Evaluating Blocking Biases in Entity Matching

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 sources refer to the same real-world object. As data sources expand and become more heterogeneous, the challenge of accurately and efficiently matching entities has intensified. EM, also known as entity linkage or record matching, is fundamental in data integration, with broad applications across industries [1], [2], [3], [4], [5], [6], [7]. In commercial sectors, EM is crucial for tasks like matching customer or product records across databases, while in healthcare and security, accurate EM can have significant implications [8]. For instance, matching patient records across hospitals is vital to ensure comprehensive care and avoid duplication despite variations in data entry.

EM systems typically consist of a matching component that compares entities and labels them as either "match" or "non-match." A major challenge in EM is its computational complexity, often scaling quadratically as each entity must be compared to all others, making it an $O(n^2)$ problem. This complexity becomes prohibitive for large datasets with millions of records. To mitigate this, blocking methods are used as a preliminary step to reduce the number of comparisons [9], [10], [11]. As a result, EM systems typically operate in two phases: blocking to limit comparisons, followed by matching to produce 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 enhances computational efficiency and addresses the inherent quadratic complexity of EM.

Blocking methods have evolved significantly, spanning from heuristic-based approaches to advanced deep-learning techniques. Traditional methods such as Standard Blocking and Sorted Neighborhoods [12], [13] laid the groundwork, with Standard Blocking categorizing records based on a blocking key, like initials. At the same time, Sorted Neighborhood employs a sliding window over sorted records to enhance efficiency. Advanced techniques like Canopy Clustering [14] and recent machine learning-driven methods such as BSL (Blocking Scheme Learner) and BGP (Blocking based on Genetic Programming) [15], [16] further refine blocking schemes by optimizing attribute selection and comparison strategies. Additionally, frameworks like AutoBlock and DeepER [17], [18] utilize deep learning for representation learning and efficient candidate generation. The effectiveness of these methods is typically assessed using metrics like Reduction Ratio (RR), Pair Quality (PQ), Pair Completeness (PC), and their harmonic mean [19], [20], [21].

In recent years, fairness in ML has gained significant attention [22], [23], [24] 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 [25], [26], [27]. 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 [28], 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.

Traditional fairness metrics, such as Equalized Odds, Equal Opportunity, and Statistical Parity [23], [24], 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.

This paper presents an experimental study of blocking methods, focusing on fairness issues, detecting biases, and examining their impact on EM and end-to-end matching tasks. In the remainder of this paper, Section II presents related work, including a summary of the state-of-the-art blocking methods. Section III provides a formal definition of EM and blocking and reviews the metrics used to evaluate blocking. In Section IV, we extend these blocking metrics to fairness measures, proposing them as a sufficient method for assessing bias in blocking. Section V presents our experimental results, highlighting the effectiveness of various blocking methods and the biases they may introduce. Section VI offers our conclusions. All the implementations are available at https://github.com/mhmoslemi2338/pre-EM-bias.

II. Related Work

We briefly review the existing blocking methods for EM and then discuss fairness in EM and related areas such as clustering and ranking.

A. Blocking Methods

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 [29], [21], [30]). 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. Mosatafa: It is not clear what redundancy- categories are. Please add a few sentences to clarify 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 [30].

Traditional blocking methods such as Standard Blocking and Sorted Neighborhoods have fundamentally shaped the field [12], [13]. 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.

Advanced approaches like Canopy Clustering [14] have been introduced to address some of the limitations of traditional methods. 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 [16].

Blocking techniques, such as BSL (Blocking Scheme Learner) [15] and BGP (Blocking based on Genetic Programming) [16], have further enhanced the efficiency and accuracy of the traditional methods. These more advanced 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 [31]. 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.

Deep learning has recently 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 [17], [18]. 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 [32].

Various metrics have been used to measure the quality of blocking. Still, three are most commonly used and considered the most comprehensive in assessing the effectiveness of blocking methods: RR, PQ, and PC [19], [20], [21]. 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, and PC indicates the percentage of true matches in the candidate set after blocking.

There is little agreement on the evaluation measures in the literature on blocking. Some studies only consider PC and RR as their primary metrics [17], [33], [34], [35], [36], [37], [9], [10], [38], while some only considered PC and PQ [32], [39]. Several studies also consider runtime as a main evaluation measure in blocking methods [40], [41], [42], [43]. There is also research that employs the harmonic mean of PC and RR [44], [45], [46], [47], [48] or the harmonic mean of PC

and PQ [21]. We will formally define these metrics in Section III and discuss what measure provides a more meaningful way to evaluate the methods considered in Section V.

B. Fairness in EM 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. Mosatafa: I suggest giving a few real-world examples of fairness issues in EM, e.g., nofly list, etc that we had before. Mohammad-Hossein: A real-world example that underscores these fairness issues is the use of "no-fly" lists, where individuals, particularly people of color, have been wrongly flagged as security threats due to misidentification, resulting in denial of services. In such cases, the blocking and matching algorithms often fail to consider cultural and linguistic diversity, leading to higher rates of false positives among minority groups [49]. Another notable example is Google's advertising algorithms, which have been found to display prestigious job advertisements more frequently to men than to women. This bias occurs because the algorithms tend to match male profiles more closely with the job requirements [50]. These discrepancies highlight how EM algorithms can perpetuate existing gender biases, resulting in unequal opportunities and outcomes. These examples illustrate the necessity for developing blocking and matching algorithms that ensure fair and equitable treatment across all demographic groups.

Recent research has sought to address these fairness challenges. The FairER algorithm, introduced by [51], 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, [27] 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. Recent work in [28], [52] extensively studies biases in EM as a data preparation task to avoid bias and ensure that no groups are disproportionately excluded or misrepresented. Still, they mainly focus on EM and barely discuss blocking.

Blocking can be seen as clustering. We, therefore, briefly review fairness concepts in clustering as a related task. Fairness in clustering 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 [53] to measure proportional representation. Techniques such as *Fairlet Decomposition* [54] utilize these metrics to create balanced groups across sensitive attributes before final clustering, ensuring fair representation. While having balanced clusters is a reasonable fairness requirement for clustering, it does not apply to blocking, where the goal is not equal representation within blocks but rather to create small blocks that include all equivalent record pairs.

III. Background

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

EM generally has two major phases: blocking and matching [1]. We start with matching and then explain blocking.

A. Entity Matching

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

A entity matcher (matcher in short) for entities of schema $\mathcal{S}$ is a binary classifier $f : \mathcal{D} \times \mathcal{D} \mapsto \{0, 1\}$; that labels entity pairs 1, called they match, or 0, saying they don't match. Given relations D_1, D_2 with schema $\mathcal{S}$ and equivalent entity pairs $M \subseteq D_1 \times D_2$, the goal of f is to find the equivalent entities in M, i.e., label the entity pairs in M as a match and the non-equivalent entities 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 entity pairs are correctly labeled.

B. Blocking

The set of entity 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; $M \subseteq C$. A blocking method is obviously expected to run much faster than a matching method for comparing all possible pairs. Blocking saves unnecessary checking of some of the non-equivalent pairs while searching for the equivalent pairs.

Definition III.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. $|C| \ll |P|$ (there are much fewer candidates compared to total entity Paris) and $M \subseteq C$ in a much less time compared to matching methods.*

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 entity pairs. This measure does not consider the quality of the generated candidate entity 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 [55]. 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 [30]. PQ ranges from 0 to 1, with higher values indicating that the blocker is more effective at eliminating non-matches.

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.

There is no clear consensus on which blocking measures are most important or how they relate to each other. Some studies [29], [42], [35], [47] suggest that PQ is typically sensitive to small changes in RR. We observe in experiments that blocking is generally applied to datasets with millions of pairs with a much smaller number of true matches. Therefore, a small change in RR leads to a significant increase in the size of the candidate set that greatly impacts PQ, which explains the high sensitivity of PQ. Therefore, in this study, we focus on PC, RR, their harmonic mean, and runtime as the key metrics for evaluating blocking methods. The harmonic mean, which balances the trade-off between PC and RR, is defined as:

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

A detailed numerical analysis of these measures and their relationships is provided in Section V. We explain these measures using an example.

Example III.2. *Figure 1 shows a relation D with ten entities. We consider matching entities in the same relation entity pairs $P = D \times D$. The red dotted lines show the equivalent pairs; $M = \{(t_2, t_6), (t_6, t_7), (t_2, t_7), (t_4, t_5), (t_8, t_9), (t_8, t_{10}), (t_9, t_{10})\}$. The solid black lines show blocking methods that specify three blocks that result in $3 + 6 + 3 = 12$ candidate pairs. The blocking result misses two equivalent pairs, (t_2, t_6) and (t_2, t_7), but reduces the number of pairs to check from 45 to 12, resulting in RR $= 1 - \frac{12}{45} \approx 0.73$, PC $= \frac{5}{7} \approx 0.71$, and $F_{RR,PC} \approx 0.72$.*

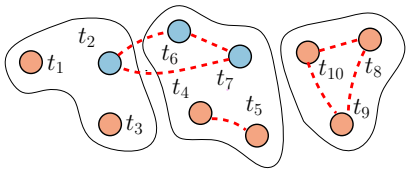


Fig. 1: Disparity in blocking: minority and majority entities are highlighted in blue and red resp. and the equivalent pairs are linked by dotted lines. Solid lines show the blocks.

IV. Measuring Bias in Blocking

Blocking methods can suffer from disparities; the quality of blocking might differ for the minority group compared with the majority, 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.

To define biases in blocking, we assume the relation schema $\mathcal{S}$ includes a sensitive (or protected) attribute A (e.g., gender) with domains $\text{DOM}(A) = a, b$, where a and b correspond to the minority (e.g., female or nonbinary) and majority groups (e.g., male), respectively. This means that an entity t with $t[A] = a$ belongs to the minority group, while $t[A] = b$ indicates it belongs to the majority group. In the context of matching, we use the protected attribute of entities in a record pair to determine whether the pair concerns minority or majority groups. We define a minority pair as any pair of entities $(t_1, t_2) \in D_1 \times D_2$ where at least one entity belongs to a minority group based on the specified sensitive attribute (e.g., ethnicity, gender). Conversely, a majority pair consists of both entities from the majority group. A minority pair is thus defined as any pair that could lead to an entity matching (EM) decision that negatively impacts a minority entity, either by incorrectly matching it with another minority or majority entity or by missing the opportunity to correctly match it with another minority entity.

In our definition of blocking disparities, 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 and pair completeness for the minority groups g as follows:

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

Intuitively, the reduction ratio per group g specifies the reduction in the number of comparisons for the entity pairs in group g. Similarly, pair completeness per group g measures these measures between the pairs in the group g only. We don't define and use PQ as we explained due to its sensitivity and redundancy with RR and PC. Using RR and PC per group, we define their harmonic mean per group g that is $\text{F}^g_{\text{RR,PC}}$. We now use our running example to explain these quality measures per group g.

We define disparities as the differences between the blocking quality measures for the majority and minority groups. For example, $\Delta\text{RR} = \text{RR}_b - \text{RR}_a$ represents the reduction ratio disparity, $\Delta\text{PC} = \text{PC}_b - \text{PC}_a$ denotes the pair completeness disparity, and $\Delta\text{F}_{\text{RR,PC}} = \text{F}^b_{\text{RR,PC}} - \text{F}^a_{\text{RR,PC}}$ indicates the mean disparity. An important observation from our experimental analysis is that these disparities can be negative; this contrasts with general disparities in fairness literature, such as demographic disparity, where disparities are typically assumed to be non-negative (i.e., minorities can only be discriminated against). In the context of blocking, some methods may actually perform better for minority groups due to the specific nature of the blocking task—a topic we further discuss in the experimental section.

Example IV.1. *Continuing with the example in Figure 1, three entities are from the minority group, and seven are from the majority groups, as highlighted by different colors. These give five majority pairs (with both entities from the majority group) and* $3\times7+3 = 24$ *minority pairs (with at least one entity 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),* $PC_a = 1/3 \approx 0.33$*,* $F^a_{RR,PC} \approx 0.45$ $RR_b = 1 - 5/21 = 0.76$*,* $PC_b = 4/4 = 1$*, and* $F^b_{RR,PC} \approx 0.86$*. These measures give reduction ratio and pair completeness disparities of* $\Delta RR \approx 0.76 - 0.71 = 0.05$*,* $\Delta PC \approx 0.66$*, and* 0.45*, respectively. In this example, all disparities are positive, indicating a lower blocking quality for the minority group a. This is evident in the missing equivalences for the minority pairs (2 missing pairs for the minority vs. none for the majority) and the smaller reduction gain for the minority group (reduction from 24 to 7 for the minority vs. 21 to 5 for the majority).*

Introducing these disparity metrics provides a tool for evaluating the fairness of blocking techniques. By quantifying the differences in the blocking quality between demographic groups, it becomes possible to identify potential biases and inequalities in the blocking process.

V. Evaluation and Analysis

The purpose of our experiments is twofold: first, to understand the quality of the existing blocking methods for the benchmarks we employed in this paper where we use RR, PC, $\text{F}_{\text{RR,PC}}$, and time to compare; second, to analyze the same methods in terms of their possible biases using the introduced measures of disparity.

A. Experimental Setup

We briefly explain the datasets and the blocking methods used in this paper before presenting the experimental results.

1) 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], [56]. The research community commonly adopts these datasets to evaluate EM system performance. Their full details are available in our GitHub repository.

Consistent with prior studies [27], [52], [26], we categorize entities into minority and majority groups across various datasets. For instance, in `DBLP-ACM`, including a female name in the "authors" attribute defines the minority. Similarly, entities in the `FEBRL` dataset are considered a minority if the "Given Name" attribute is female. In `FOD-ZAG`, entities with the "Type" attribute exactly equal to "Asian" are classified as minority. In `AMZ-GOO`, the presence of "Microsoft" in the "manufacturer" attribute signifies a minority group. For `WAL-AMZ`, entities classified as "printers" under the "category" attribute are considered a minority group. In `DBLP-GOO`, entities are considered minority if the "venue" attribute includes "vldb j,". For `BERR`, entities with "Beer Name" containing the phrase "red" are classified as minority. Finally, in `ITU-AMZ`, the minority group includes those where the "Genre" attribute contains the word "Dance,".

Detailed statistical information about these datasets is provided in Table I. The numbers in the bracket refer to the corresponding parameter in the minority group, e.g., $2.6k$ (96) for $|D_1|$ in `WAL-AMZ` means there are $2.6k$ entities in D_1 while only 96 are a minority. The parameters for the majority group can be clearly inferred from the data in the table.

Dataset	#Attr.	$\lvert D_1\rvert$	$\lvert D_2\rvert$	$\lvert P\rvert$	$\lvert M\rvert$
`WAL-AMZ`	5	2.6k (96)	22.0k (172)	56.4m (2.5m)	962 (88)
`BERR`	4	4.3k (1.3k)	3.0k (932)	13.0m (6.8m)	68 (29)
`AMZ-GOO`	3	1.4k (83)	3.2k (4)	4.4m (272.9k)	1.2k (60)
`FOD-ZAG`	6	533 (72)	331 (3)	176.4k (25.2k)	111 (10)
`ITU-AMZ`	8	6.9k (1.9k)	55.9k (12.7k)	386.2m	132 (40)
`DBLP-GOO`	4	2.6k (191)	64.3k (389)	168.1m	5.3k (403)
`DBLP-ACM`	4	2.6k (251)	2.3k (225)	6.0m (1.1m)	2.2k (310)

TABLE I: Datasets and their characteristics.

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

i) *Standard Blocking (`StdBlck`)*: 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 [57].

ii) *Q-Grams (`QGram`) and Extended Q-Grams Blocking (`XQGram`)*: 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 [21], [58].

iii) *Suffix (`Suffix`) and Extended Suffix (`XSuffix`) 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 [21], [58].

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].

v) *Semantic-Based Graph Blocking (GRAPH)*: The algorithm concatenates entity attributes into a single string, from which a transformer-based model extracts context-aware embeddings. These embeddings are reduced to a lower-dimensional space for efficiency. A k-nearest neighbor (KNN) algorithm then constructs a graph of database tuples, connecting nodes based on embedding proximity. Unsupervised clustering is applied to group similar entities, framing the blocking task as a graph clustering problem [48].

Mosatafa: Add a few sentences about why these methods are selected.

B. Experimental Results

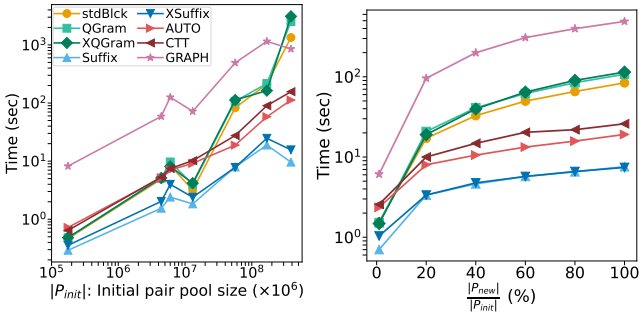
Our experimental results include an analysis of blocking methods based on their performance—specifically, the quality of blocking and runtime—which aligns with our first experimental objective. Additionally, we conduct an extensive analysis of the existing biases in these methods to address our second objective.

1) Runtime and Scalability: Perhaps the very first concern in blocking is to check whether the methods run fast compared to the matching methods, and whether they scale when the data size grows. Table II and Figure 2 show the runtime of the blocking methods for datasets of varying sizes (dataset size is the number of entity pairs). The results represent the average runtime of the methods over five runs per dataset. The standard deviation was minimal compared to the average values ($<$ 0.5 sec) and is therefore not reported. The main observation is that all methods run within a reasonable timeframe ($<$ 1 hour), even for datasets with up to 500 million pairs, with some methods completing in less than a minute for the largest datasets.

Another important observation is that not all methods scale equally; some are more suitable for handling large volumes of data. This analysis is crucial for investigating biases in blocking because understanding the runtime superiority of any method allows us to make more informed judgments about their biases as well.

Model	AMZ-GOO	WAL-AMZ	DBLP-GOO	DBLP-ACM	BERR	FOD-ZAG	ITU-AMZ
StdBlck	4.9s	1.4m	3.5m	8.5s	3.2s	0.5s	22.3m
QGram	5.1s	1.8m	3.6m	9.7s	4.0s	0.5s	42.0m
XQGram	5.1s	1.9m	2.7m	8.1s	4.2s	0.5s	51.8m
Suffix	**1.5s**	7.9s	**18.9s**	**2.4s**	**1.8s**	**0.3s**	**9.6s**
XSuffix	2.0s	**7.8s**	24.7s	4.0s	2.4s	0.4s	15.7s
AUTO	5.2s	18.9s	58.1s	7.0s	9.2s	0.7s	1.9m
CTT	5.3s	27.6s	1.5m	7.5s	10.2s	0.6s	2.6m
GRAPH	58.9s	8.2m	19.2m	2.1m	1.2m	8.2s	14.2m

TABLE II: Runtime of blocking methods



a) All datasets b) WAL-AMZ

Fig. 2: Runtime of blocking methods

Model	AMZ-GOO	WAL-AMZ	DBLP-GOO	DBLP-ACM	BERR	FOD-ZAG	ITU-AMZ
StdBlck	99.73	99.81	99.94	99.94	99.91	98.72	99.86
QGram	99.69	99.77	99.95	99.94	99.90	98.83	99.79
XQGram	99.70	99.75	99.95	99.94	99.90	98.88	99.77
Suffix	99.85	99.96	99.98	99.95	**99.95**	99.31	**99.99**
XSuffix	**99.86**	**99.97**	**99.99**	99.94	**99.95**	**99.33**	**99.99**
AUTO	98.45	99.77	99.92	97.82	98.33	84.89	99.91
CTT	98.45	99.77	99.92	97.82	98.33	84.89	99.91
GRAPH	97.74	97.59	98.14	**99.96**	98.91	80.17	98.37

TABLE III: RR values for different models across datasets, with best (bold), second-best (underlined), and worst (framed) highlighted

2) Quality of Blocking: Table III shows the RR (reduction ratios) of blocking methods across various datasets. Almost all methods achieve a high RR close to 1, with Suffix and XSuffix slightly outperforming the others in most datasets, which could be attributed to [insert reason here]. However, some methods underperform in specific datasets, such as AUTO, CTT, and GRAPH in FOD-ZAG, likely due to [insert reason here]. The overall RR is lower in FOD-ZAG, which might be due to the smaller dataset size.

Mosatafa: Can we claim larger datasets will usually have higher RR as there is more opportunity to reduce? smaller datasets are harder to get higher RR?

Model	AMZ-GOO	WAL-AMZ	DBLP-GOO	DBLP-ACM	BERR	FOD-ZAG	ITU-AMZ
StdBlck	**98.29**	**99.06**	98.73	99.86	**95.59**	**99.10**	**97.73**
QGram	95.72	**99.06**	**98.75**	**99.95**	92.65	**99.10**	73.48
XQGram	94.17	98.75	97.59	**99.95**	91.18	**99.10**	71.97
Suffix	88.52	91.16	82.38	99.91	88.24	**99.10**	50.76
XSuffix	83.89	88.36	76.47	99.46	89.71	96.40	51.52
AUTO	88.52	96.36	95.23	99.86	85.29	**99.10**	90.15
CTT	95.97	97.51	95.96	99.86	94.12	**99.10**	91.67
GRAPH	93.92	83.89	79.13	98.78	88.24	20.54	54.55

TABLE IV: PC values for different models across datasets

Table IV demonstrates the PC values, where, unlike RR, not all methods perform equally well. For instance, Suffix, XSuffix, and AUTO underperform in AMZ-GOO, while the other models achieve high PC. In ITU-AMZ, QGram, XQGram, Suffix, and XSuffix all exhibit significantly

lower performance, missing many equivalences, whereas StdBlck, AUTO, and CTT preserve most of them. This disparity is due to the varying characteristics of the datasets, highlighting that no single method consistently outperforms others. Therefore, it suggests that specific blocking methods should be employed based on the dataset.

Mosatafa: Question: why some models perform well and some don't? What data characteristics and model properties decide? What is the general guideline to decide what blocking method to use for a given dataset? Mohammad-Hossein: The performance of different models in blocking can vary significantly based on the characteristics of the data and the properties of the models themselves. There is no one-size-fits-all rule for model selection; rather, a nuanced understanding of the specific requirements of the task is essential. Key considerations include whether labeled data is available and whether there is prior knowledge of which matcher will be employed at the next stage of the EM pipeline, as this can guide the choice of blocking methods. Additionally, selecting the best blocking method involves the balance between RR and PC, which requires thoughtful tuning and debugging. Moreover, choosing the relevant data columns that contribute most effectively to distinguishing entities is crucial. These considerations are comprehensively addressed in [6], which offers a guideline for the entire EM process, including both the blocking and matching steps.

Model	AMZ-GOO	WAL-AMZ	DBLP-GOO	DBLP-ACM	BERR	FOD-ZAG	ITU-AMZ
StdBlck	9.58	0.90	5.34	66.26	0.54	4.88	0.02
QGram	8.29	0.75	6.22	59.22	0.51	5.31	0.01
XQGram	8.40	0.69	5.99	60.25	0.48	5.56	0.01
Suffix	16.06	3.49	**16.61**	**73.06**	0.88	9.03	**0.24**
XSuffix	**15.85**	**4.31**	17.78	60.34	**0.89**	**9.04**	0.19
AUTO	1.52	0.73	3.89	1.69	0.03	0.41	0.03
CTT	1.64	0.73	3.92	1.69	0.03	0.41	0.04
GRAPH	0.45	0.01	0.01	61.75	0.02	0.03	0.001

TABLE V: PQ values for different models across datasets

The data in Table V confirms our claim that PQ is highly sensitive to RR and can change significantly with a small variation in RR. For example, in the case of AMZ-GOO, a slight decrease in RR between Suffix (99.85) and AUTO (98.45) leads to a substantial drop in PQ (from 16.06 to 1.52). This can be attributed to the large number of pairs and the fact that PQ heavily depends on the number of candidate pairs. A similar pattern can be observed when comparing Suffix and XQGram in WAL-AMZ, as well as XSuffix and CTT in DBLP-ACM. Additionally, it is noteworthy that the drop in PQ is less pronounced for smaller datasets such as BERR and ITU-AMZ, which aligns with and further supports our explanation regarding the impact of dataset size.

As the final quality measure, we report $F_{RR,PC}$ (the harmonic mean of PC and RR) in Table VI. The results indicate that while there is no single dominant method across all datasets, which aligns with our observations in PC, certain methods, such as StdBlck, consistently perform well across various

Model	AMZ-GOO	WAL-AMZ	DBLP-GOO	DBLP-ACM	BERR	FOD-ZAG	ITU-AMZ
StdBlck	**99.00**	**99.44**	99.33	99.90	**97.70**	98.91	**98.78**
QGram	97.66	99.42	**99.34**	**99.95**	96.14	98.96	84.64
XQGram	96.86	99.25	98.75	**99.95**	95.34	98.99	83.62
Suffix	93.84	95.36	90.33	99.93	93.73	**99.20**	67.34
XSuffix	91.18	93.80	86.66	99.70	94.55	97.84	68.00
AUTO	93.22	98.04	97.52	98.83	91.35	91.45	94.78
CTT	97.20	98.63	97.90	98.83	96.18	91.45	95.61
GRAPH	95.78	90.22	87.61	99.37	93.27	32.70	70.18

TABLE VI: $F_{RR,PC}$ values for different models across datasets

datasets, demonstrating more stable and reliable results overall. Mosatafa: Please say something about why, also add any other claim about $F_{RR,PC}$.

Mosatafa: Up to hear.

3) Bias Analysis of Blocking Methods: We now turn our analysis to studying biases in the blocking methods. We start with disparities in the reduction ratio, ΔRR. Table **??** shows RR per minority and majority group together with their difference, which defines the disparity per dataset and per method. The disparities are insignificant (less than one percentage point) across all blocking methods and datasets. This can be attributed to the fact that RR was close to 1, its maximum value, as reported in Table III. However, some methods exhibit higher disparities, such as GRAPH in most datasets and AUTO in DBLP-ACM. These methods also have slightly lower overall RR, as reported in Table III.

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

[10] 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.

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

[12] 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.

Model	AMZ-GOO	WAL-AMZ	DBLP-GOO	DBLP-ACM	BERR	FOD-ZAG	ITU-AMZ
StdBlck	0.08 (99.73, 99.66)	0.14 (99.82, 99.68)	-0.05 (99.90, 99.95)	**-0.01** (99.86, 99.87)	**-0.01** (99.91, 99.92)	0.01 (98.72, 98.71)	-0.13 (99.86, 99.99)
QGram	0.38 (99.72, 99.33)	0.25 (99.79, 99.54)	-0.03 (99.90, 99.93)	**-0.01** (99.95, 99.96)	**-0.01** (99.90, 99.91)	-0.20 (98.80, 99.00)	-0.15 (99.79, 99.94)
XQGram	0.41 (99.73, 99.32)	0.18 (99.76, 99.58)	-0.03 (99.90, 99.93)	**-0.01** (99.95, 99.96)	**-0.01** (99.90, 99.91)	-0.16 (98.86, 99.01)	-0.20 (99.77, 99.99)
Suffix	0.07 (99.86, 99.79)	-0.01 (99.95, 99.97)	**-0.00** (99.94, 99.96)	**-0.01** (99.91, 99.92)	-0.02 (99.94, 99.96)	-0.18 (99.28, 99.46)	**-0.00** (99.99, 99.99)
XSuffix	0.04 (99.86, 99.82)	**-0.00** (99.96, 99.97)	**0.00** (99.94, 99.96)	**-0.01** (99.91, 99.94)	-0.02 (99.96, 99.98)	-0.18 (99.30, 99.48)	**-0.00** (99.99, 99.99)
AUTO	**-0.01** (98.45, 98.46)	-0.01 (99.77, 99.79)	0.01 (98.36, 98.31)	-0.48 (85.02, 84.12)	0.04 (98.36, 98.31)	0.90 (85.02, 84.12)	-0.04 (99.91, 99.94)
CTT	**-0.01** (98.45, 98.46)	0.01 (99.77, 99.77)	0.01 (98.29, 98.38)	-0.21 (84.89, 84.89)	-0.09 (98.29, 98.38)	**0.00** (84.89, 84.89)	-0.04 (99.91, 99.99)
GRAPH	0.30 (97.74, 97.45)	0.81 (97.62, 96.81)	1.39 (98.79, 99.03)	**-0.01** (98.78, 99.00)	-0.24 (99.03, 99.27)	-2.99 (79.74, 82.73)	-1.09 (99.94, 99.99)

TABLE VII: For each cell, a (b,c) shows RR values within minority and majority groups (b,c) and the RR disparities (a)

Model	AMZ-GOO	WAL-AMZ	DBLP-GOO	DBLP-ACM	BERR	FOD-ZAG	ITU-AMZ
StdBlck	1.71 (98.37, 96.67)	1.47 (99.20, 97.73)	0.77 (98.73, 97.96)	-0.16 (99.86, 99.88)	-1.68 (94.87, 96.55)	**-0.99** (99.01, 98.00)	7.50 (99.99, 92.49)
QGram	**-1.00** (95.66, 96.66)	1.47 (99.20, 97.73)	1.33 (99.20, 97.87)	**-0.05** (99.95, 99.97)	-6.81 (89.74, 96.55)	**-0.99** (99.01, 98.00)	-9.35 (92.49, 99.99)
XQGram	6.16 (94.49, 88.33)	1.13 (98.86, 97.73)	1.15 (98.86, 97.73)	**-0.05** (99.95, 99.97)	-9.37 (87.18, 96.55)	**-0.99** (99.01, 98.00)	**-0.76** (99.99, 99.75)
Suffix	16.01 (89.34, 73.33)	5.28 (91.65, 86.36)	**0.27** (99.27, 98.00)	-0.10 (99.91, 99.92)	-8.49 (84.62, 93.10)	**-0.99** (99.01, 98.00)	1.09 (99.99, 98.91)
XSuffix	18.16 (84.82, 66.67)	**0.94** (88.44, 87.50)	1.66 (98.91, 97.74)	-0.63 (99.46, 99.91)	-5.92 (87.18, 93.10)	-3.96 (99.48, 99.91)	2.17 (99.99, 97.82)
AUTO	8.98 (88.98, 80.00)	4.75 (96.80, 92.05)	-0.60 (96.80, 96.55)	-0.16 (85.02, 85.18)	**-1.59** (84.62, 86.21)	**-0.99** (99.01, 98.00)	-6.96 (99.91, 93.49)
CTT	2.78 (96.12, 93.33)	4.76 (97.94, 93.18)	0.46 (99.91, 99.46)	-0.16 (85.18, 85.02)	-4.24 (92.31, 98.31)	**-0.99** (99.01, 98.00)	-1.20 (99.99, 98.79)
GRAPH	2.37 (94.04, 91.67)	17.29 (97.62, 96.81)	-7.28 (98.78, 89.52)	-1.04 (99.00, 98.76)	-2.48 (99.09, 96.61)	-11.78 (79.74, 82.73)	-4.24 (99.99, 97.82)

TABLE VIII: PC disparities and PC values within the minority and majority groups

Model	AMZ-GOO	WAL-AMZ	DBLP-GOO	DBLP-ACM	BERR	FOD-ZAG	ITU-AMZ
StdBlck	0.91 (99.05, 98.14)	0.81 (99.51, 98.69)	0.37 (98.73, 98.36)	-0.08 (99.88, 99.96)	**-0.88** (97.32, 98.20)	-0.49 (98.87, 98.91)	3.83 (99.99, 97.32)
QGram	**-0.33** (97.65, 97.98)	0.87 (99.49, 98.62)	0.66 (98.87, 98.21)	**-0.03** (99.97, 99.98)	-3.65 (94.55, 98.20)	-0.59 (98.90, 98.91)	-6.13 (99.99, 93.32)
XQGram	3.53 (97.04, 93.51)	0.66 (99.31, 98.65)	0.58 (98.75, 98.17)	**-0.03** (99.97, 99.98)	-5.09 (93.11, 98.20)	-0.57 (98.93, 98.91)	**-0.58** (99.99, 99.45)
Suffix	9.77 (94.31, 84.54)	2.95 (95.62, 92.67)	0.16 (90.33, 90.50)	-0.06 (99.92, 99.96)	-4.77 (91.64, 96.41)	-0.58 (99.15, 99.06)	0.96 (99.99, 98.85)
XSuffix	11.79 (91.73, 79.94)	**0.53** (93.85, 93.32)	1.07 (86.66, 85.59)	-0.32 (99.91, 99.77)	-3.28 (93.12, 96.40)	-2.09 (99.48, 99.91)	1.90 (99.99, 98.82)
AUTO	5.20 (93.48, 88.28)	2.50 (98.26, 95.76)	-0.30 (91.40, 91.70)	-0.32 (85.18, 85.02)	-0.89 (90.97, 91.86)	0.11 (91.49, 91.38)	-3.81 (99.94, 97.49)
CTT	1.44 (97.27, 95.83)	2.49 (98.85, 96.36)	**0.24** (95.20, 94.96)	-0.18 (91.41, 91.59)	-2.25 (95.20, 96.18)	**-0.42** (91.41, 91.59)	-0.67 (99.99, 99.32)
GRAPH	1.39 (95.86, 94.47)	11.13 (91.14, 80.01)	-3.70 (92.62, 86.01)	-0.53 (99.00, 98.52)	-1.49 (99.27, 97.78)	16.37 (82.73, 79.74)	-3.76 (99.99, 96.18)

TABLE IX: $F_{RR,PC}$ disparities and $F_{RR,PC}$ values within the minority and majority groups

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

[31] 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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[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] ACLU, "Airline "no fly" lists trample the rights of people of color; seattle should not allow hotels to create a similar system," https://www.aclu-wa.org/story/airline-%E2%80%9Cno-fly%E2%80%9D-lists-trample-rights-people-color-seattle-should-not-allow-hotels-create--0, 2020, accessed: 2024-08-31.

[50] A. Peterson, "Google's algorithm shows prestigious job ads to men but not to women. here's why that should worry you," https://www.washingtonpost.com/news/the-intersect/wp/2015/07/06/googles-algorithm-shows-prestigious-job-ads-to-men-but-not-to-women-heres-why-that-should-worry-you/, 2015, accessed: 2024-08-31.

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

[52] 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.

[53] 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.

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

[55] 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.

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

[57] 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.

[58] 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.