[28] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. *Advances in neural information processing systems* 30 (2017).

[29] Bo-Han Li, Yi Liu, An-Man Zhang, Wen-Huan Wang, and Shuo Wan. 2020. A survey on blocking technology of entity resolution. *Journal of Computer Science and Technology* 35 (2020), 769–793.

[30] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *arXiv preprint arXiv:2004.00584* (2020).

[31] Ling Liu. 2022. Ensemble Learning Methods for Dirty Data. In *CIKM, Keynote*.

[32] Christina Makri, Alexandros Karakasidis, and Evaggelia Pitoura. 2022. Towards a more Accurate and Fair SVM-based Record Linkage. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 4691–4699.

[33] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[34] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*. 19–34.

[35] Fatemeh Nargesian, Abolfazl Asudeh, and HV Jagadish. 2022. Responsible Data Integration: Next-generation Challenges. In *Proceedings of the 2022 International Conference on Management of Data*. 2458–2464.

[36] Soudeh Nilforoushan, Qianfan Wu, and Mostafa Milani. 2022. Entity Matching with AUC-Based Fairness. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 5068–5075.

[37] George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, and Themis Palpanas. 2020. Blocking and filtering techniques for entity resolution: A survey. *ACM Computing Surveys (CSUR)* 53, 2 (2020), 1–42.

[38] George Papadakis, Jonathan Svirsky, Avigdor Gal, and Themis Palpanas. 2016. Comparative analysis of approximate blocking techniques for entity resolution. *Proceedings of the VLDB Endowment* 9, 9 (2016), 684–695.

[39] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[40] Anna Primpeli, Ralph Peeters, and Christian Bizer. 2019. The WDC training dataset and gold standard for large-scale product matching. In *Companion Proceedings of The 2019 World Wide Web Conference*. 381–386.

[41] Mark Scanlon. 2016. Battling the digital forensic backlog through data deduplication. In *2016 sixth international conference on innovative computing technology (INTECH)*. IEEE, 10–14.

[42] Nima Shahbazi, Nikola Danevski, Fatemeh Nargesian, Abolfazl Asudeh, and Divesh Srivastava. 2023. Through the Fairness Lens: Experimental Analysis and Evaluation of Entity Matching. https://github.com/DataIntelligenceCrew/FairEM/blob/main/techrep.pdf.

[43] Nima Shahbazi, Yin Lin, Abolfazl Asudeh, and HV Jagadish. 2022. A Survey on Techniques for Identifying and Resolving Representation Bias in Data. *arXiv preprint arXiv:2203.11852* (2022).

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[45] Jin Wang, Yuliang Li, and Wataru Hirota. 2021. Machamp: A Generalized Entity Matching Benchmark. In *CIKM*. ACM, 4633–4642.

[46] Liu Yi, Diao Xing-Chun, Cao Jian-Jun, Zhou Xing, and Shang Yu-Ling. 2017. A method for entity resolution in high dimensional data using ensemble classifiers. *Mathematical Problems in Engineering* 2017 (2017).

[47] Minghe Yu, Guoliang Li, Dong Deng, and Jianhua Feng. 2016. String similarity search and join: a survey. *Frontiers of Computer Science* 10, 3 (2016), 399–417.

[48] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*. PMLR, 962–970.

[49] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *International conference on machine learning*. PMLR, 325–333.

[50] Dongxiang Zhang, Yuyang Nie, Sai Wu, Yanyan Shen, and Kian-Lee Tan. 2020. Multi-context attention for entity matching. In *Proceedings of The Web Conference 2020*. 2634–2640.

[51] Hantian Zhang, Nima Shahbazi, Xu Chu, and Abolfazl Asudeh. 2021. FairRover: explorative model building for fair and responsible machine learning. In *Proceedings of the Fifth Workshop on Data Management for End-To-End Machine Learning*. 1–10.

APPENDIX

A EVALUATING THE CORRECTNESS OF ENTITY MATCHERS

The correctness of a matcher measures how well its matching predictions match the ground truth. Given a test dataset with correspondences of $t = (e_i, e_j, h, y)$, where h is a binary variable indicating the result of EM (*match* or *non-match*) for entities with encodings e_i and e_j, and y is a binary variable indicating the ground-truth for matching, we profile predictions of h using the numbers of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), respectively. Unlike a classification task, in the confusion matrix of a matching task, the result is counted both for the group(s) of e_i and the group(s) of e_j.

Example 2: Consider a test dataset, shown in Table 13, for a matcher $\mathcal{M}$, where columns id_1 and id_2 contain entity encodings, column h is the output decision of $\mathcal{M}$, and column y is the ground-truth. Comparing columns h and y, we add and populate column *Result* for each entity pair. Consider the simple case of having two groups $\mathcal{G} = \{g_1, g_2\}$. Suppose we would like to evaluate single fairness for g_1 and g_2. We describe how the confusion matrices of these groups are created. Consider the first row in Table 13a, which happens to be an FP. Since e_1 and e_2 both belong to subgroup g_1, the value 2 will be added to the count of FPs in the confusion matrix of g_1. However, in the second row which happens to be a TN, e_3 belongs to g_2 while e_4 belongs to g_1. Thereby, we will add one to both TN values of the confusion matrix corresponding to subgroup g_1 and g_2. We repeat the same procedure for rows three and four. The completed confusion matrices are shown in Figures 13b and 13c. □

id_1	id_2	group(id_1)	group(id_2)	h	y	Result
e_1	e_2	g_1	g_1	'M'	'N'	FP
e_3	e_4	g_2	g_1	'N'	'N'	TN
e_1	e_4	g_1	g_1	'M'	'M'	TP
e_2	e_3	g_1	g_2	'N'	'M'	FN

(a)

Predicted	Actual: y='M'	Actual: y='N'
h='M'	TP=2	FP=2
h='N'	FN=1	TN=1

(b)

Predicted	Actual: y='M'	Actual: y='N'
h='M'	TP=0	FP=0
h='N'	FN=1	TN=1

(c)

Figure 13: (a) Matching Results (b) Confusion Matrix of g_1 (c) Confusion Matrix of g_2.

We measure correctness for single and pairwise fairness through well-studied metrics in literature, including precision, recall, and F-1 score.

B GROUP ENCODING

To unify all attribute-value types, we summarize sub-groups in an encoding and use this encoding to represent individual entities and entity pairs. Given a set of sensitive attributes $\mathcal{A} = \{A_1, \ldots, A_n\}$ and value domains $dom(A_i)$ for attributes A_i, $\mathcal{G} = \{g_1, \ldots, g_m\}$ denotes the set of all level-1 groups, i.e. $\mathcal{G} = \bigcup_{A_i \in \mathcal{A}} dom(A_i)$. We represent a subgroup s of level k (k-combination) consisting of groups $s = \{g_1, \ldots, g_k\}$, with a binary encoding $s = \langle a_1, \ldots, a_m \rangle$, where $m = |dom(A_1)| \times \ldots |dom(A_i)|$) and a_i is one if $g_i \in s$ and is zero otherwise. Note that for a k-combination subgroup, exactly k entries of s get the value one. We represent an entity e associated with groups $G \subseteq \mathcal{G}$ with a binary encoding $\langle b_1, \ldots, b_m \rangle$, where b_i is one if $g_i \in G$ and is zero otherwise. An entity e with groups G belongs to subgroup s if $s \subseteq G$. Given an entity encoding $e = \langle b_1, \ldots, b_m \rangle$ and a subgroup encoding $s = \langle a_1, \ldots, a_m \rangle$, we say e belongs to subgroup s if $s\ AND\ e\ ==\ s$, i.e. the entity belongs to every group that define the subgroup s. The encoding of an entity pair e_i, e_j is the concatenation of the encodings of e_i and e_j.

Example 3: Consider attributes `genre` and `gender` of Figure 1. Assuming a lexicographical order on all groups, the encoding of entity e with associated groups G={`Female`, `Pop`, `Rock`} is $\langle 1, 0, 0, 1, 1 \rangle$. The encoding of a level-2 subgroup s={`Female`, `Pop`} is $\langle 1, 0, 0, 1, 0 \rangle$. □

C ADDITIONAL EXPERIMENTS

C.1 Correctness

Accurate	Fair	Evidence
×	×	RfMatcher: Cameras: {TPRP,PPVP} BooleanRuleMatcher: iTunes-Amazon: {AP,SP,PPVP} Gnem: iTunes-Amazon: {AP,PPVP,...}
×	✓	LinRegMatcher: Shoes Gnem: Dblp-Acm BooleanRuleMatcher: Cricket
✓	×	HierMatcher: iTunes-Amazon: {AP,PPVP,...} SvmMatcher: Dblp-Acm: PPVP Mcan: Cameras: TPRP Ditto: Dblp-Scholar: {AP,TPRP,...} DeepMatcher: Camera: {PPVP,TPRP}
✓	✓	Mcan: Dblp-Acm Ditto: Cricket NbMatcher: Dblp-Scholar

Figure 14: Fairness and accuracy synergies

Neural matchers are more accurate than non-neural matchers on textual and dirty data. The correctness results of the textual datasets: Shoes and Camera, can be found in Table 7. Non-neural matchers extensively suffer in F-1 score, compared to neural matchers that have higher ranges of F-1 score. Modern neural matchers such as Ditto and DeepMatcher draw on external knowledge by incorporating language models, which helps a matcher to learn the relevance of entities despite the lack of structure and syntactic similarity in text entities. This result is consistent with what is reported by the state-of-art matchers.

Non-neural matchers are more accurate than neural matchers on structured data. Considering the structured datasets: iTunes-Amazon and Dblp-Acm, although all matchers, with the exception of BooleanRuleMatcher, perform quite well, the non-neural matchers have slightly higher F-1 scores overall.

	Compas		CSRankings		iTunes-Amazon		Dblp-Acm		Dblp-Scholar		Cricket		Shoes		Camera	
Matcher	**Acc**	**F-1**	**Acc**	**F-1**	**Acc**	**F-1**	**Acc**	**F-1**	**Acc**	**F-1**	**Acc**	**F-1**	**Acc**	**F-1**	**Acc**	**F-1**
BooleanRuleMatcher	0.99	0.14	0.99	0.37	0.29	0.41	0.41	0.38	0.38	0.38	0.03	0.0	0.82	0.28	0.81	0.4
DeepMatcher	0.99	0.84	0.99	0.70	0.94	0.88	0.99	0.98	0.97	0.92	0.87	0.92	0.96	0.82	0.93	0.81
Ditto	0.99	0.79	0.99	0.78	0.91	0.84	0.99	0.98	0.95	0.87	0.96	0.98	0.95	0.78	0.91	0.76
Gnem	0.99	0.86	0.99	0.85	0.64	0.31	0.70	0.18	0.83	0.58	0.96	0.98	0.96	0.80	0.97	0.91
HierMatcher	0.99	0.77	0.99	0.72	0.93	0.87	0.95	0.88	0.96	0.9	0.81	0.89	0.96	0.81	0.94	0.83
Mcan	0.99	0.69	0.99	0.68	0.97	0.94	0.99	0.99	0.97	0.92	0.95	0.97	0.95	0.73	0.93	0.78
SvmMatcher	0.99	0.99	0.99	0.94	0.92	0.84	0.96	0.90	0.94	0.86	0.96	0.98	0.89	0.0	0.84	0.27
RfMatcher	1.00	1.00	0.99	0.95	0.94	0.89	0.99	0.97	0.98	0.94	0.96	0.98	0.88	0.29	0.82	0.38
NbMatcher	0.99	0.98	0.96	0.11	0.88	0.78	0.98	0.97	0.99	0.97	0.96	931	0.86	0.26	0.82	0.38
LogRegMatcher	1.00	1.00	0.99	0.93	0.91	0.83	0.99	0.97	0.99	0.97	0.96	0.98	0.89	0.04	0.84	0.31
LinRegMatcher	0.99	0.95	0.99	0.38	0.97	0.94	0.97	0.93	0.95	0.88	0.96	0.97	0.89	0.0	0.84	0.30
Dedupe	-	-	-	-	0.94	0.89	0.95	0.85	0.95	0.87	0.96	0.98	-	-	-	-
DtMatcher	1.00	1.00	0.99	0.93	0.94	0.89	0.99	0.97	0.98	0.94	0.93	0.96	0.85	0.30	0.84	0.31

Table 7: Overall performance of matchers across different datasets[5](Acc: Accuracy)

The main job of a matcher is to find matching entities. A failure in doing so results in a low number of TPs, which reflects in not only a low F-1 score but also unfairness with respect to TPRP and PPVP measures for many groups across the board, as we observe in Figure 17 and 12. Recall that these measures verify how well a matcher performs in identifying true positives. Similar observations can be made in both datasets regarding the unfairness of the BooleanRuleMatcher. One interesting observation in Gnem, which is the only neural matcher with a low F-1 score for the Dblp-Acm dataset, is that the high number of FNs of Gnem results in the pairwise unfairness for $g_i|g_i$ pairs (e.g., `SIGMOD|SIGMOD` and `ACM TODS|ACM TODS`). Gnem does not demonstrate NPVP unfairness on $g_i|g_j$ pairs, particularly in the Dblp-Acm dataset, because often two entities with $g_i|g_j$ are not a match (e.g. it is rare that a `ACM TODS` publication is matched with a publication `SIGMOD`).

The flip side of correctness and fairness also exists in EM. For example, the BooleanRuleMatcher and Gnem have low accuracy and F-1 score on Dblp-Acm, while no unfairness issue is reported for these matchers, in Figure 5. This can be explained by the low accuracy of these matchers for all groups across the board which makes the disparity a low value. In Figure 14, we present a selective overview of the unfairness and accuracy of matchers across all datasets. The general message is that similar to accuracy, unfairness is dataset and measure dependent. First, no matcher is unfair across all datasets. For example, Gnem is unfair for iTunes-Amazon and Dblp-Acm but is not unfair for the Cricket dataset. No matcher is unfair across all measures. For example, on Dblp-Scholar dataset, HierMatcher is only unfair with respect to PPVP and FPRP and not other measures. We will have a more detailed discussion on the behavior of macther w.r.t measures in § 5.3.2.

C.2 Cameras Dataset: Pairwise Fairness

C.3 iTunes-Amazon Dataset: Pairwise Fairness

In Figure 16, extensive unfair behavior of neural matchers happens in the pairwise matching of `Dance & Electronic` with `Music` and `Dance`.

C.4 Shoes Dataset: Single and Pairwise Fairness

Model	Marker
DeepMatcher	●
Ditto	●
GNEM	●
HierMatcher	●
MCAN	●
SvmMatcher	★
RfMatcher	★
NbMatcher	★
LogRegMatcher	★
LinRegMatcher	★
DtMatcher	★
Dedupe	★
BooleanRuleMatcher	★

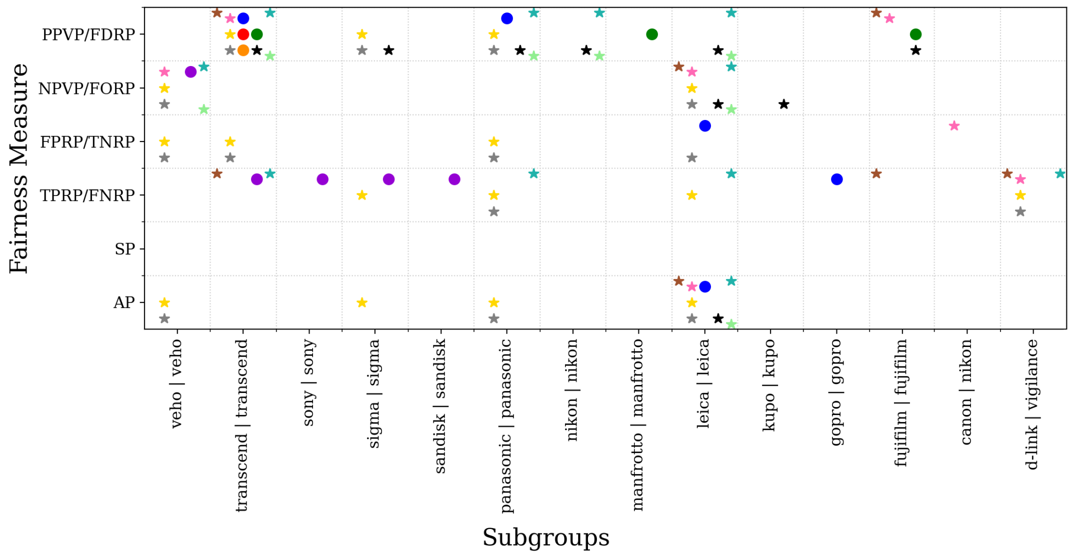

Figure 15: **CAMERAS: Pairwise Fairness.**

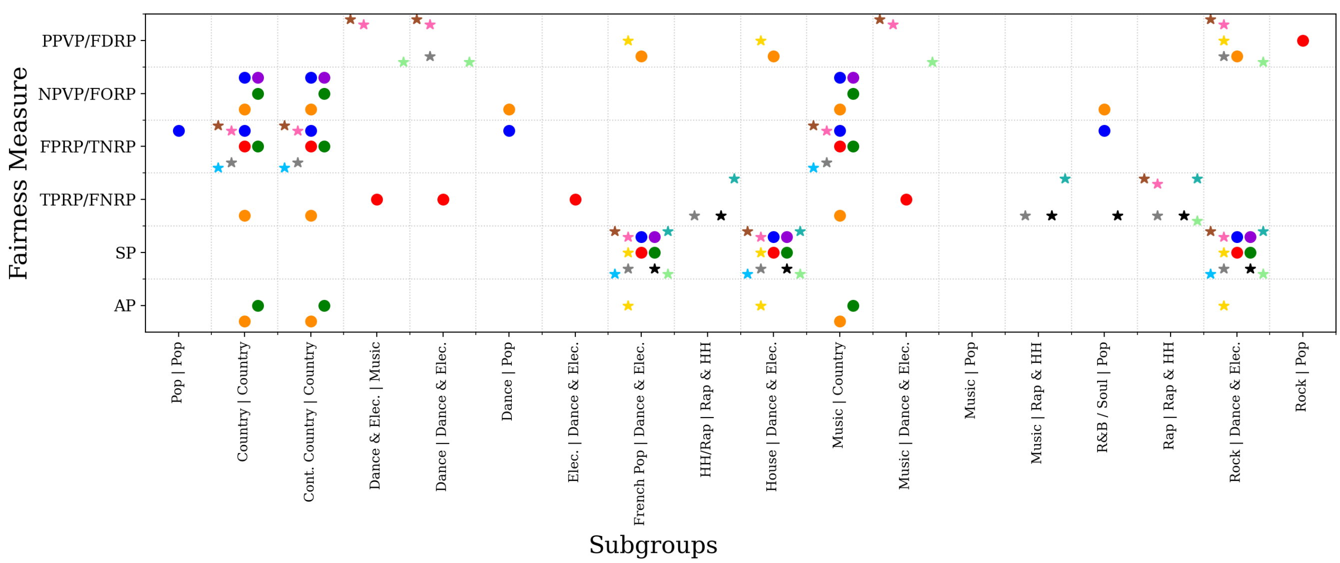

Figure 16: **iTUNES-AMAZON: Pairwise Fairness. (HH: Hip-Hop, Elec.: Electronic, Cont.: Contemporary)**

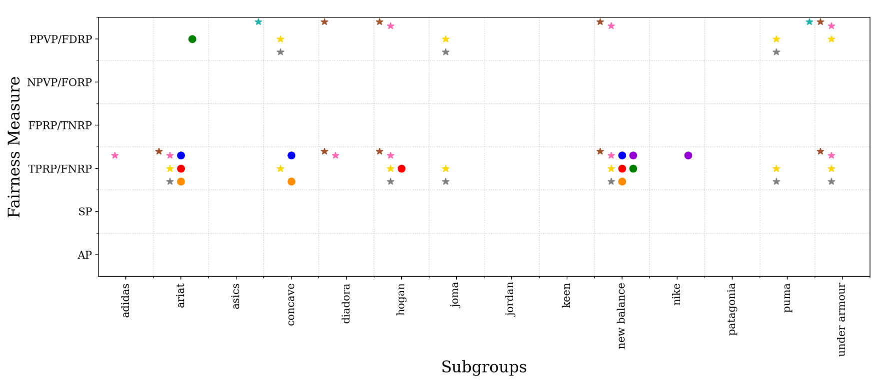

Figure 17: **SHOES: Single Fairness**

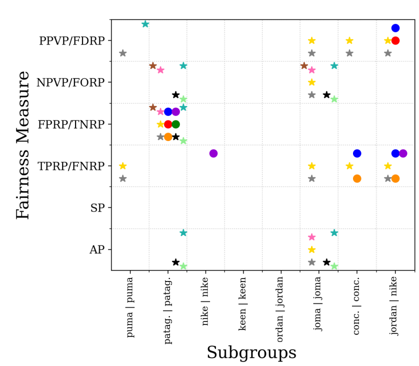

Figure 18: **SHOES: Pairwise Fairness**