# Evaluating Bias in Entity Matching data preparation

Mohmmad Hossein Moslemi
*The University of Western Ontario*
*London, Ontario, Canada*
mohammad.moslemi@uwo.ca

Harini Balamurugan
*The University of Western Ontario*
*London, Ontario, Canada*
hbalamur@uwo.ca

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

***Abstract*—Entity Matching (EM) is crucial for identifying equivalent data entities across different sources, a task that becomes increasingly challenging with the growth and heterogeneity of data. Blocking techniques, which reduce the computational complexity of EM, play a vital role in making this process scalable. Despite advancements in blocking methods, the issue of fairness—where blocking may inadvertently favor certain demographic groups—has been largely overlooked. This study extends traditional blocking metrics to incorporate fairness, providing a framework for assessing bias in blocking techniques. Through experimental analysis, we evaluate the effectiveness and fairness of various blocking methods, offering insights into their potential biases. Our findings highlight the importance of considering fairness in EM, particularly in the blocking phase, to ensure equitable outcomes in data integration tasks.**

## I. Introduction

Entity Matching (EM) is the process of determining whether two or more data entities from the same or different data sources refer to the same real-world object. As data sources grow and become increasingly heterogeneous, the challenge of accurately and efficiently matching entities has intensified. Also known as entity linkage or record matching, EM is a fundamental task in data integration with broad applications across various industries. It has attracted significant attention from researchers [1], [2], [3], [4], [5], [6], [7]. In commercial domains, EM is commonly used for tasks like matching customer or product records across different databases. In critical public sector areas such as healthcare and security, accurate and efficient EM is essential and can have significant implications [8]. For instance, in the healthcare industry, patient records from multiple hospitals or clinics must be accurately matched to ensure comprehensive treatment and avoid duplication. Variations in data entry, such as typographical errors or different naming conventions, can result in multiple records for the same patient. In these scenarios, EM is crucial for consolidating records and providing a complete view of a patient's medical history.

EM systems typically include a matching component that takes two entities and determines whether they are equivalent, labeling them as a "match" or "non-match." A significant challenge in the EM process is its computational complexity, which often scales quadratically because each entity must be compared against all others, resulting in an $O(n^2)$ problem [9], [10]. This complexity can become prohibitive for large datasets containing millions of records, as the number of potential comparisons increases exponentially. To address this challenge, blocking methods are employed as a preliminary step to reduce the number of record comparisons [11], [12], [13]. Consequently, EM systems are generally divided into two phases: the first phase involves blocking, and the second phase involves the matcher producing the final labels.

Blocking reduces the number of comparisons by grouping similar entities into distinct or overlapping blocks, ensuring that comparisons are only made within these smaller, more manageable groups. This step is critical as it significantly reduces the computational load, making the EM process more scalable. By partitioning the dataset into blocks where records are more likely to match, blocking not only enhances computational efficiency but also addresses the inherent quadratic complexity of EM.

Over the years, a wide range of blocking techniques has been developed, from simple heuristic-based methods to advanced approaches involving deep neural networks (for surveys, see [14], [15], [9]). These blocking methods can be categorized in various ways, each providing a different perspective on how these techniques function and their applications. One way is by distinguishing between learning-based and non-learning-based algorithms. Rule-based methods, a type of non-learning-based approach, rely on expert knowledge or simple heuristics to define the blocking criteria. In contrast, learning-based methods require training data to learn how to block the data using machine learning techniques.

Another categorization is based on schema awareness. Schema-aware methods perform blocking by focusing on the most important attributes of the data, while schema-agnostic methods treat the entire entity as a single attribute, utilizing all available information. A third categorization concerns redundancy awareness, dividing methods into redundancy-free, redundancy-positive, and redundancy-neutral subcategories, which differ in how they handle the assignment of entities to blocks and the overlap between blocks. These various categorizations highlight the diversity of blocking techniques available, each suited to different scenarios depending on the data and the desired balance between computational efficiency and match accuracy [9].

Traditional blocking methods such as Standard Blocking and Sorted Neighborhood have fundamentally shaped the field [16], [17]. Standard Blocking categorizes records according to a Blocking Key, such as a phone number or surname initials, to conduct intensive comparisons within these blocks. However, this method risks inefficiencies when block sizes are large. In

contrast, Sorted Neighborhood enhances efficiency by sorting records according to a key and employing a sliding window for comparisons, though it may overlook matches when key values exceed the window's boundaries.

To address some of the limitations of traditional methods, advanced approaches like Canopy Clustering [18] have been introduced. This technique uses a less costly, coarse similarity measure to initially group records, which are then subjected to more precise and computationally demanding comparisons within each canopy. Although designed to minimize the total number of comparisons, this method may occasionally result in the erroneous grouping of distinct entities [19].

Recent advancements in blocking techniques, such as BSL (Blocking Scheme Learner) [20] and BGP (Blocking based on Genetic Programming) [19], have further enhanced efficiency and accuracy. These methods leverage machine learning to refine blocking schemes, focusing on attribute selection and comparison methods to generate candidate matches efficiently. CBLOCK offers an automated approach to canopy formation within a map-reduce framework, tailored specifically for large-scale de-duplication tasks involving diverse datasets, thus optimizing the trade-off between recall and computational efficiency [21]. Additionally, Token-Based Blocking effectively addresses challenges in heterogeneous datasets by comparing records based on shared tokens, providing a versatile solution for integrating diverse data sources.

Moreover, deep learning has revolutionized the blocking phase in EM, shifting from traditional heuristic methods to more adaptive and automated approaches. Frameworks like AutoBlock and DeepER exploit deep learning for representation learning and nearest neighbor search, demonstrating significant effectiveness across varied, large-scale datasets [22], [23]. DeepBlock, which merges syntactic and semantic similarities through deep learning, further enhances blocking quality by accurately grouping similar records, even in noisy or heterogeneous datasets [24].

Researchers have defined various metrics to measure the quality of blocking, but three are most commonly used and considered the most comprehensive in assessing the effectiveness of blocking methods: Reduction Ratio (RR), Pair Quality (PQ), and Pair Completeness (PC) [25], [26], [15]. The RR measures the extent to which a blocking system reduces the total number of comparisons. PQ denotes the percentage of candidate pairs that are true matches after blocking. Finally, PC indicates the percentage of true matches present in the candidate set after blocking. We will formally define these metrics in Section II.

In recent years, fairness in ML has garnered significant attention [27], [28], [29] due to its critical impact on real-life applications. Fairness is particularly important in the context of EM because both EM and blocking systems can produce biased results, often exhibiting higher accuracy for one demographic group over another. Despite the significant implications of EM on real-life decisions, research on the fairness of EM remains limited, with only a few studies exploring this issue [30], [31], [32]. Even fewer studies have addressed the fairness of blocking methods, leaving this area largely unexplored. To the best of our knowledge, only one study has investigated the fairness of blocking, which merely touched on the topic by defining a fairness metric for blocking based on the representation ratio, similar to RR, and proposing simple algorithms to address bias in blocking [33].

Traditional fairness metrics, such as Equalized Odds, Equal Opportunity, and Statistical Parity [28], [29], are typically defined for ML models and are based on accuracy metrics. These metrics cannot be directly applied to blocking methods, as blocking is a pre-processing step and does not produce accuracy metrics like ML models. One of the contributions of our study is to extend the existing blocking metrics to incorporate fairness and to evaluate how effectively these metrics can measure bias in different blocking techniques. For simplicity, this work focuses solely on binary sensitive attributes.

In the remainder of this paper, Section II provides a formal definition of EM systems, detailing their components, specifically the blocking and matching stages. This section also introduces the metrics used to evaluate blocking. In Section III, we extend these blocking metrics to fairness measures, proposing them as a sufficient method for assessing bias in blocking. Section IV presents our experimental results, highlighting the effectiveness of various blocking methods and the biases they may introduce. Section V summarizes related work, and finally, Section VI offers our conclusions. All the implementations are available at https://github.com/mhmoslemi2338/pre-EM-bias.

## II. Background

We start from a relation schema $\mathcal{S}$ consisting of a set of attributes $A_1, ..., A_m$ with domains $\text{DOM}(A_i), i \in [1, m]$. A record (tuple) $t$ with schema $\mathcal{S}$ is a member of $\text{DOM}(A_1) \times ... \times \text{DOM}(A_m)$, the set of all possible records, which we denote by $\mathcal{D}$. We use $t[A_i]$ to refer to the value of attribute $A_i$ in record $t$. A relation $D \subseteq \mathcal{D}$ is a set of records.

### A. Record Matching

Given two relations $D_1, D_2$, the problem of *record matching* is to find a subset $M$ of $D_1 \times D_2$ that consists of record pairs referring to the same real-world entities. We refer to such record pairs as *equivalent pairs*.

A record matcher (matcher in short) for records of schema $\mathcal{S}$ is a binary classifier $f : \mathcal{D} \times \mathcal{D} \mapsto \{0, 1\}$; that labels record pairs 1, called they match, or 0, saying they don't match. Given relations $D_1, D_2$ with schema $\mathcal{S}$ and equivalent record pairs $M \subseteq D_1 \times D_2$, the goal of $f$ is to find the equivalent records in $M$, i.e., label the record pairs in $M$ as a match and the non-equivalent records as non-match, where the accuracy of $f$, e.g., true/false positive/negative rates and F1 score, are defined based on whether the equivalent record pairs are correctly labeled.

A typical EM system generally has two major components: blocking and matching [1].

**Blocking** The set of record pairs for relations $D_1, D_2$ in a matching setting, which we denote by $P = D_1 \times D_2$, grows quadratically in size w.r.t the size of $D_1$ and $D_2$. This makes it costly to run expensive matching methods for all possible pairs. The problem of blocking is to find a candidate set $C \subseteq P$ which is much smaller than $|P|$, while it still includes all equivalent pairs; $M \subseteq C$. Blocking allows saving unnecessary checking some of the non-equivalent pairs while searching the equivalent pairs.

**Definition II.1** (Blocking)**.** *Given two datasets, $D_1$ and $D_2$, the set of all possible pairs is denoted by $P = D_1 \times D_2$, an. The goal of blocking is to generate a candidate set $C$ s.t. $|C| \ll |P|$ and $M \subseteq C$ in a reasonable time.*

**Blocking Metrics** How effective a blocking method is measured using three main quality measures defined as follows:

$$\text{RR} = 1 - \frac{|C|}{|P|} \qquad \text{PC} = \frac{|C \cap M|}{|M|} \qquad \text{PQ} = \frac{|C \cap M|}{|C|} \tag{1}$$

*Reduction ratio* (RR) is the ratio of the reduction in the number of comparisons after blocking to the total number of possible comparisons. A higher RR value signifies a greater reduction in the number of candidate record pairs. This measure does not consider the quality of the generated candidate record pairs. *Pairs completeness* PC represents the ratio of equivalent pairs retained after blocking to the total number of equivalent pairs. This measure evaluates how effectively $C$ preserves equivalences, corresponding to recall in information retrieval [34]. PC ranges from 0 to 1, where a value of 1 indicates that the blocker retains all true matches. *Pairs quality* (PQ) denotes the fraction of equivalent pairs produced by $C$ relative to the total number of pairs. A higher PQ indicates that $C$ is efficient, primarily generating true matches. In contrast, a lower PQ suggests that many non-matching pairs are included. PQ is equivalent to precision in information retrieval [9]. PQ ranges from 0 to 1, with higher values indicating that the blocker is more effective at eliminating non-matches.

Mosatafa: Add harmonic mean and cite... justify, also talk about time! Mohammad-Hossein: I added the below

In addition to evaluating the effectiveness of blocking methods, their efficiency is equally important. Specifically, the runtime of a blocking method plays a crucial role in determining the optimal approach. If runtime is overlooked, one could end up using a matching technique that, while achieving high recall rate (RR), precision (PC), and pair quality (PQ), might take days to execute.

Unlike the consensus found in accuracy measures for machine learning, there is little agreement in the literature on the evaluation measures for blocking. Some studies only consider PC and RR as their primary metrics [22], [35], [36], [37], [38], [39], [11], [12], [10]. Others focus solely on PC and runtime [40], [41], [42], [43]. There is also research that employs the harmonic mean of two measures, defined as $F_{a,b} = \frac{2 \times a \times b}{a+b}$. However, even in these cases, there is no consensus on which measures to combine using the harmonic mean. For example, [44], [45], [46], [47], [48] used the harmonic mean of PC and RR, whereas [15] utilized the harmonic mean of PC and PQ. Some studies only considered PC and PQ [24], [49].

One study attempted to combine runtime and harmonic mean into a single blocking measure, but did so incorrectly. They defined runtime as the time needed for both blocking and matching, then calculated the harmonic mean of precision and recall at the end of the entity matching (EM) pipeline. Their evaluation metric was $\sqrt{time^2 + (1 - F_{precision,recall})^2}$, which they tried to minimize. However, this approach is flawed because runtime and harmonic mean are fundamentally different in nature, making it inappropriate to combine them in this manner.

There is currently no clear explanation of which blocking measures are most important or how they relate to each other. Some studies [14], [42], [37], [47] have briefly touched on this topic, suggesting that PQ is typically low in blocking methods because if both PC and PQ were high, further matching would not be necessary. Achieving high PQ is typically the role of the matching phase. However, this explanation is insufficient.

In this study, we argue that focusing solely on PC, RR, and runtime is sufficient for evaluating the effectiveness of different blocking methods, with PQ being inherently related to PC and RR. Blocking is generally applied to datasets with a large number of pairs (on the order of millions), where the number of true matches is typically much smaller (on the order of thousands). For a well-performing blocking method, RR should exceed 90% (often around 99%), meaning that a PC of 1 could result in a PQ of only 10%. Even a small change in RR (on the order of 0.01%) can significantly impact the candidate set size, making PQ highly sensitive. This sensitivity is one reason why focusing only on PC, RR, and runtime is sufficient. Additionally, due to the usual sizes of candidate pairs, pairs, and equivalent pairs, PQ correlates with a combination of RR and PC, allowing us to focus on just these two metrics.

Furthermore, there is a trade-off between PC and RR: maximizing one can lead to a decrease in the other. For this reason, similar to prior studies mentioned earlier, we also use the harmonic mean of PC and RR, defined as follows:

$$F_B = \frac{2 \times \text{PC} \times \text{RR}}{\text{PC} + \text{RR}} \tag{2}$$

We will offer a more detailed numerical analysis of the relationships between these metrics in section IV.

**Example II.2.** *Figure 1 shows a relation $D$ with ten records. We consider matching records in the same relation record pairs $P = D \times D$. The red dotted lines show the equivalent pairs; $M = \{(t_2, t_6), (t_6, t_7), (t_4, t_5), (t_8, t_9), (t_8, t_{10}), (t_9, t_{10})\}$. The solid black lines show blocking methods that specify three blocks that result in $3 + 6 + 3 = 12$ candidate pairs. The blocking result misses one equivalent pair, $(t_2, t_6)$, but reduces the number of pairs to check from 45 to 12, resulting in RR $= 1 - \frac{12}{45} \approx 0.73$, PC $= \frac{5}{6} \approx 0.83$, PQ $= \frac{5}{12} \approx 0.42$, and $F_B = \frac{2 \times \frac{5}{6} \times \frac{33}{45}}{\frac{5}{6} + \frac{33}{45}} = \frac{110}{141} \approx 0.78$.*

**Matching** The primary objective of the matching component in the EM system is to identify pairs and produce a label or score that reflects the likelihood of the pair being "equivalent." The aim is to achieve optimal matching accuracy. However, this study does not focus on the matching aspect [1].

**Definition II.3** (Matching)**.** *Given an arbitrary pair $p = (a, b)$, where $a \in D_1$ and $b \in D_2$, the goal of a matching system is to assign a score to each pair $p$ in a manner that maximizes a chosen accuracy measure, tailored to the specific application.*

## III. Measuring Bias in Blocking

Blocking methods can suffer from disparities; the quality of blocking might differ for the minority group, leading either to missing equivalent minority pairs, or being ineffective in reducing unnecessary matching of non-equivalent methods, which can happen if the size of the candidate set is not much smaller than all the minority pairs. A minority pair is defined as any pair of records $(t_1, t_2) \in D_1 \times D_2$ where at least one of the records belongs to a minority group based on a specified sensitive attribute (e.g., ethnicity, gender, etc.); a majority pair will be a pair with both majority records. Mosatafa: Add a few sentences to justify this...

Mohammad-Hossein: pls remove $\Delta$PQ as bias measure since we do not conisder PQ, and in previus section and experimetns section I explained why, by it would be nice also recall to the reader why we do not consider it here.

To define disparities in blocking, we use $P_g$, $C_g$, and $M_g$ with $g \in \{a, b\}$ to respectively refer to the set of pairs in group $g$, and the set of candidate pairs in group $g$, and the set of equivalent pairs in group $g$. Then we define the reduction ratio, pair completeness, and pair quality for the minority groups $g$ as follows:

$$\text{RR}_g = 1 - \frac{|C_g|}{|P_g|} \quad \text{PC}_g = \frac{|C_g \cap M|}{|M_g|} \quad \text{PQ}_g = \frac{|C_g \cap M|}{|C_g|} \tag{3}$$

Intuitively, the reduction ratio per group $g$ specifies the reduction in the number of comparisons for the record pairs in group $g$. Similarly, pair completeness and pair quality per group $g$ measures these measures between the pairs in the group $g$ only.

Mohammad-Hossein: i added the two paragraphs below:

The output from the blocking process will be fed into the matcher, with the ultimate goal of the EM pipeline being to maximize both precision and recall. If a true match is mistakenly removed during the blocking phase, it will be incorrectly labeled as a non-match, thus contributing to the false negatives. Therefore, the primary objective of blocking is to maximize the PC.

While it is acceptable for the blocker to introduce some additional candidates into the candidate set, slightly reducing the RR or PQ, this is not a significant issue as the matcher can handle the increased number of candidates. However, if a matching pair is incorrectly removed at the blocking stage, it results in the EM pipeline prematurely assigning it a non-match label. Consequently, it is crucial to maximize PC and minimize bias in PC, as this measure is more critical than the others.

Mosatafa: harmonic mean...

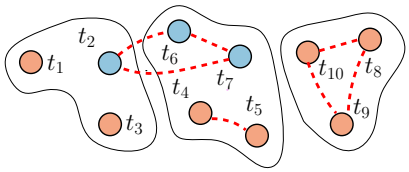



Fig. 1: Disparity in blocking: minority and majority records are highlighted in blue and red resp. and the equivalent pairs are linked by dotted lines. Solid lines show the blocks.

**Example III.1.** *Continuing with the example in Figure 1, three records are from the minority group, and seven are from the majority groups, as highlighted by different colors. These give five majority pairs (with both records from the majority group) and $3\times7+3 = 24$ minority pairs (with at least one record from the minority group). The following are the blocking quality measures per group: $RR_a = 1 - 7/24 \approx 0.71$ (there are seven minority pairs after blocking), $PC = 1/6 \approx 0.17$, $PQ = 1/24 \approx= 0.04$, $RR_b = 1 - 5/5 = 0$, $PC = 4/4 = 1$, and $PQ = 4/5 = 0.8$. These measures give reduction ratio, pair completeness, and pair quality disparities of $0.71$, $0.83$, and $0.76$, respectively.*

The disparities for blocking are defined as the differences between the blocking quality measures for the minority and the majority groups, e.g., $|\text{RR}_a - \text{RR}_b|$ is reduction ratio disparity and $|\text{PC}_a - \text{PC}_b|$ is pair completeness disparity. We use these disparity measures to evaluate the fairness of the existing blocking methods.

The introduction of these disparity metrics provides a crucial tool for evaluating the fairness of blocking techniques. By quantifying the differences in Reduction Ratio, Pair Completeness, and Pair Quality between demographic groups, it becomes possible to identify potential biases and inequalities in the blocking process. These insights are invaluable for refining existing blocking methods and developing new techniques that aim to minimize these disparities. For instance, if a significant Pair Completeness Disparity is detected, the blocking criteria can be adjusted to ensure that minority groups are not disproportionately disadvantaged. Similarly, balancing the Reduction Ratio across groups can help ensure that the computational benefits of blocking are equitably distributed.

The proposed method for quantifying biases in blocking techniques extends traditional performance measures and introduces a critical framework for evaluating and improving the fairness of these methods.

## IV. Evaluation and Analysis

### A. Experimental Setup

**Datasets** We utilize datasets from prominent EM benchmarks: Amazon-Google (`AMZ-GOO`), Walmart-Amazon (`WAL-AMZ`),

DBLP-Google Scholar (DBLP-GOO), DBLP-ACM (DBLP-ACM), Beer (BERR), Fodors-Zagat (FOD-ZAG), iTunes-Amazon (ITU-AMZ), and Febrl (FEBRL), as referenced in [1], [50]. These datasets are commonly adopted within the research community to evaluate EM system performance.

Consistent with prior studies [32], [51], [31], we categorize entities into minority and majority groups across various datasets. For instance, in AMZ-GOO, the presence of "Microsoft" in the "manufacturer" attribute signifies a minority group. In WAL-AMZ, entities classified as "printers" under the "category" attribute are considered a minority group. In DBLP-GOO, entities are deemed minority if the "venue" attribute includes "vldb j," while in DBLP-ACM, the inclusion of a female name in the "authors" attribute defines the minority. In FOD-ZAG, entities with the "Type" attribute exactly equal to "Asian" are classified as minority. For BERR, entities with "Beer Name" containing the phrase "red" are classified as minority, in ITU-AMZ, the minority group includes those where the "Genre" attribute contains the word "Dance," and in the FEBRL dataset, entities are considered minority if the "Given Name" attribute is female. Detailed statistical information about these datasets is provided in Table I.

| **Dataset** | $\|D_1\|$ | $\|D_2\|$ | $\|P\|$ | $\|M\|$ | Minority | | |
|---|---|---|---|---|---|---|---|
| | | | | | $\|D_1\|$ | $\|D_2\|$ | $\|M\|$ |
| WAL-AMZ | 2.6k | 22.0k | 56.4m | 962 | 96 | 172 | 88 |
| FEBRL | 6.5k | 6.5k | 42.1m | 8.5k | 2.5k | 2.5k | 3.7k |
| BERR | 4.3k | 3.0k | 13.0m | 68 | 1.3k | 932 | 29 |
| AMZ-GOO | 1.4k | 3.2k | 4.4m | 1.2k | 83 | 4 | 60 |
| FOD-ZAG | 533 | 331 | 176.4k | 111 | 72 | 3 | 10 |
| ITU-AMZ | 6.9k | 55.9k | 386.2m | 132 | 1.9k | 12.7k | 40 |
| DBLP-GOO | 2.6k | 64.3k | 168.1m | 5.3k | 191 | 389 | 403 |
| DBLP-ACM | 2.6k | 2.3k | 6.0m | 2.2k | 251 | 225 | 310 |

TABLE I: Datasets and their characteristics.

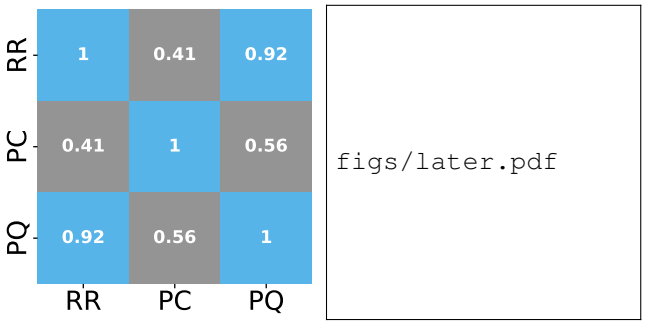
**Blocking Methods** This part outlines the blocking methods employed in our study, including both traditional and deep learning-based techniques.

i) *Standard Blocking (StandBlock)*: This hash-based method groups records using concatenated attribute values (blocking keys) to form redundancy-free blocks. However, it is sensitive to noise; any variation in the key may exclude matching records from the same block [52].

ii) *Q-Grams (Q-Gram) and Extended Q-Grams Blocking (Ext-Q-Gram)*: Q-grams blocking splits blocking keys into subsequences of $q$ characters, improving noise tolerance but potentially increasing block size and number. Extended Q-Grams further combines q-grams for more distinctive keys, reducing block size and enhancing efficiency [15], [53].

iii) *Suffix (Suffix) and Extended Suffix (Ext-Suffix) Arrays Blocking*: Suffix Arrays Blocking converts blocking keys into suffixes of a minimum length to form blocks, filtering out common suffixes to prevent oversized blocks. Extended Suffix Arrays consider all substrings longer than the minimum length, boosting noise tolerance [15], [53].

iv) *AutoEncoder (AUTO) and Cross-Tuple Training (CTT) Blocking*: These deep learning methods group similar records into blocks using embeddings. AUTO generates embeddings via an autoencoder to handle diverse and noisy data, while CTT uses a Siamese Summarizer to create embeddings from synthetic data, enhancing distinction between matching and non-matching tuples [1].




a) correlation matrix b) later

Fig. 2: Fig. 2a displays the correlation of various blocking measures.

## B. Results

**Blocking Methods Analysis** To the best of our knowledge, no study comprehensively examines the effectiveness of various blocking methods across a wide range of EM dataset benchmarks. One possible reason for this is that researchers often focus primarily on the final accuracy of the matcher. However, it is crucial to recognize that the output of the blocking process serves as the input for the matcher; therefore, a poorly performing blocking method can potentially result in a less accurate matcher. The evaluation of different blocking techniques across EM dataset benchmarks is fragmented across various studies, and not all studies report all relevant blocking metrics [45], [14], [10], [14], [15]. In this study, we evaluated different blocking techniques on EM benchmark datasets using three key metrics: RR, PC, and PQ. The results of our evaluation are presented in Table II

**Bias Analysis of Blocking Methods** Very few studies have explored bias in blocking methods. The only notable work in this area is [33], which focused solely on bias in the RR. In Table IV, we present our evaluation of bias across different blocking methods and datasets. Similar to the field of Fairness in Machine Learning, blocking methods also exhibit various bias metrics that warrant careful consideration.

## C. Discussion

Mohammad-Hossein: later

| Model | AMZ-GOO | | | | WAL-AMZ | | | | DBLP-GOO | | | | DBLP-ACM | | | | BERR | | | | FOD-ZAG | | | | ITU-AMZ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RR | PC | PQ | $F_B$ | RR | PC | PQ | $F_B$ | RR | PC | PQ | $F_B$ | RR | PC | PQ | $F_B$ | RR | PC | PQ | $F_B$ | RR | PC | PQ | $F_B$ | RR | PC | PQ | $F_B$ |
| StandBlock | 99.73 | 98.29 | 9.58 | 99.00 | 99.81 | 99.06 | 0.90 | 99.44 | - | - | - | - | - | - | - | - | 99.91 | 95.59 | 0.54 | 97.70 | 98.72 | 99.10 | 4.88 | 98.91 | - | - | - | - |
| Q-Gram | 99.69 | 95.72 | 8.29 | 97.66 | 99.77 | 99.06 | 0.75 | 99.42 | - | - | - | - | - | - | - | - | 99.90 | 92.65 | 0.51 | 96.14 | 98.83 | 99.10 | 5.31 | 98.96 | - | - | - | - |
| Ext-Q-Gram | 99.70 | 94.17 | 8.40 | 96.86 | 99.75 | 98.75 | 0.69 | 99.25 | - | - | - | - | - | - | - | - | 99.90 | 91.18 | 0.48 | 95.34 | 98.88 | 99.10 | 5.56 | 98.99 | - | - | - | - |
| Suffix | 99.85 | 88.52 | 16.06 | 93.84 | 99.96 | 91.27 | 3.49 | 95.41 | - | - | - | - | - | - | - | - | 99.95 | 88.24 | 0.88 | 93.73 | 99.31 | 99.10 | 9.07 | 99.21 | - | - | - | - |
| Ext-Suffix | 99.86 | 83.89 | 15.85 | 91.18 | 99.97 | 88.57 | 4.32 | 93.92 | - | - | - | - | - | - | - | - | 99.95 | 89.71 | 0.89 | 94.55 | 99.33 | 95.50 | 8.96 | 97.37 | - | - | - | - |
| AUTO | 98.45 | 89.80 | 1.54 | 93.93 | 99.77 | 96.26 | 0.73 | 97.98 | - | - | - | - | - | - | - | - | 98.33 | 83.82 | 0.03 | 90.50 | 84.89 | 99.10 | 0.41 | 91.45 | - | - | - | - |
| CTT | 98.45 | 95.97 | 1.64 | 97.20 | 99.77 | 97.51 | 0.73 | 98.63 | - | - | - | - | - | - | - | - | 98.33 | 94.12 | 0.03 | 96.18 | 84.89 | 99.10 | 0.41 | 91.45 | - | - | - | - |

TABLE II: RR, PC, PQ, and $F_B$ for different models across datasets

| Model | AMZ-GOO | | | WAL-AMZ | | | DBLP-GOO | | | DBLP-ACM | | | BERR | | | FOD-ZAG | | | ITU-AMZ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\Delta$RR | $\Delta$PC | $\Delta F_B$ | $\Delta$RR | $\Delta$PC | $\Delta F_B$ | $\Delta$RR | $\Delta$PC | $\Delta F_B$ | $\Delta$RR | $\Delta$PC | $\Delta F_B$ | $\Delta$RR | $\Delta$PC | $\Delta F_B$ | $\Delta$RR | $\Delta$PC | $\Delta F_B$ | $\Delta$RR | $\Delta$PC | $\Delta F_B$ |
| StandBlock | 0.08 | 1.71 | 0.91 | 0.14 | 1.47 | 0.81 | - | - | - | - | - | - | -0.01 | -1.68 | -0.88 | 0.01 | -0.99 | -0.49 | - | - | - |
| Q-Gram | 0.38 | -1.00 | -0.33 | 0.25 | 1.47 | 0.87 | - | - | - | - | - | - | -0.01 | -6.81 | -3.65 | -0.20 | -0.99 | -0.59 | - | - | - |
| Ext-Q-Gram | 0.41 | 6.16 | 3.53 | 0.18 | 1.13 | 0.66 | - | - | - | - | - | - | -0.01 | -9.37 | -5.09 | -0.16 | -0.99 | -0.57 | - | - | - |
| Suffix | 0.07 | 16.01 | 9.77 | -0.02 | 5.40 | 3.01 | - | - | - | - | - | - | -0.02 | -8.49 | -4.77 | -0.18 | -0.99 | -0.58 | - | - | - |
| Ext-Suffix | 0.04 | 18.16 | 11.79 | -0.00 | 1.17 | 0.66 | - | - | - | - | - | - | -0.02 | -5.92 | -3.28 | -0.17 | -4.95 | -2.61 | - | - | - |
| AUTO | -0.01 | 15.61 | 9.22 | -0.01 | 4.64 | 2.44 | - | - | - | - | - | - | -0.07 | -4.16 | -2.45 | 1.36 | -0.99 | 0.38 | - | - | - |
| CTT | -0.01 | 2.78 | 1.44 | 0.01 | 4.76 | 2.49 | - | - | - | - | - | - | -0.09 | -4.24 | -2.25 | 0.00 | -0.99 | -0.42 | - | - | - |

TABLE III: RR, PC, and $F_B$ disparities for different models across datasets: Major - Minor

| Model | AMZ-GOO | | | | | | WAL-AMZ | | | | | | BERR | | | | | | FOD-ZAG | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $RR_1$ | $RR_2$ | $PC_1$ | $PC_2$ | $F_{B1}$ | $F_{B2}$ | $RR_1$ | $RR_2$ | $PC_1$ | $PC_2$ | $F_{B1}$ | $F_{B2}$ | $RR_1$ | $RR_2$ | $PC_1$ | $PC_2$ | $F_{B1}$ | $F_{B2}$ | $RR_1$ | $RR_2$ | $PC_1$ | $PC_2$ | $F_{B1}$ | $F_{B2}$ |
| StandBlock | 99.66 | 99.73 | 96.67 | 98.37 | 98.14 | 99.05 | 99.68 | 99.82 | 97.73 | 99.20 | 98.69 | 99.51 | 99.91 | 99.90 | 96.55 | 94.87 | 98.20 | 97.32 | 98.71 | 98.72 | 100.00 | 99.01 | 99.35 | 98.87 |
| Q-Gram | 99.33 | 99.72 | 96.67 | 95.66 | 97.98 | 97.65 | 99.54 | 99.79 | 97.73 | 99.20 | 98.62 | 99.49 | 99.91 | 99.90 | 96.55 | 89.74 | 98.20 | 94.55 | 99.00 | 98.80 | 100.00 | 99.01 | 99.50 | 98.90 |
| Ext-Q-Gram | 99.32 | 99.73 | 88.33 | 94.49 | 93.51 | 97.04 | 99.58 | 99.76 | 97.73 | 98.86 | 98.65 | 99.31 | 99.90 | 99.90 | 96.55 | 87.18 | 98.20 | 93.11 | 99.01 | 98.86 | 100.00 | 99.01 | 99.50 | 98.93 |
| Suffix | 99.79 | 99.86 | 73.33 | 89.34 | 84.54 | 94.31 | 99.97 | 99.95 | 86.36 | 91.76 | 92.67 | 95.68 | 99.96 | 99.94 | 93.10 | 84.62 | 96.41 | 91.64 | 99.46 | 99.29 | 100.00 | 99.01 | 99.73 | 99.15 |
| Ext-Suffix | 99.82 | 99.86 | 66.67 | 84.82 | 79.94 | 91.73 | 99.97 | 99.96 | 87.50 | 88.67 | 93.32 | 93.98 | 99.96 | 99.94 | 93.10 | 87.18 | 96.41 | 93.12 | 99.48 | 99.30 | 100.00 | 95.05 | 99.74 | 97.13 |
| AUTO | 98.46 | 98.45 | 75.00 | 90.61 | 85.14 | 94.36 | 99.79 | 99.77 | 92.05 | 96.68 | 95.76 | 98.20 | 98.37 | 98.30 | 86.21 | 82.05 | 91.89 | 89.44 | 83.73 | 85.09 | 100.00 | 99.01 | 91.15 | 91.52 |
| CTT | 98.46 | 98.45 | 93.33 | 96.12 | 95.83 | 97.27 | 99.77 | 99.77 | 93.18 | 97.94 | 96.36 | 98.85 | 98.38 | 98.29 | 96.55 | 92.31 | 97.46 | 95.20 | 84.89 | 84.89 | 100.00 | 99.01 | 91.83 | 91.41 |

TABLE IV: $RR_1$ is minority and the other majority

## V. Related Work

**Fairness in Entity Matching and Blocking** Fairness is a growing concern in EM systems, especially as these systems are widely employed in data-driven decision-making processes. A key concern is that EM processes can unintentionally perpetuate biases present in the data, leading to unfair outcomes. This issue is particularly significant during the blocking stage, where the method of grouping records for comparison can introduce bias, affecting both the accuracy and fairness of the matching process.

Recent research has sought to address these fairness challenges. The FairER algorithm, introduced by [54], incorporates fairness constraints directly into the EM process, highlighting the need to consider fairness from the very beginning of the EM pipeline. This approach is crucial in the blocking stage, where biases in record grouping can skew the entire matching process. Additionally, [32] proposed an AUC-based fairness metric to evaluate how well EM systems perform across different groups. Such metrics could be critical in refining blocking methods to prevent bias from affecting EM outcomes.

Further, [33] emphasized the importance of fairness-aware data preparation, particularly in blocking methods, to avoid bias and ensure that no groups are disproportionately excluded or misrepresented.

**Fairness in Clustering and Ranking** Fairness in clustering and ranking algorithms is assessed using metrics that ensure equitable treatment across demographic groups. In clustering, fairness is evaluated by the balance within clusters, using metrics like demographic parity and balance ratio [55] to measure proportional representation. Techniques such as Fairlet Decomposition [56] utilize these metrics to create balanced groups across sensitive attributes before final clustering, ensuring fair representation. In ranking algorithms, fairness is measured by metrics assessing visibility and exposure across ranked lists. Fairness-Aware Ranking (FA*IR) [57] and exposure fairness metrics, like those in FAIR-PG-RANK [58], aim to ensure balanced exposure for different groups, preventing systematic favoritism or marginalization.

These fairness metrics are essential for developing unbiased algorithms and are crucial when designing metrics for fairness in blocking methods for entity matching. By ensuring equitable block formation, these metrics help identify and mitigate potential biases, promoting fairness throughout the entity matching process.

## VI. Conclusion

## References


[1] 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.

[2] R. Wu, S. Chaba, S. Sawlani, X. Chu, and S. Thirumuruganathan, "Zeroer: Entity resolution using zero labeled examples," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 1149–1164.

[3] C. Fu, X. Han, J. He, and L. Sun, "Hierarchical matching network for heterogeneous entity resolution," in *IJCAI*, 2021, pp. 3665–3671.

[4] Y. Li, J. Li, Y. Suhara, A. Doan, and W.-C. Tan, "Deep entity matching with pre-trained language models," *arXiv preprint arXiv:2004.00584*, 2020.

[5] D. Yao, Y. Gu, G. Cong, H. Jin, and X. Lv, "Entity resolution with hierarchical graph attention networks," in *SIGMOD*, 2022, pp. 429–442.

[6] P. V. Konda, *Magellan: Toward building entity matching management systems*. The University of Wisconsin-Madison, 2018.

[7] G. Simonini, G. Papadakis, T. Palpanas, and S. Bergamaschi, "Schema-agnostic progressive entity resolution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 6, pp. 1208–1221, 2018.

[8] J. Jonas and J. Harper, *Effective counterterrorism and the limited role of predictive data mining*. JSTOR, 2006.

[9] G. Papadakis, D. Skoutas, E. Thanos, and T. Palpanas, "Blocking and filtering techniques for entity resolution: A survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–42, 2020.

[10] R. C. Steorts, S. L. Ventura, M. Sadinle, and S. E. Fienberg, "A comparison of blocking methods for record linkage," in *Privacy in Statistical Databases: UNESCO Chair in Data Privacy, International Conference, PSD 2014, Ibiza, Spain, September 17-19, 2014. Proceedings*. Springer, 2014, pp. 253–268.

[11] M. Michelson and C. A. Knoblock, "Learning blocking schemes for record linkage," in *AAAI*, vol. 6, 2006, pp. 440–445.

[12] M. Bilenko, B. Kamath, and R. J. Mooney, "Adaptive blocking: Learning to scale up record linkage," in *Sixth International Conference on Data Mining (ICDM'06)*. IEEE, 2006, pp. 87–96.

[13] J. MESTS and M. Tang, "Distributed representations of tuples for entity resolution," *Proceedings of the VLDB Endowment*, vol. 11, no. 11, 2018.

[14] G. Papadakis, J. Svirsky, A. Gal, and T. Palpanas, "Comparative analysis of approximate blocking techniques for entity resolution," *Proceedings of the VLDB Endowment*, vol. 9, no. 9, pp. 684–695, 2016.

[15] P. Christen, "A survey of indexing techniques for scalable record linkage and deduplication," *IEEE transactions on knowledge and data engineering*, vol. 24, no. 9, pp. 1537–1555, 2011.

[16] G. Papadakis, D. Skoutas, E. Thanos, and T. Palpanas, "A survey of blocking and filtering techniques for entity resolution," *ACM Reference Format*, vol. 1, no. 1, pp. 1–38, Aug 2020.

[17] B.-H. Li, Y. Liu, A.-M. Zhang, W.-H. Wang, and S. Wan, "A survey on blocking technology of entity resolution," *Journal of Computer Science and Technology*, vol. 35, pp. 769–793, 2020.

[18] A. McCallum, K. Nigam, and L. H. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, Aug 2000, pp. 169–178.

[19] L. O. Evangelista, E. Cortez, A. S. da Silva, and W. Meira Jr, "Adaptive and flexible blocking for record linkage tasks," *Journal of Information and Data Management*, vol. 1, no. 2, pp. 167–181, 2010.

[20] M. Michelson and C. A. Knoblock, "Learning blocking schemes for record linkage," in *Proceedings of the National Conference on Artificial Intelligence*. American Association for Artificial Intelligence, 2006.

[21] A. D. Sarma, A. Jain, A. Machanavajjhala, and P. Bohannon, "An automatic blocking mechanism for large-scale de-duplication tasks," *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:11280427

[22] W. Zhang, H. Wei, B. Sisman, X. L. Dong, C. Faloutsos, and D. Page, "Autoblock: A hands-off blocking framework for entity matching," in *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20)*. Houston, TX, USA: ACM, 2020, p. 10.

[23] M. Ebraheem, S. Thirumuruganathan, S. Joty, M. Ouzzani, and N. Tang, "Distributed representations of tuples for entity resolution," *Proc. VLDB Endow.*, vol. 11, no. 11, pp. 1454–1467, jul 2018. [Online]. Available: https://doi.org/10.14778/3236187.3236198

[24] D. Javdani, H. Rahmani, M. Allahgholi, and F. Karimkhani, "Deepblock: A novel blocking approach for entity resolution using deep learning," in *2019 5th International Conference on Web Research (ICWR)*, 2019, pp. 41–44.

[25] P. Christen and K. Goiser, "Quality and complexity measures for data linkage and deduplication," in *Quality measures in data mining*. Springer, 2007, pp. 127–151.

[26] M. G. Elfeky, V. S. Verykios, and A. K. Elmagarmid, "Tailor: A record linkage toolbox," in *Proceedings 18th International Conference on Data Engineering*. IEEE, 2002, pp. 17–28.

[27] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1171–1180.

[28] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *NIPS*, 2016, pp. 3315–3323.

[29] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012, pp. 214–226.

[30] N. Shahbazi, N. Danevski, F. Nargesian, A. Asudeh, and D. Srivastava, "Through the fairness lens: Experimental analysis and evaluation of entity matching," *Proc. VLDB Endow.*, vol. 16, no. 11, p. 3279–3292, jul 2023. [Online]. Available: https://doi.org/10.14778/3611479.3611525

[31] M. H. Moslemi and M. Milani, "Threshold-independent fair matching through score calibration," in *Proceedings of the Conference on Governance, Understanding and Integration of Data for Effective and Responsible AI*, 2024, pp. 40–44.

[32] S. Nilforoushan, Q. Wu, and M. Milani, "Entity matching with auc-based fairness," in *Big Data*, 2022, pp. 5068–5075.

[33] N. Shahbazi, J. Wang, Z. Miao, and N. Bhutani, "Fairness-aware data preparation for entity matching," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 3476–3489.

[34] M. A. Hernández and S. J. Stolfo, "Real-world data is dirty: Data cleansing and the merge/purge problem," *Data mining and knowledge discovery*, vol. 2, pp. 9–37, 1998.

[35] S. Thirumuruganathan, H. Li, N. Tang, M. Ouzzani, Y. Govind, D. Paulsen, G. Fung, and A. Doan, "Deep learning for blocking in entity matching: a design space exploration," *Proceedings of the VLDB Endowment*, vol. 14, no. 11, pp. 2459–2472, 2021.

[36] 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.

[37] G. Papadakis, E. Ioannou, C. Niederée, and P. Fankhauser, "Efficient entity resolution for large heterogeneous information spaces," in *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011, pp. 535–544.

[38] D. Paulsen, Y. Govind, and A. Doan, "Sparkly: A simple yet surprisingly strong tf/idf blocker for entity matching," *Proceedings of the VLDB Endowment*, vol. 16, no. 6, pp. 1507–1519, 2023.

[39] G. Papadakis, E. Ioannou, T. Palpanas, C. Niederée, and W. Nejdl, "A blocking framework for entity resolution in highly heterogeneous information spaces," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 12, pp. 2665–2682, 2012.

[40] S. Galhotra, D. Firmani, B. Saha, and D. Srivastava, "Efficient and effective er with progressive blocking," *The VLDB Journal*, vol. 30, no. 4, pp. 537–557, 2021.

[41] H. Li, P. Konda, P. S. GC, A. Doan, B. Snyder, Y. Park, G. Krishnan, R. Deep, and V. Raghavendra, "Matchcatcher: A debugger for blocking in entity matching." in *EDBT*, 2018, pp. 193–204.

[42] A. Zeakis, G. Papadakis, D. Skoutas, and M. Koubarakis, "Pre-trained embeddings for entity resolution: an experimental analysis," *Proceedings of the VLDB Endowment*, vol. 16, no. 9, pp. 2225–2238, 2023.

[43] A. D. Sarma, A. Jain, A. Machanavajjhala, and P. Bohannon, "Cblock: An automatic blocking mechanism for large-scale de-duplication tasks," *arXiv preprint arXiv:1111.3689*, 2011.

[44] K. O'Hare, A. Jurek-Loughrey, and C. d. Campos, "A review of unsupervised and semi-supervised blocking methods for record linkage," *Linking and Mining Heterogeneous and Multi-view Data*, pp. 79–105, 2019.

[45] K. O'Hare, A. Jurek, and C. de Campos, "A new technique of selecting an optimal blocking method for better record linkage," *Information Systems*, vol. 77, pp. 151–166, 2018.

[46] H. Köpcke and E. Rahm, "Frameworks for entity matching: A comparison," *Data & Knowledge Engineering*, vol. 69, no. 2, pp. 197–210, 2010.

[47] M. Kejriwal and D. P. Miranker, "An unsupervised algorithm for learning blocking schemes," in *2013 IEEE 13th International Conference on Data Mining*. IEEE, 2013, pp. 340–349.

[48] J. B. Mugeni and T. Amagasa, "A graph-based blocking approach for entity matching using contrastively learned embeddings," *ACM SIGAPP Applied Computing Review*, vol. 22, no. 4, pp. 37–46, 2023.

[49] T. De Vries, H. Ke, S. Chawla, and P. Christen, "Robust record linkage blocking using suffix arrays," in *Proceedings of the 18th ACM*

*conference on Information and knowledge management*, 2009, pp. 305–314.

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

[51] N. Shahbazi, N. Danevski, F. Nargesian, A. Asudeh, and D. Srivastava, "Through the fairness lens: Experimental analysis and evaluation of entity matching," *Proc. VLDB Endow.*, vol. 16, no. 11, p. 3279–3292, 2023.

[52] I. P. Fellegi and A. B. Sunter, "A theory for record linkage," *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, 1969.

[53] G. Papadakis, G. Alexiou, G. Papastefanatos, and G. Koutrika, "Schema-agnostic vs schema-based configurations for blocking methods on homogeneous data," *Proceedings of the VLDB Endowment*, vol. 9, no. 4, pp. 312–323, 2015.

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

[55] L. E. Celis, D. Straszak, and N. K. Vishnoi, "Fairness first: Clustering in a multi-stage approach for mitigating bias," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2023.

[56] A. Chhabra, K. Masalkovaitė, and P. Mohapatra, "An overview of fairness in clustering," *IEEE Access*, vol. 9, pp. 130 698–130 720, 2021.

[57] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates, "Fa*ir: A fair top-k ranking algorithm," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. ACM, 2017.

[58] A. Singh and T. Joachims, "Fair-pg-rank: Fairness-aware learning to rank with policy gradients," in *Proceedings of the 2020 ACM International Conference on Web Search and Data Mining (WSDM '20)*. ACM, 2020.