Chandigarh College of Engineering and Technology
(Degree Wing)

Neural Networks
Final Project File
Artificial Intelligence Fairness in Entity Matching

Submitted By
Dipesh Singla
Roll: - CO19322
C.S.E 6th Semester

Submitted To
Dr. Varun Gupta
Professor CSE, HOD (A.S)

Artificial Intelligence Fairness in Entity Matching

Introduction

The ideas in this paper are attempting to connect Entity Matching, AI, and Fairness. We will concentrate on determining fairness in the Entity Matching issue in the following sections. First, we will propose novel approaches to integrate fairness in entity resolution using a cutting-edge baseline. We believe that this is the first research on this topic. We will define fairness and entity matching in Section 2 and provide two examples. We also want to establish the literature in both domains. Section 3 presents the background in both domains, including the purpose of the study. We conducted numerous experiments, datasets, and conclusions in Section 4. Our view is that fairness in EM can be regarded as both a challenge and a feature.

Entity Matching

The process of matching entities is known as entity matching. It is also referred to as deduplication, record linkage, entity resolution, and other phrases. is the process of locating records that all have something to do with the same physical entity. This task distinguishes between items such as and, which may refer to the same thing. Entity resolution is an important part of data integration and cleaning. Entities can be discovered from disparate data sources as well as from the same data point. Domain experts label the data after which the matching is performed to identify which records are matches and which aren't. After the labeled dataset is divided into a train and at least set, we train the matcher on the labeled dataset and then view it as a Machine Learning-based classifier.

The above-depicted technique was used for controlled coordination. It is possible to match with rules, with supervision, or without supervision. Solo and rule-based matching do not require a named dataset. Due to the vast volume of data, labeled data cannot be retrieved, and domain specialists are often unavailable. As a result, this research focuses on the unsupervised matching space. Techniques such as hierarchical clustering and other unsupervised machine-learning techniques are employed to find matches.

Fairness

Fairness concepts in Machine Learning algorithms are becoming increasingly popular in recent years. This is of the utmost significance because these algorithms are frequently utilized in decision-making and may exhibit bias toward particular communities. Fairness has begun to be incorporated into a variety of supervised algorithms, such as classification and others. And untested techniques like outlier detection and clustering.

Individual fairness and group fairness are two concepts of fairness that are present in the research that has been carried out in this area. To be fair to everyone in a group, we must group people according to sensitive characteristics like race, gender, etc., and make certain that those groups receive the same amount of output. Additionally, the outcome ought not to be correlated with the characteristics of the entities under consideration. We try to limit the number of outcomes that are different for similar things in individual fairness. In our research, group fairness for unsupervised learning is considered.

Motivation

When run on its own, the Entity Resolution pipeline typically tries to improve accuracy by increasing the number of reported matches. However, studies have shown that concentrating on "popular" entities rather than "rare" ones can lead to significant performance penalties [1]. Knowledge bases are dominated by "popular" things, whereas "rare" ones are underrepresented. Examples include small organizations, songs that aren't in music databases, socially underrepresented groups, and other "unusual" things. Because it is difficult to link entities, it may make it more difficult to resolve them. The values of some "sensitive" attributes, such as race, gender, caste, and so on, produce this distinction. Similarity measurements can potentially induce algorithmic bias in entity matching [2]. A certain measurement may not be accurate for each type of string under consideration. When comparing Chinese and American names, for example, using the same measure may not be adequate. Vanilla entity matching does not consider a diverse collection of outputs with a wide range of matches to facilitate data-driven decision-making. Furthermore, entities with certain features may not be included in the results since vanilla entity matching does not consider a diversified collection of outputs with a wide range of matches.

A site where an award show selection committee must be constituted is one example. Members of the committee must come from a mix of genders, races, and other backgrounds to ensure that award judgments are not prejudiced. Another example is the selection of shows for a certain Online Streaming Service (OTT platform). Viewers may consider moving to other platforms if the shows do not cover a wide range of genres. As evidenced by the difficulties and examples above, fairness must be considered in the Entity Matching process. We provide two examples of entity-matching use cases to illustrate how fairness in entity-matching problems can be handled.

Use Case 1

Defining groups within a particular "sensitive" attribute in this instance. For instance, suppose we have a process for matching entities composed of two Indian population tables. The term "religion" appears in both tables, and it is one of the characteristics that, in actual tasks, could be deemed sensitive. Values such as Hindu, Muslim, Sikh, Christian, and so on can be taken from religion. which would be our organization.

To ensure that our produced matches' distribution of sensitive attribute groups reflects our predefined constraints to the greatest extent possible, we emphasize fairness. This is the idea of fairness for the group.

Our predetermined restrictions may be natural:

For our produced matches to accurately reflect the values of our sensitive attributes, we must ensure that there is an equitable distribution across all groups.

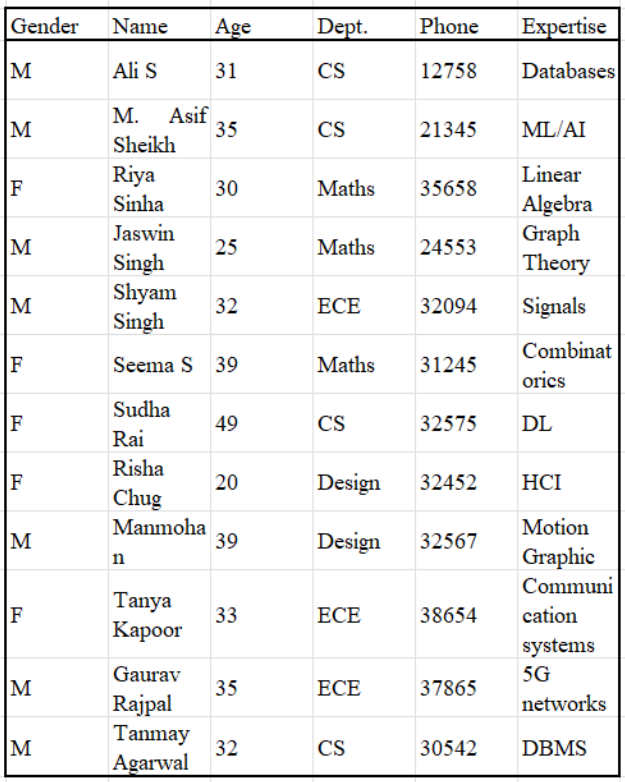
A site where an award show selection committee must be constituted is one example. Members of the committee must come from a mix of genders, races, and other backgrounds to ensure that award judgments are not prejudiced. Another example is the selection of shows for a certain Online Streaming Service (OTT platform). Viewers may consider moving to other platforms if the shows do not cover a wide range of genres. As evidenced by the difficulties and examples above, fairness must be considered in the Entity Matching process. There are two different scenarios for the entity matching problem based on a few specific examples:

Quality: to see how well our model provided relevant matches. It measures the process's quality of matching entities. To assess how well our findings were, we used the Vanilla Entity Matching setting.

Fairness: to see how well the created sensitive attribute distribution guarantees that our essential restrictions are satisfied. We use the skew metric because the number of matches should be spread identically across groups.

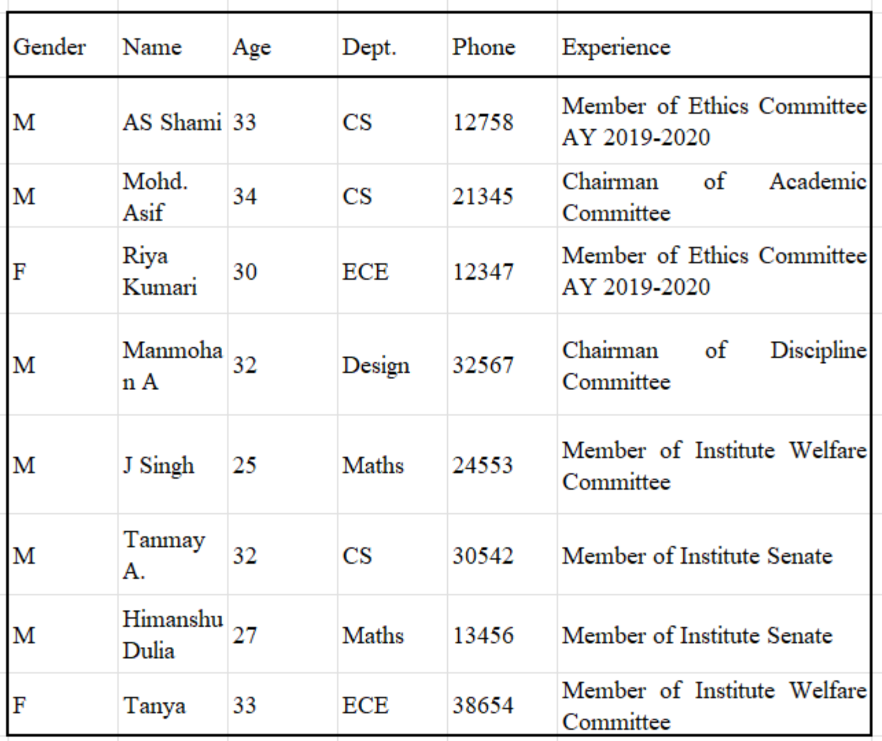
Any result that deviates from our requirement of constraints would, according to our assertion, lead to a bias in favor of or against a particular group in our model. To achieve a more equitable outcome, we would be able to complete our task of matching entities while also rewarding our model for adhering to the specified constraints. We believe that the group distribution of paired matches meets our established restrictions requirement, but at the cost of a reduction in the overall quality of our results, is an equitable sensitive group distribution of paired matches. As we strive to be more complete and include fewer coordinates with vanilla emergency rooms, our system suffers from a lack of value measures. The fairness indicator helps us since the distributions of the groups and the original dataset are more comparable.

Faculty Table (Table A)



Gender	Name	Age	Dept.	Phone	Expertise
M	Ali S	31	CS	12758	Databases
M	M. Asif Sheikh	35	CS	21345	ML/AI
F	Riya Sinha	30	Maths	35658	Linear Algebra
M	Jaswin Singh	25	Maths	24553	Graph Theory
M	Shyam Singh	32	ECE	32094	Signals
F	Seema S	39	Maths	31245	Combinatorics
F	Sudha Rai	49	CS	32575	DL
F	Risha Chug	20	Design	32452	HCI
M	Manmohan	39	Design	32567	Motion Graphic
F	Tanya Kapoor	33	ECE	38654	Communication systems
M	Gaurav Rajpal	35	ECE	37865	5G networks
M	Tanmay Agarwal	32	CS	30542	DBMS

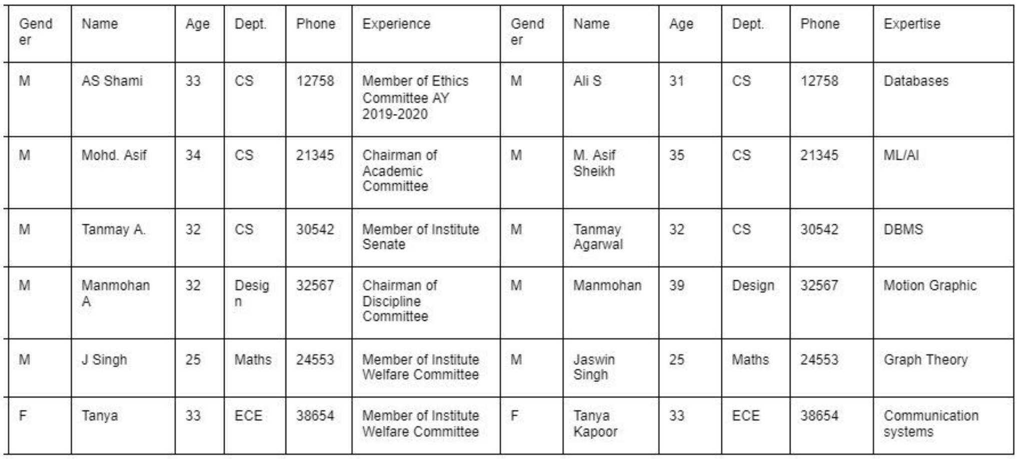
Ethics Committee Candidate Table (Table B)



Gender	Name	Age	Dept.	Phone	Experience
M	AS Shami	33	CS	12758	Member of Ethics Committee AY 2019-2020
M	Mohd. Asif	34	CS	21345	Chairman of Academic Committee
F	Riya Kumari	30	ECE	12347	Member of Ethics Committee AY 2019-2020
M	Manmohan A	32	Design	32567	Chairman of Discipline Committee
M	J Singh	25	Maths	24553	Member of Institute Welfare Committee
M	Tanmay A.	32	CS	30542	Member of Institute Senate
M	Himanshu Dulia	27	Maths	13456	Member of Institute Senate
F	Tanya	33	ECE	38654	Member of Institute Welfare Committee

Department faculty would be selected as matches significantly more frequently, according to our argument, if there was a correlation between the attributes "qualifiable" and "department". In such settings, this could lead to departmental bias.

Matched Pairs for vanilla Entity Matching -

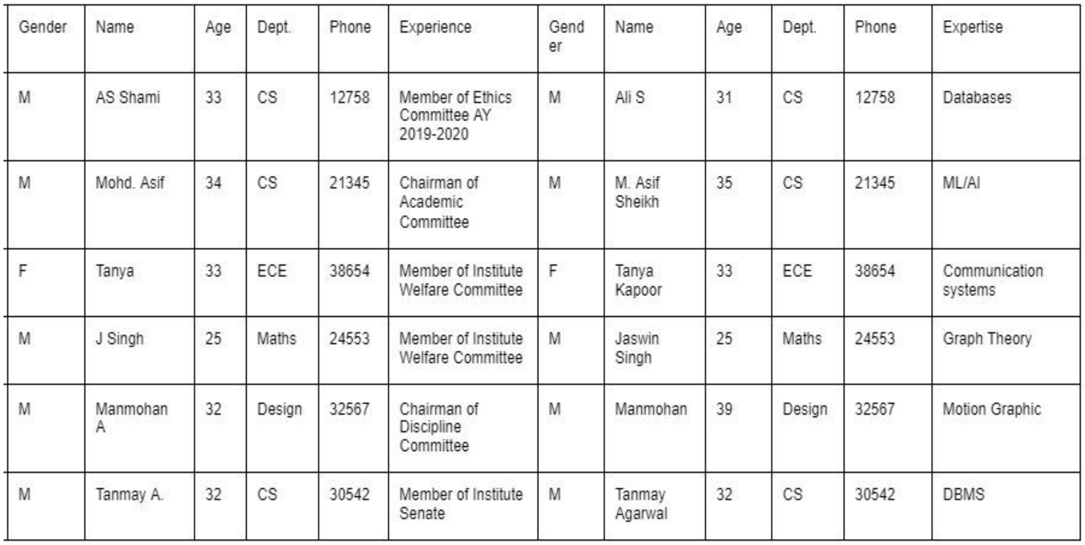


Gender	Name	Age	Dept.	Phone	Experience	Gender	Name	Age	Dept.	Phone	Expertise
M	AS Shami	33	CS	12758	Member of Ethics Committee AY 2019-2020	M	Ali S	31	CS	12758	Databases
M	Mohd. Asif	34	CS	21345	Chairman of Academic Committee	M	M. Asif Sheikh	35	CS	21345	ML/AI
M	Tanmay A.	32	CS	30542	Member of Institute Senate	M	Tanmay Agarwal	32	CS	30542	DBMS
M	Manmohan A	32	Design	32567	Chairman of Discipline Committee	M	Manmohan	39	Design	32567	Motion Graphic
M	J Singh	25	Maths	24553	Member of Institute Welfare Committee	M	Jaswin Singh	25	Maths	24553	Graph Theory
F	Tanya	33	ECE	38654	Member of Institute Welfare Committee	F	Tanya Kapoor	33	ECE	38654	Communication systems

ECE does not have a representative, but the CS department has three representatives in the top five matched entities. Design and mathematics, on the other hand, share one.
Accordingly, we characterize our requirements for decency. According to our FAIR-EM approach, we should compare the distributions of the groups in our sensitive attribute dataset to those of the groups in our initial dataset, the Faculty Table in this instance, to

achieve a successful match. In our example, we have four departments, so there should be the same proportion of successful matches across the original dataset's three groups (design, CS, and math) as (4, 3, and 2) in our approach. As a consequence, FAIR-EM has applied to the same dataset and yields the following result. CS, Math, ECE, and design appear as the top five findings instead of the original dataset (faculty table of 4,3,3,2), which is extremely similar to it (2,1,1,1).

Matched Pairs for FAIR-EM -



Gender	Name	Age	Dept.	Phone	Experience	Gender	Name	Age	Dept.	Phone	Expertise
M	AS Shami	33	CS	12758	Member of Ethics Committee AY 2019-2020	M	Ali S	31	CS	12758	Databases
M	Mohd. Asif	34	CS	21345	Chairman of Academic Committee	M	M. Asif Sheikh	35	CS	21345	ML/AI
F	Tanya	33	ECE	38654	Member of Institute Welfare Committee	F	Tanya Kapoor	33	ECE	38654	Communication systems
M	J Singh	25	Maths	24553	Member of Institute Welfare Committee	M	Jaswin Singh	25	Maths	24553	Graph Theory
M	Manmohan A	32	Design	32567	Chairman of Discipline Committee	M	Manmohan	39	Design	32567	Motion Graphic
M	Tanmay A.	32	CS	30542	Member of Institute Senate	M	Tanmay Agarwal	32	CS	30542	DBMS

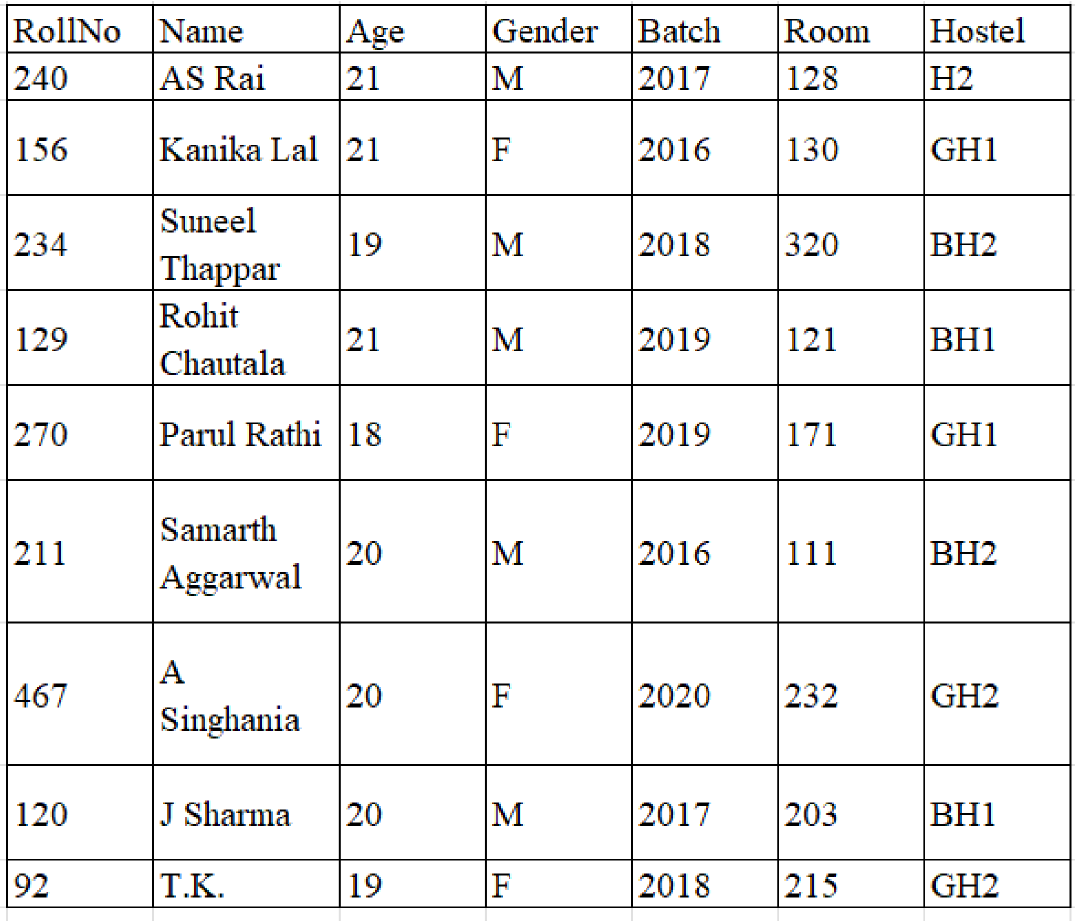
Fairness aims to distribute the privilege of serving on the ethics committee proportionately among sensitive attribute groups.

The more precise matches that we achieve by removing entity fairness constraints would be guaranteed to be fair and accurate, however, the outcomes would not be guaranteed to be fair and accurate. The faculty and applicant tables might, for example, favor one of the current groups that we are sensitive to. As a result, when we set our parameters, our model would learn to reward our constraints, producing outputs that were somewhat less precise but closer to our parameters.

Example 2

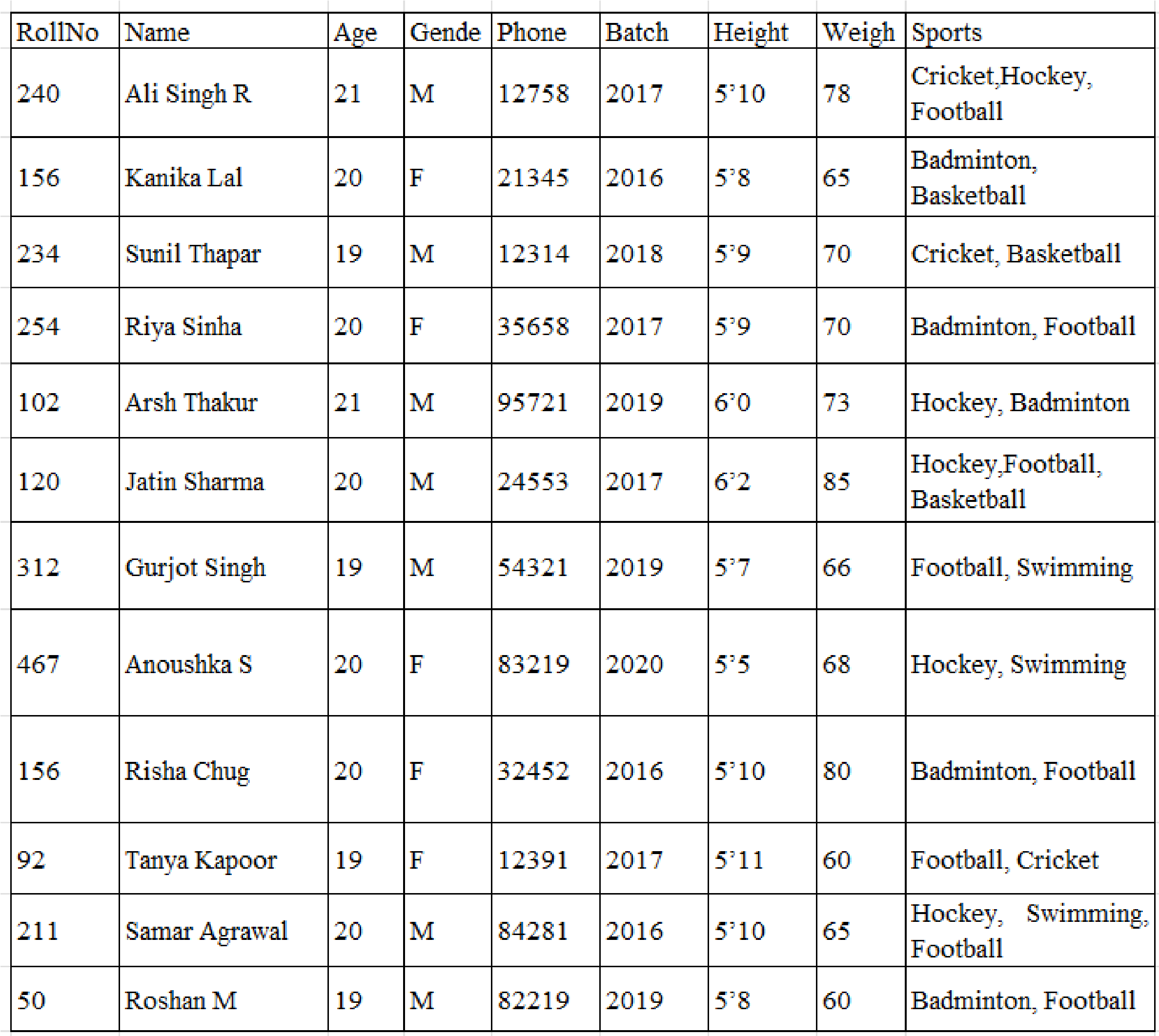
For the upcoming sports tournament, it is our responsibility to put together a college team. We use two student tables, one of which comes from the hostel coordinator's dataset and contains information about the residents (suppose Table A). The sports coordinator keeps a record of the other one's students' success in sports (suppose Table B). We have to keep the gender ratio under control because we desire equal numbers of men and women on our team.

Hostel Dataset (Table A)



RollNo	Name	Age	Gender	Batch	Room	Hostel
240	AS Rai	21	M	2017	128	H2
156	Kanika Lal	21	F	2016	130	GH1
234	Suneel Thappar	19	M	2018	320	BH2
129	Rohit Chautala	21	M	2019	121	BH1
270	Parul Rathi	18	F	2019	171	GH1
211	Samarth Aggarwal	20	M	2016	111	BH2
467	A Singhania	20	F	2020	232	GH2
120	J Sharma	20	M	2017	203	BH1
92	T.K.	19	F	2018	215	GH2

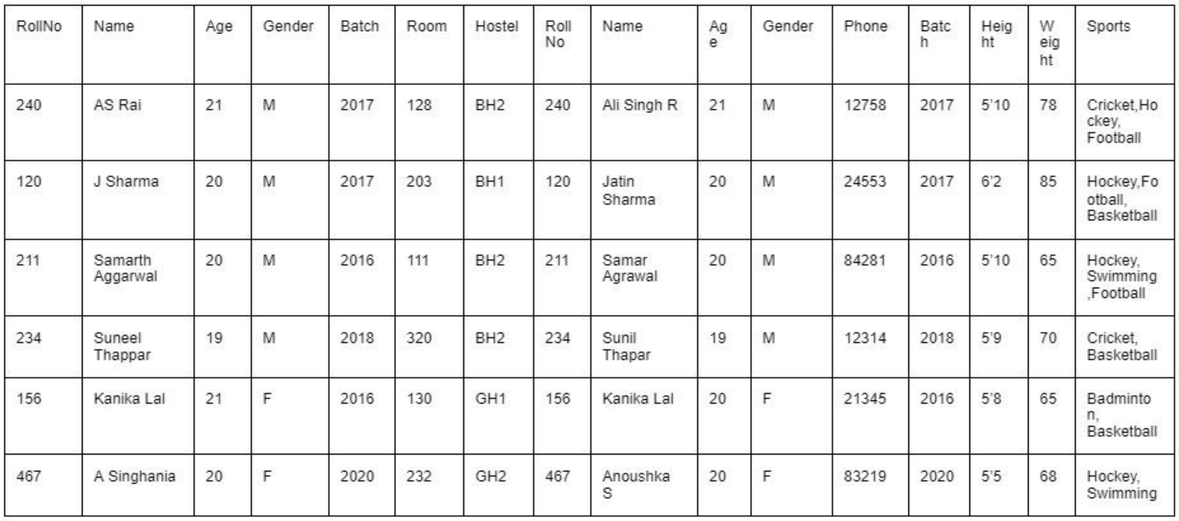
Sports Dataset (Table B)



RollNo	Name	Age	Gende	Phone	Batch	Height	Weigh	Sports
240	Ali Singh R	21	M	12758	2017	5’10	78	Cricket,Hockey, Football
156	Kanika Lal	20	F	21345	2016	5’8	65	Badminton, Basketball
234	Sunil Thapar	19	M	12314	2018	5’9	70	Cricket, Basketball
254	Riya Sinha	20	F	35658	2017	5’9	70	Badminton, Football
102	Arsh Thakur	21	M	95721	2019	6’0	73	Hockey, Badminton
120	Jatin Sharma	20	M	24553	2017	6’2	85	Hockey,Football, Basketball
312	Gurjot Singh	19	M	54321	2019	5’7	66	Football, Swimming
467	Anoushka S	20	F	83219	2020	5’5	68	Hockey, Swimming
156	Risha Chug	20	F	32452	2016	5’10	80	Badminton, Football
92	Tanya Kapoor	19	F	12391	2017	5’11	60	Football, Cricket
211	Samar Agrawal	20	M	84281	2016	5’10	65	Hockey, Swimming, Football
50	Roshan M	19	M	82219	2019	5’8	60	Badminton, Football

In the past, matching students from both databases was possible using the entity matching approach, but constraint-based fairness could not be achieved. As a result of using Machine Learning methods for the EM, our model may not provide an adequate gender proportion to be restricted afterward. We can attempt to match students from both databases in this way, but constraint-based fairness may not be achieved.

Vanilla Entity Matching Matched Pairs -

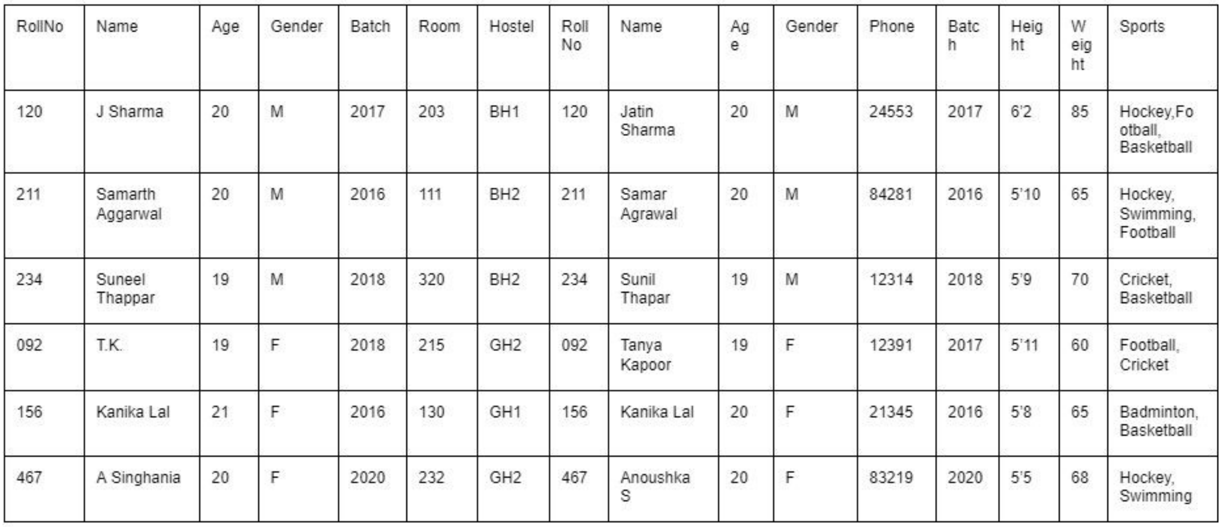


RollNo	Name	Age	Gender	Batch	Room	Hostel	Roll No	Name	Age	Gender	Phone	Batch	Height	Weight	Sports
240	AS Rai	21	M	2017	128	BH2	240	Ali Singh R	21	M	12758	2017	5'10	78	Cricket,Hockey, Football
120	J Sharma	20	M	2017	203	BH1	120	Jatin Sharma	20	M	24553	2017	6'2	85	Hockey,Football, Basketball
211	Samarth Aggarwal	20	M	2016	111	BH2	211	Samar Agrawal	20	M	84281	2016	5'10	65	Hockey, Swimming ,Football
234	Suneel Thappar	19	M	2018	320	BH2	234	Sunil Thapar	19	M	12314	2018	5'9	70	Cricket, Basketball
156	Kanika Lal	21	F	2016	130	GH1	156	Kanika Lal	20	F	21345	2016	5'8	65	Badminton, Basketball
467	A Singhania	20	F	2020	232	GH2	467	Anoushka S	20	F	83219	2020	5'5	68	Hockey, Swimming

Three ladies and five gentlemen were chosen from the nine records. The non-matching record (named T.K. or Tanya Kapoor) was not chosen because our matcher would not be able to recognize it due to its large name difference. As a result, gender distribution was uneven. The top six results are shown.

Consequently, in this section, we present our Fair Entity Matcher. As a consequence of this, we can impose constraints on our Entity Matching task, such as making certain that the sensitive attribute gender contains gender-equal representations.

Matched Pairs for FAIR-EM -



RollNo	Name	Age	Gender	Batch	Room	Hostel	Roll No	Name	Age	Gender	Phone	Batch	Height	Weight	Sports
120	J Sharma	20	M	2017	203	BH1	120	Jatin Sharma	20	M	24553	2017	6'2	85	Hockey,Football, Basketball
211	Samarth Aggarwal	20	M	2016	111	BH2	211	Samar Agrawal	20	M	84281	2016	5'10	65	Hockey, Swimming, Football
234	Suneel Thappar	19	M	2018	320	BH2	234	Sunil Thapar	19	M	12314	2018	5'9	70	Cricket, Basketball
092	T.K.	19	F	2018	215	GH2	092	Tanya Kapoor	19	F	12391	2017	5'11	60	Football, Cricket
156	Kanika Lal	21	F	2016	130	GH1	156	Kanika Lal	20	F	21345	2016	5'8	65	Badminton, Basketball
467	A Singhania	20	F	2020	232	GH2	467	Anoushka S	20	F	83219	2020	5'5	68	Hockey, Swimming

The gender ratio is three in this instance. One male result has been removed, and one female result has been added. Consequently, our model would generate the matches while adhering to our constraints to ensure their overall fairness.

Use Case 2: Using this approach, we create "blocks" in our input dataset by exploiting predetermined groups from a restricted characteristic. For example, ethnicities are a restricted property with five unique values: Hispanic, Asian, Caucasian, and so on. As a result, there will be five distinct blocks. According to a domain expert, the ground truth should apply to the match results from one of the blocks. We will learn the weights for the characteristics specified by our matcher from that block. Those set weights will be the foundation for matching the remaining blocks. We can claim that the match was fair because the weights were assigned to a fair group according to the expert.

A "fair" group is defined as one that matches the specified loads at the same level as a "fair" block, whereas a "fair" group is defined as one that matches the specified loads at a level equal to or lower than the level of the "fair" block. In contrast, a group that matches the specified loads at a level equal to or higher than the level of the "fair" block is regarded as being "fair" in comparison to a group that matches the specified loads at a level equal to or lower than the level of the "fair" block. The matching produced from the fair block's loads would be fair by definition if our basic assumptions were correct. However, we are not confident about its quality. Consequently, we employ measures to assess the tradeoff between justice and quality. Precision, recall, and other approaches may be used to assess match quality, which would indicate fairness.

A significant bias would be indicated in this case, because a bias in favor of one group would be observed. If we want to determine the blocks randomly or all groups generated by a sensitive trait, we can do so or we can include all groups generated by a particular characteristic. The f-score and other quality indicators are missed out on as we seek to be more inclusive and include fewer matches than vanilla-ER. The fairness indicator helps us since the distributions of the groups and the original dataset are more comparable. In use case 1, the metrics are the same.

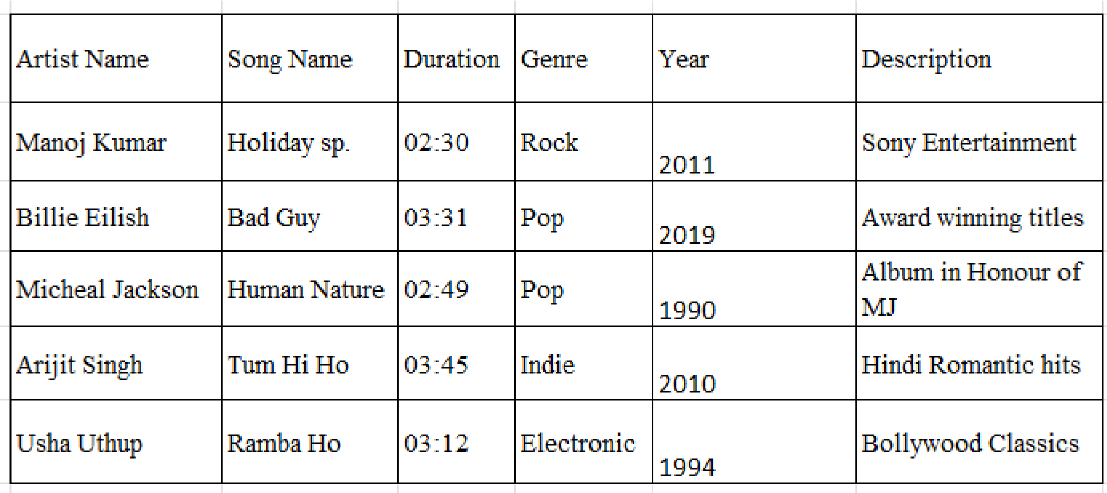
Use Case 1 is used as a basis for our proposed approach.

2.2.1 Motivating Example

Consider a scenario in which we must select songs for a melody program such as Spotify. People will go to other applications if the final music selection is not diversified in the genre. Two datasets are displayed, with attributes such as Melody name, Craftsman name, Term, and so on. Based on the non-sensitive nature of this record, the link would look like these records.

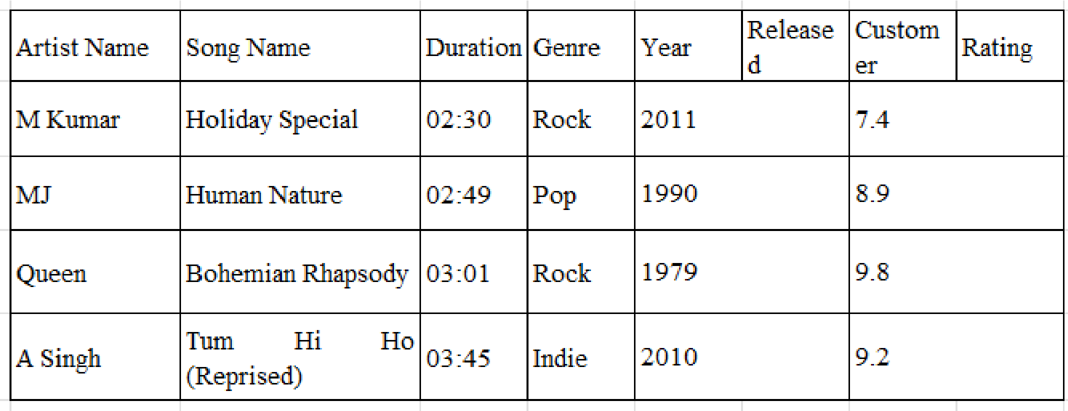
We started by defining the blocks based on the sensitive property "Genre" to establish fairness. Domain experts have advised us that the Pop group's findings will be fair owing to their large number and high possibility of being picked for the application.

Two tables as input -
Table A: Songs Dataset



Artist Name	Song Name	Duration	Genre	Year	Description
Manoj Kumar	Holiday sp.	02:30	Rock	2011	Sony Entertainment
Billie Eilish	Bad Guy	03:31	Pop	2019	Award winning titles
Micheal Jackson	Human Nature	02:49	Pop	1990	Album in Honour of MJ
Arijit Singh	Tum Hi Ho	03:45	Indie	2010	Hindi Romantic hits
Usha Uthup	Ramba Ho	03:12	Electronic	1994	Bollywood Classics

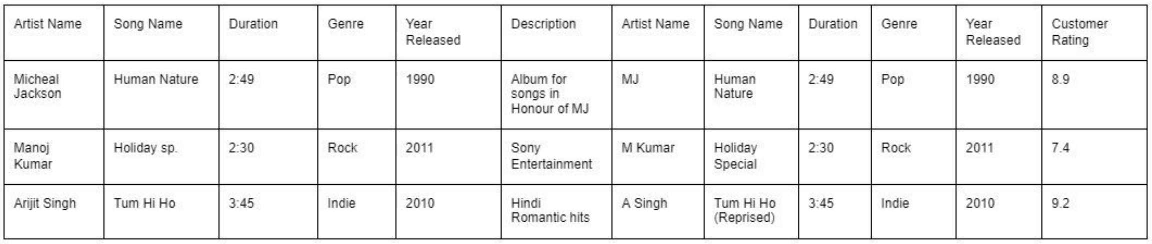
Table B: Songs to be selected



Artist Name	Song Name	Duration	Genre	Year	Released	Customer	Rating
M Kumar	Holiday Special	02:30	Rock	2011		7.4	
MJ	Human Nature	02:49	Pop	1990		8.9	
Queen	Bohemian Rhapsody	03:01	Rock	1979		9.8	
A Singh	Tum Hi Ho (Reprised)	03:45	Indie	2010		9.2	

Results for this Fair-EM-

First, the Pop group would be looked for matches. Matches in the other blocks are found after the weights for that block are utilized. In this instance, the weights learned from the fair block would determine our final matches. This is a possible outcome that can be demonstrated using this strategy.



Artist Name	Song Name	Duration	Genre	Year Released	Description	Artist Name	Song Name	Duration	Genre	Year Released	Customer Rating
Micheal Jackson	Human Nature	2:49	Pop	1990	Album for songs in Honour of MJ	MJ	Human Nature	2:49	Pop	1990	8.9
Manoj Kumar	Holiday sp.	2:30	Rock	2011	Sony Entertainment	M Kumar	Holiday Special	2:30	Rock	2011	7.4
Arijit Singh	Tum Hi Ho	3:45	Indie	2010	Hindi Romantic hits	A Singh	Tum Hi Ho (Reprised)	3:45	Indie	2010	9.2

In this case, the output may contain two or three matches. Precision, recall, and other approaches can be used to assess the match's quality. In comparison to Vanilla EM, the number of matches here is lower to accommodate the fair group's results.

Literature Review: Entity Matching

Numerous research on entity matching has been undertaken. To efficiently capture similarities between tuples, a complex composition mechanism is employed to transform each tuple into a distributed representation (a vector). SVM and clustering-based matching are two other methods that make use of machine learning, as is the case with [4] and [3]. Metrics for entity matching can be found in [5]. In opposition to our principal technique, imperative-based element matching is a subfield that can be valuable for acquiring significant bits of knowledge. Both weighted constraint-based matching and constraints-based matching are mentioned in [6] and [7].

Unsupervised matching has been used with a variety of clustering methods, including hierarchical agglomerative clustering [8].In another method [9], the App join algorithm is utilized to locate records whose similarity exceeds a predetermined threshold. The Disjoint set algorithm is then used to cluster the entities. In the present level of unsupervised learning, a modified version of the clustering optimization function is utilized, and each attribute in the EM pipeline is weighted [10]. This business has also produced graphic models as well as methodologies such as k-means clustering [12], genetic programming [13], and others [11]. We use [14] as the current standard for unsupervised entity matching for our needs. This serves as the foundation for our goal of decency. Fairness In terms of fairness, several publications [15, 16] and [17] have been used in both supervised and unsupervised learning.

$$RLS_{\mathcal{W}}(r_x, r_y) = \sum_{a_j \in \mathcal{A}} \sum_{S \in \mathcal{S}_j} w_{jS}^2 \times S(r_x.a_j, r_y.a_j)$$

Regarding an ML-based pipeline from beginning to end, fairness is discussed in [18]. For the name-matching algorithm, fairness about entity matching is addressed in part in [19]. For our current research focus, this is a very useful reference. The most recent unsupervised record linkage method is used by us [14]. A record linkage score for a record pair (rx, ry) that represents the chance that the two records pertain to the same entity is obtained by combining the pair's rx and ry records' linkages scores. A higher score increases the likelihood of a match. Equation 1 shows the RLS score formula in detail:

The weight wj of the attribute aj A corresponds to the similarity measure in Sj, and the record pair is (rx, ry). W is the collective designation for the collection of weights that are set to 1/ (the total number of attributes).

$$AMB_{\mathcal{W}}(\mathcal{R}) = \sum_{\{r_x, r_y \in \mathcal{R}, x \neq y\}} min\{RLS_{\mathcal{W}}(r_x, r_y), \tau - RLS_{\mathcal{W}}(r_x, r_y)\}$$

A fairer approach is employed to maximize the ambiguity loss. For any pair of records in the dataset, we must score the likelihood of a match as close to tau (the maximum value for a record to be a match) or as close to 0 (a non-match) as possible. This is demonstrated by: By using a gradient descent technique, the derivative is calculated. The linked weight update condition is used to update each weight in each cycle. The pace of learning is referred to as Nu.

$$w_{j'S'} = w_{j'S'} - \mu \times \frac{\partial AMB_{\mathcal{W}}(\mathcal{R})}{\partial w_{j'S'}}$$

In the parts that follow, we will first go through the assessment measures that we employ to establish fairness. The two datasets that we utilized for our review are then displayed. Finally, we describe our recommended strategy as well as a fresh approach to our inquiry.

Evaluation metrics

To assess fairness-aspect, the logarithmic ratio of the proportion of candidates with group g among the top k ranked results to the required proportion of group ai for a group ai is used. "skew@k" refers to the logarithmic ratio of the proportion of candidates with group g among the top k ranked results to the required proportion of group ai for a group ai.

$$Skew_{a_i}@k(\tau_r) = \log_e\left(\frac{p_{\tau_r^k, r, a_i}}{p_{q, r, a_i}}\right)$$

The distribution of the final ranking group (ai) is indicated in the numerator, while the intended distribution is shown in the denominator. Group Ai is distributed in the first dataset in our example. As a consequence, the numerator is the number of records of gathering artificial intelligence in the top k outcomes/k, and the denominator is the number of recordings of gathering artificial intelligence in the whole dataset or the size of the dataset.

We wish to aggregate the slant values for every feasible group in order to see the values for all possible slopes. This is accomplished through the use of two methods: To begin, we split the lowest skew by the highest skew across all categories. This highlights the disparity between a group's greatest advantage and greatest disadvantage. In this case, the group with the lowest skew value is at a disadvantage. The one with the highest skew value has the most benefit or representation.

To create a more realistic depiction, we use the average of all groups' absolute skew values, because the gap between the biggest skew and the smallest skew would only indicate the group with the greatest disadvantage.

Datasets considered

There are several benchmark datasets available for Entity Matching analysis. Some of them are as follows: (the name includes a link)-

Cora : Citation dataset including information from the RIDDLE data repository

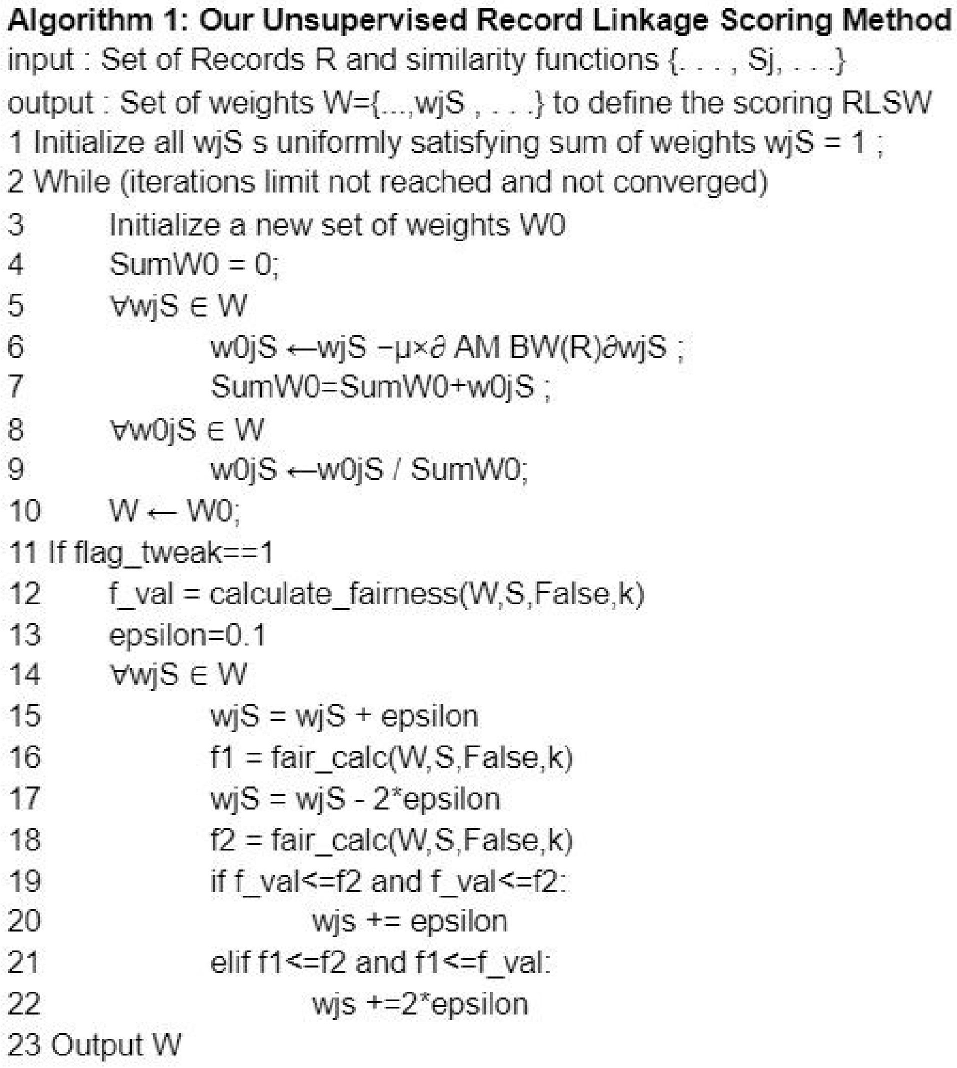
Wiki-links dataset: Dataset for disambiguating Wikipedia references

Rexa v0.1: A collection of author records derived from bibliographies and scholarly references.

According to preliminary research, these datasets were not suitable for introducing fairness into the Entity Matching method. We couldn't argue that their characteristics were sensitive. Consequently, we looked at datasets from Anhai D's collection. There, we found datasets with two unmistakable information sources that should have been matched so we could undoubtedly carry out the EM pipeline. After going through a number of datasets, we completed two datasets with delicate characteristics that we can consider. Both the Music Dataset and the Restaurant Dataset were completed. We select the restaurant's menu as the sensitive attribute in the Restaurant dataset. In both tables A and B, the attributes of the two datasets are nearly identical. It's possible that different cuisines will be given different treatment in the future. Additionally, the dataset includes a wide range of cuisines and quantities for each. There are 533 and 331 entries, respectively, in Tables A and B. 112 connections are found. Because different representations of this attribute may yield different results, we estimate that the attribute 'Genre' is the most sensitive in the Music dataset when compared to other important features such as Song Name and ArtistName. A sample size of 539 was provided for assessment purposes in the music dataset. Table A has 6906 records, and Table B has 55923 entries.

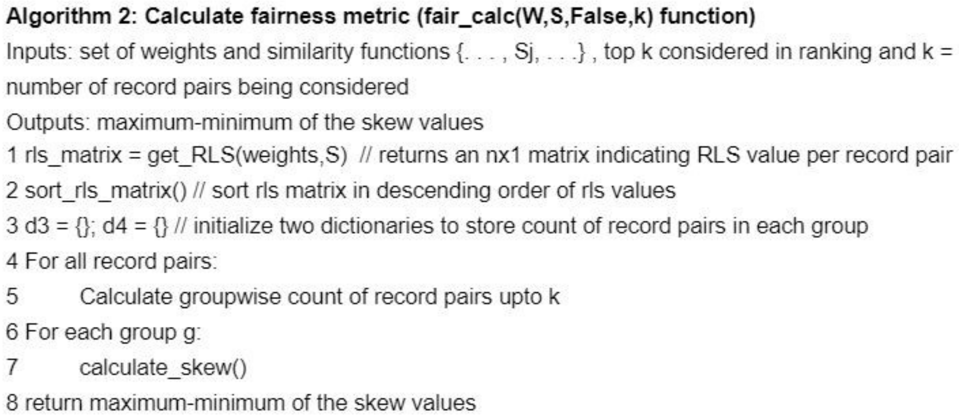
Weight Tweaking

To boost reasonableness while keeping a really constant distribution, we integrate weight tweaking in our unique element matching approach. We add a weight-tweaking technique to each iteration of learning weights for entity matching after fairness-independent steps:
We go through each weight separately. We contend that the original w, w + epsilon, and w-epsilon are all conceivable. After determining the fairness-skew-value for each of the three scenarios, the new w is the most compelling case for providing fairness.
The pseudocode for algorithm 1 is listed below. It is a modified version of algorithm 1 from [14]. We increased the weights of our entity matching algorithm to evaluate its fairness. We evaluated the fairness of our entity matching algorithm by moving to the next weight with w new . We hypothesized that increasing the weights would improve the fairness of our entity matching without compromising its quality. The two new algorithms are listed below. Algorithm 1 is a modified version of algorithm 1 from [14]. We evaluated the fairness of our entity matching algorithm by increasing the weights and thereby assigning the same weight to all entities. The fairness-skew meter calculation technique is depicted in Algorithm 2.



Algorithm 1: Our Unsupervised Record Linkage Scoring Method

```
input : Set of Records R and similarity functions {. . . , Sj, . . .}
output : Set of weights W={...,wjS , . . .} to define the scoring RLSW
Initialize all wjS s uniformly satisfying sum of weights wjS = 1 ;
While (iterations limit not reached and not converged)
        Initialize a new set of weights W0
        SumW0 = 0;
        ∀wjS ∈ W
                w0jS ←wjS −µ×∂ AM BW(R)∂wjS ;
                SumW0=SumW0+w0jS ;
        ∀w0jS ∈ W
                w0jS ←w0jS / SumW0;
        W ← W0;
If flag_tweak==1
        f_val = calculate_fairness(W,S,False,k)
        epsilon=0.1
        ∀wjS ∈ W
                wjS = wjS + epsilon
                f1 = fair_calc(W,S,False,k)
                wjS = wjS - 2*epsilon
                f2 = fair_calc(W,S,False,k)
                if f_val<=f2 and f_val<=f2:
                        wjs += epsilon
                elif f1<=f2 and f1<=f_val:
                        wjs +=2*epsilon
Output W
```



Algorithm 2: Calculate fairness metric (fair_calc(W,S,False,k) function)

```
Inputs: set of weights and similarity functions {. . . , Sj, . . .} , top k considered in ranking and k =
number of record pairs being considered
Outputs: maximum-minimum of the skew values
rls_matrix = get_RLS(weights,S)  // returns an nx1 matrix indicating RLS value per record pair
sort_rls_matrix() // sort rls matrix in descending order of rls values
d3 = {}; d4 = {} // initialize two dictionaries to store count of record pairs in each group
For all record pairs:
        Calculate groupwise count of record pairs upto k
For each group g:
        calculate_skew()
return maximum-minimum of the skew values
```

Formulation of the Loss Function

In previous machine learning approaches, we assumed that the unfairness constraints were taken into account. We modify our actions so that we can exercise our good sense. The entity matching task's unappealing capability limits our obligations to be caring. We define the ambiguity minimization loss function, including a fairness penalty, by applying fairness to the ambiguity minimization loss. The loss function is formulated. It now has the FW (R) sign for fairness.

$$LW\ (R) = AMBW\ (R) + \lambda * FW\ (R)$$

A weighting border is used to assign a weight to the two components so that they are equally strong. We must retain a legal balance between the two components even if the distribution of the top k-matched entities is unequal. We must add weights in the distribution formulation in order to take the derivative in reference to the weights when optimizing. As a result, we employ RLS values to express group distribution. The RLS value is calculated using the aforementioned equation (1).

When all weights in the dataset are w=1/n, the initial distribution of the groups is w=1/n, and this pi depicts the distribution when weights are equal to W. The frequency of group I divided by the total number of record pairings in the initial dataset gives the distribution, fi, in which fi is the number of groups I record divided by the number of record pairings in the initial dataset. We go ahead in two ways after obtaining the initial and final distribution formulation:

Ratio of pi/fi

We wish to get as near to 1 as feasible in order to evaluate the difference between the initial and final distributions, in this case the ratio pi/fi. We wish to minimize the square of (1-pi/fi) so that (1-pi/fi)2 is as small as possible. We wish to total up all of these values for all groups. A smaller pi/fi number indicates a greater difference between the two distributions, whereas a value near to 0 implies the inverse. We also reduce the squared term (1 pi/fi)2 to ensure that it is as positive as possible (1 pi/fi).

Together with the Entity Matching loss, we add this to the original loss function. Taking this term's derivative during optimization would be necessary for gradient descent.

Experiments And Dataset

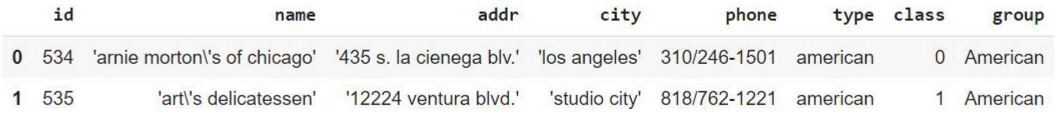
Experiment Details

The two datasets are blocked. Word 2-grams are used to block passages. First, the two datasets are blocked. This is accomplished through the usage of word 2 grams. Any two-gram of the two entities that match is a potential pair for the entity matching method. Using the blocking technique, this feature table displays many similarity measures for various qualities. When our two tables had three characteristics A1, A2, and A3, our feature table would be just A1, lev A1, jacc A2, lev A2, jacc A3, and lev A3, with jacc and lev signifying Jacobian similarity and Levenshtein distance, respectively. In this example, we used two similarity metrics: Levand Levenshtein distance. However, we used four methods in our experiments: the cosine, the jaccard, the Lowenstein Similarity, and the Monge-

Elkan similarity.

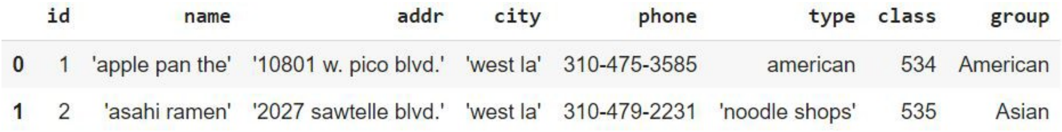
For each of the similarity measures in our feature table, we compute and store the similarity values for each record pair in our feature matrix. We will use this feature matrix for all of our subsequent calculations. After that, we begin the process of fairness-integrated entity matching and weigh each similarity measure. For our experiments, we use the weighting parameter lambda at values of 0,500, 1000, and 5000 to identify the top k matches for which k = 250, 500, 750, and 1000.

Details About the Restaurant Dataset The same attributes—[id], name, address, city, phone number, type, class, and group—were found in both Table A and Table B. We will consider "bunch" to be our delicate quality that best mirrors the eatery's menu. The first and second rows of Table A are shown below.



	id	name	addr	city	phone	type	class	group
0	534	'arnie morton\'s of chicago'	'435 s. la cienega blv.'	'los angeles'	310/246-1501	american	0	American
1	535	'art\'s delicatessen'	'12224 ventura blvd.'	'studio city'	818/762-1221	american	1	American

Table B

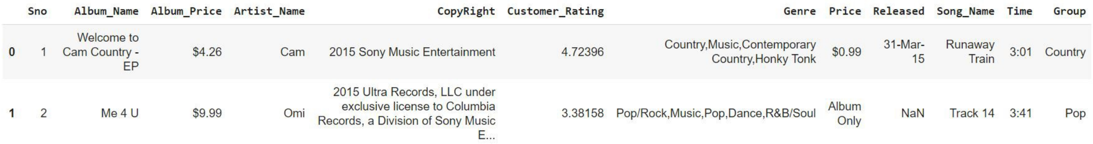


	id	name	addr	city	phone	type	class	group
0	1	'apple pan the'	'10801 w. pico blvd.'	'west la'	310-475-3585	american	534	American
1	2	'asahi ramen'	'2027 sawtelle blvd.'	'west la'	310-479-2231	'noodle shops'	535	Asian

In the first place, impeding is finished for the two datasets It is done utilizing word 2-grams. A candidate pair for the entity matching procedure is any two-gram of the two entities that match. When building the feature table, we compute and save the similarity values for each record pair for each of the similarity measures in our feature table. This feature matrix will be used in all following computations. After that, we begin the process of fairness-integrated entity matching and weigh each similarity measure. For our experiments, we use the weighting parameter lambda at values of 0,500, 1000, and 5000 to identify the top k matches for which k = 250, 500, 750, and 1000.

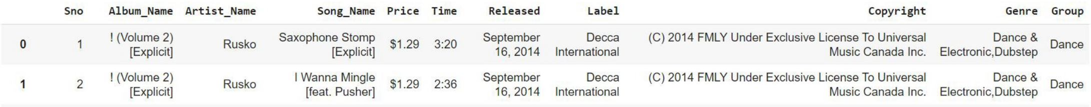
Details About the Restaurant Dataset The same attributes—[id], name, address, city, phone number, type, class, and group—were found in both Table A and Table B. We will consider "bunch" to be our delicate quality that best mirrors the eatery's menu. The first and second rows of Table A are shown below.

Table A



	Sno	Album_Name	Album_Price	Artist_Name	CopyRight	Customer_Rating	Genre	Price	Released	Song_Name	Time	Group
0	1	Welcome to Cam Country - EP	$4.26	Cam	2015 Sony Music Entertainment	4.72396	Country,Music,Contemporary Country,Honky Tonk	$0.99	31-Mar-15	Runaway Train	3:01	Country
1	2	Me 4 U	$9.99	Omi	2015 Ultra Records, LLC under exclusive license to Columbia Records, a Division of Sony Music E...	3.38158	Pop/Rock,Music,Pop,Dance,R&B/Soul	Album Only	NaN	Track 14	3:41	Pop

Table B



	Sno	Album_Name	Artist_Name	Song_Name	Price	Time	Released	Label	Copyright	Genre	Group
0	1	! (Volume 2) [Explicit]	Rusko	Saxophone Stomp [Explicit]	$1.29	3:20	September 16, 2014	Decca International	(C) 2014 FMLY Under Exclusive License To Universal Music Canada Inc.	Dance & Electronic,Dubstep	Dance
1	2	! (Volume 2) [Explicit]	Rusko	I Wanna Mingle [feat. Pusher]	$1.29	2:36	September 16, 2014	Decca International	(C) 2014 FMLY Under Exclusive License To Universal Music Canada Inc.	Dance & Electronic,Dubstep	Dance

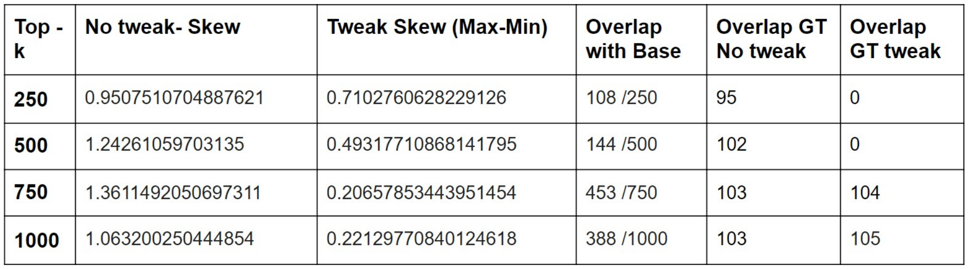
In Tables A, which had 6907 data records and 55923 records, a potential candidate set of 38,62,60,161 was generated. The candidate pool was reduced to 63,246 because of "Song Name", "Album Name", and "Copyright". "Album Name" was the first blocking attribute. In the last stage, the candidate set was reduced to 12,829 pair sets, which was the case with the attribute "Copyright".

We have added seven new matrix properties to our album metadata: "Album Name," "Artist Name," "Time," "Genre," "Released," "Copyright," and "Song Name." The seven new values are identical to the four we already have, but we now have four proximity measures to determine their similarity: Cosine, Jaccard, Levenshtein Likeness, and Monge-Elkan comparability. Our feature matrix has 28 rows, one for each similarity measure and attribute. We also have 12829 x 28 features.

In addition, we possessed a Ground Truth file with 539 sets of data, 132 of which were positive matches and 407 of which were not.

Results We begin by applying the weight-tuning algorithm. The outcomes demonstrate that our strategy of altering the weights to raise the fairness metrics at the expense of a small reduction in the Entity Matching outcomes is correct. We modify the loss function in accordance with the weight-tuning mechanism, as demonstrated in the Methodology Section. In this article, we present our observations and the outcomes of those changes for various lambda values.

Weight Tweaking

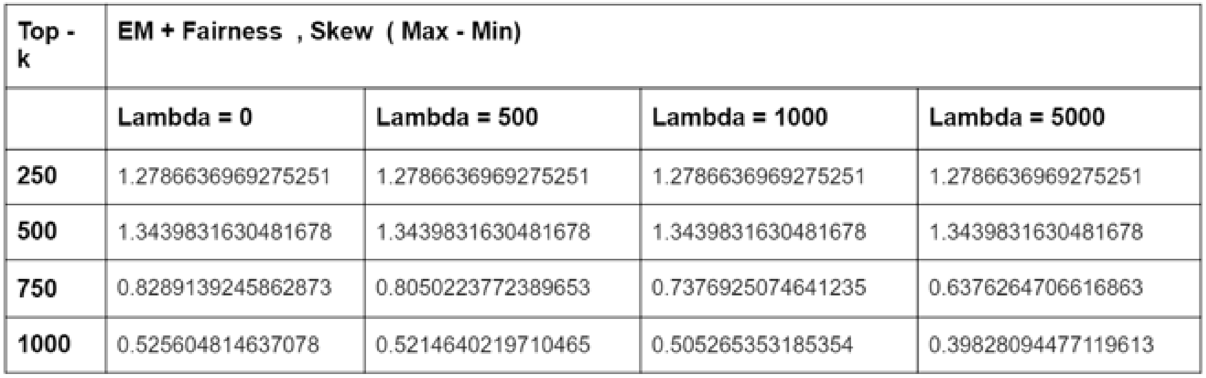


Top - k	No tweak- Skew	Tweak Skew (Max-Min)	Overlap with Base	Overlap GT No tweak	Overlap GT tweak
250	0.9507510704887621	0.7102760628229126	108 /250	95	0
500	1.24261059703135	0.49317710868141795	144 /500	102	0
750	1.3611492050697311	0.20657853443951454	453 /750	103	104
1000	1.063200250444854	0.22129770840124618	388 /1000	103	105

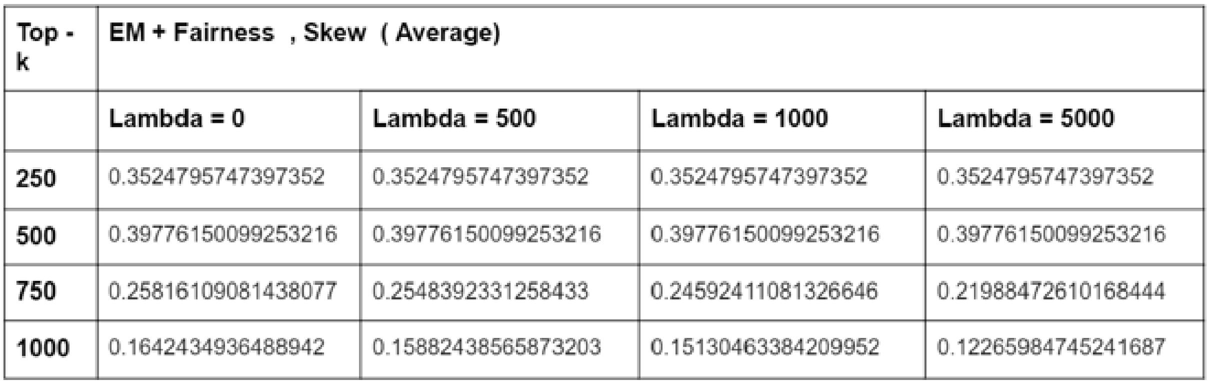
The weight-tuning calculation was intended to show that, in return for forfeiting result quality, further developing fairness is conceivable. The expected lines are shown in the results. Because it is dependent on k, the algorithm works for the top-k values in each iteration. Adjusting the weights leads in a better fairness outcome or a lower skew value for all k values. In addition, we see less overlap with Base, the conventional baseline EM method. We also find that when lower k values are considered, there is less overlap with the ground truth, but this shifts when bigger k values are considered. This is not an issue because the impromptu way of modifying one's weight just provides proof that changing one's weight may produce the desired results.

Ratio of pi/fi

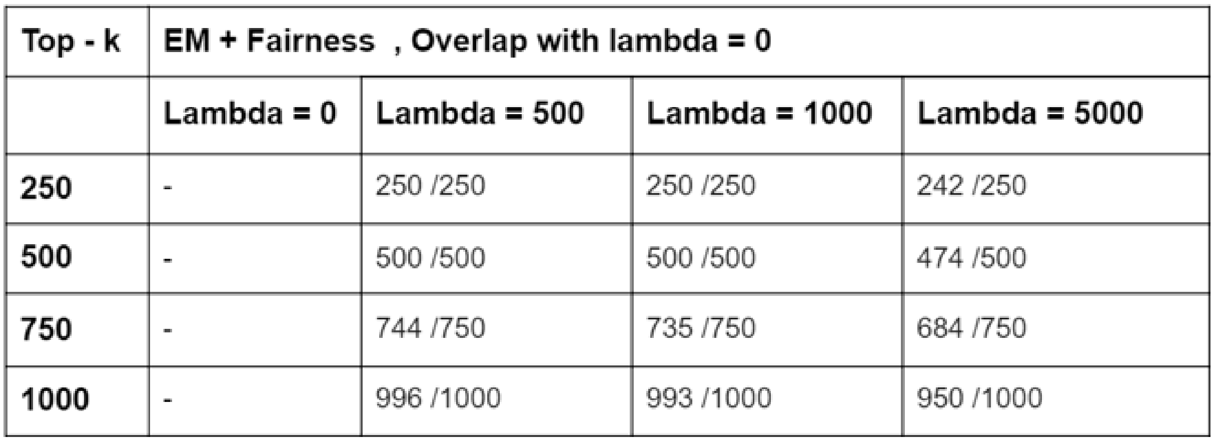
In our main method, the loss function is adjusted by multiplying a lambda-valued fairness term based on the ratio of pi to fi. Because the method is not reliant on k, the skew diminishes and the importance of fairness in our learning process grows as the value of lambda increases. We propose that because skew is inversely proportional to fairness, overall fairness improves. This pattern appears often for both the average skew and the maximum and minimum skew discrepancies.



Top - k	EM + Fairness , Skew (Max - Min)			
	Lambda = 0	Lambda = 500	Lambda = 1000	Lambda = 5000
250	1.2786636969275251	1.2786636969275251	1.2786636969275251	1.2786636969275251
500	1.3439831630481678	1.3439831630481678	1.3439831630481678	1.3439831630481678
750	0.8289139245862873	0.8050223772389653	0.7376925074641235	0.6376264706616863
1000	0.525604814637078	0.5214640219710465	0.505265353185354	0.39828094477119613



Top - k	EM + Fairness , Skew (Average)			
	Lambda = 0	Lambda = 500	Lambda = 1000	Lambda = 5000
250	0.3524795747397352	0.3524795747397352	0.3524795747397352	0.3524795747397352
500	0.39776150099253216	0.39776150099253216	0.39776150099253216	0.39776150099253216
750	0.25816109081438077	0.2548392331258433	0.24592411081326646	0.21988472610168444
1000	0.1642434936488942	0.15882438565873203	0.15130463384209952	0.12265984745241687

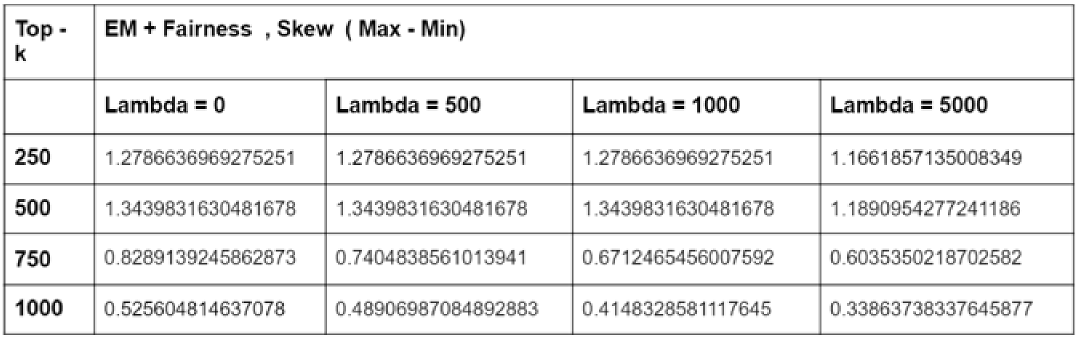


Top - k	EM + Fairness , Overlap with lambda = 0			
	Lambda = 0	Lambda = 500	Lambda = 1000	Lambda = 5000
250	-	250 /250	250 /250	242 /250
500	-	500 /500	500 /500	474 /500
750	-	744 /750	735 /750	684 /750
1000	-	996 /1000	993 /1000	950 /1000

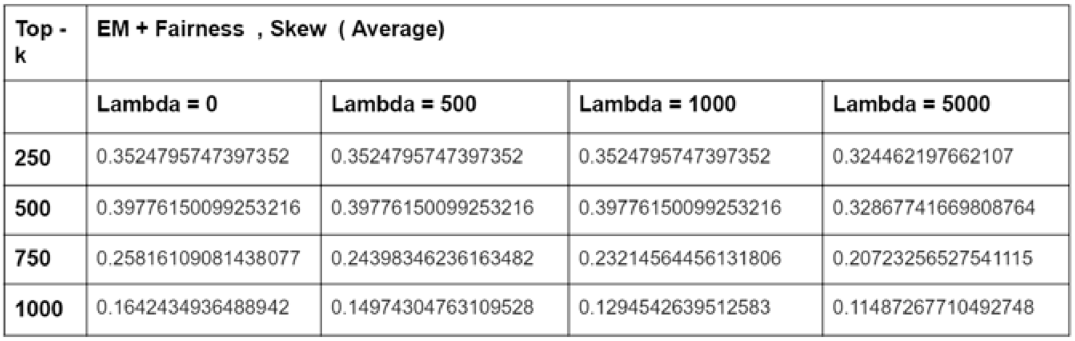
As lambda and fairness increase, the overlap with basic entity-matching results eventually reduces. With lambda as the governing hyper-parameter, this follows the exact patterns we predicted, with an increase in fairness and a fall in result quality. This pattern emerges for a variety of lambda values, with the base entity matching result lambda = 0. The same outcomes can also be obtained by altering our observational lens by using various k values.

KL Divergence With pi and fi

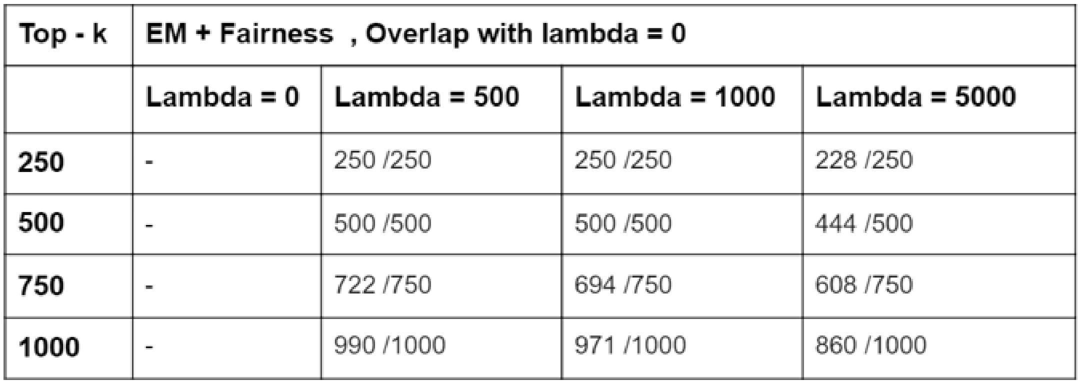
Examining our second strategy, which involves multiplying a lambda-valued fairness factor based on KL divergence with pi and fi to alter the loss function. Similarly to prior studies, we see that the importance of fairness in our learning process grows as lambda increases, whereas skew diminishes. In addition to the normal skew, the maximum and minimum skew variations follow the same pattern.



Top - k	EM + Fairness , Skew (Max - Min)			
	Lambda = 0	Lambda = 500	Lambda = 1000	Lambda = 5000
250	1.2786636969275251	1.2786636969275251	1.2786636969275251	1.1661857135008349
500	1.3439831630481678	1.3439831630481678	1.3439831630481678	1.1890954277241186
750	0.8289139245862873	0.7404838561013941	0.6712465456007592	0.6035350218702582
1000	0.525604814637078	0.48906987084892883	0.4148328581117645	0.33863738337645877



Top - k	EM + Fairness , Skew (Average)			
	Lambda = 0	Lambda = 500	Lambda = 1000	Lambda = 5000
250	0.3524795747397352	0.3524795747397352	0.3524795747397352	0.324462197662107
500	0.39776150099253216	0.39776150099253216	0.39776150099253216	0.32867741669808764
750	0.25816109081438077	0.24398346236163482	0.23214564456131806	0.20723256527541115
1000	0.1642434936488942	0.14974304763109528	0.1294542639512583	0.11487267710492748



Top - k	EM + Fairness , Overlap with lambda = 0			
	Lambda = 0	Lambda = 500	Lambda = 1000	Lambda = 5000
250	-	250 /250	250 /250	228 /250
500	-	500 /500	500 /500	444 /500
750	-	722 /750	694 /750	608 /750
1000	-	990 /1000	971 /1000	860 /1000

As lambda rises, so does the overlap with base entity matching results. When lambda is utilized as the governing hyper-parameter, a fall in result quality is accompanied by an increase in fairness. Because the basic entity corresponds to the outcome lambda = 0, this pattern emerges across a wide range of lambda values. Using other k values and changing our observational lens produces the same findings.

As a result, this demonstrates that our outcomes meet our expectations despite a different fairness term configuration.

Code:

Google Collab Link to the Project: https://colab.research.google.com/drive/1rH_r9-njaxvxDOVV-VBopS3OU3rSibxq?usp=sharing

Results:

Precision: 21.53% (73/339)
Recall: 55.3% (73/132)
F1: 31.0%
False positives: 266 (out of 339 positive predictions)
False negatives: 59 (out of 200 negative predictions)

Conclusion And Future Work

The concept of fairness was first codified in Entity Resolution. Along with a few examples that were motivating, we presented our motivation. We used a single model for an element-matching cycle in our method. Taking the most recent approach to unsupervised entity resolution as our starting point, we developed the concept of fairness.

The outcomes were generated from two datasets: restaurant data and music data. Our studies established that the new Fair-EM algorithm enhances the fairness of Entity Matching results without losing quality. As fairness importance increased, we observed that entity matching quality gradually decreased, leading to more equitable outcomes. We could make our current model more effective in our subsequent work by adjusting the hyperparameters. Another area of potential research is the fairness of various sensitive characteristics. We can also continue using other strategies (with a constraint set by the user) or even concentrate on the fairness implications of various similarity metrics.

Bibliography

1. Esquivel J, Albakour D, Martinez M, Corney D, Moussa S. On the long-tail entities in news. InEuropean Conference on Information Retrieval 2017 Apr 8 (pp. 691-697). Springer, Cham.
2. Hajian S, Bonchi F, Castillo C. Algorithmic bias: From discrimination discovery to fairness- aware data mining. InProceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining 2016 Aug 13 (pp. 2125-2126).
3. Ebraheem M, Thirumuruganathan S, Joty S, Ouzzani M, Tang N. DeepER–Deep Entity Resolution. arXiv preprint arXiv:1710.00597. 2017 Oct 2.
4. Bilenko M, Mooney RJ. Adaptive duplicate detection using learnable string similarity mea- sures. InProceedings of the ninth ACM SIGKDD international conference on Knowledge dis- covery and data mining 2003 Aug 24 (pp. 39-48).
5. Cohen WW, Richman J. Learning to match and cluster large high-dimensional data sets for data integration. InProceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining 2002 Jul 23 (pp. 475-480).
6. Gemmell J, Rubinstein BI, Chandra AK. Improving entity resolution with global constraints. arXiv preprint arXiv:1108.6016. 2011 Aug 30.
7. Shen W, Li X, Doan A. Constraint-based entity matching. InAAAI 2005 Jul 9 (pp. 862-867).
8. Shen Z, Wang Q. Entity resolution with weighted constraints. InEast European Conference on Advances in

Databases and Information Systems 2014 Sep 7 (pp. 308-322). Springer, Cham.

9. Bhattacharya I, Getoor L. Collective entity resolution in relational data. ACM Transactions on Knowledge Discovery from Data (TKDD). 2007 Mar 1;1(1):5-es.

10. Nie T, Lee WC, Shen D, Yu G, Kou Y. Distributed entity resolution based on similarity join for large-scale data clustering. InInternational Conference on Web-Age Information Management 2014 Jun 16 (pp. 138-149). Springer, Cham.

11. Steorts RC, Hall R, Fienberg SE. A Bayesian approach to graphical record linkage and deduplication. Journal of the American Statistical Association. 2016 Oct 1;111(516):1660-72.

12. Elfeky, M.G., Verykios, V.S., Elmagarmid, A.K.: TAILOR: a record linkage toolbox. In: 2002 Proceedings of 18th International Conference on Data Engineering, pp. 17–28. IEEE (2002)

13. Nikolov A, d'Aquin M, Motta E. Unsupervised learning of link discovery configuration. InExtended Semantic Web Conference 2012 May 27 (pp. 119-133). Springer, Berlin, Heidelberg.

14. Jurek A, Deepak P. It pays to be certain: unsupervised record linkage via ambiguity mini- mization. InPacific-Asia Conference on Knowledge Discovery and Data Mining 2018 Jun 3 (pp. 177-190). Springer, Cham.

15. Jurek A, Deepak P. It pays to be certain: unsupervised record linkage via ambiguity mini- mization. InPacific-Asia Conference on Knowledge Discovery and Data Mining 2018 Jun 3 (pp. 177-190). Springer, Cham.

16. Kleindessner M, Samadi S, Awasthi P, Morgenstern J. Guarantees for spectral clustering with fairness constraints. arXiv preprint arXiv:1901.08668. 2019 Jan 24.

17. Deepak P, Abraham SS. Fair Outlier Detection. InInternational Conference on Web Infor- mation Systems Engineering 2020 Oct 20 (pp. 447-462). Springer, Cham.

18. Abraham SS, Sundaram SS. Fairness in Clustering with Multiple Sensitive Attributes. arXiv preprint arXiv:1910.05113. 2019 Oct 11.

19. Shaikh S, Vishwakarma H, Mehta S, Varshney KR, Ramamurthy KN, Wei D. An end-to-end machine learning pipeline that ensures fairness policies. arXiv preprint arXiv:1710.06876. 2017 Oct 18.

20. Karakasidis A, Pitoura E. Identifying Bias in Name Matching Tasks. InEDBT 2019 (pp. 626-629).

21. Geyik SC, Ambler S, Kenthapadi K. Fairness-aware ranking in search recommendation systems with application to linkedin talent search. InProceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining 2019 Jul 25 (pp. 2221-2231).