It pays to be Certain: Unsupervised Record Linkage via Ambiguity Minimization

Anna Jurek Deepak P
Queen's University Belfast, UK
a.jurek@qub.ac.uk deepaksp@acm.org

Abstract. Record linkage (RL) is a process of identifying records that refer to the same real-world entity. Many existing approaches to RL apply supervised machine learning (ML) techniques to generate a classification model that classifies a pair of records as either linked or non-linked. In such techniques, the labeled data helps guide the choice and relative importance to similarity measures to be employed in RL. Unsupervised RL is therefore a more challenging problem since the quality of similarity measures needs to be estimated in the absence of linkage labels. In this paper we propose a novel optimization approach to unsupervised RL. We define a scoring technique which aggregates similarities between two records along all attributes and all available similarity measures using a weighted sum formulation. The core idea behind our method is embodied in an objective function representing the overall ambiguity of the scoring across a dataset. Our goal is to iteratively optimize the objective function to progressively refine estimates of the scoring weights in the direction of lesser overall ambiguity. We have evaluated our approach on multiple real world datasets which are commonly used in the RL community. Our experimental results show that our proposed approach outperforms state-of-the-art techniques, while being orders of magnitude faster.

Introduction

RL, also referred to as data matching or entity resolution, is the task of finding records that correspond to the same entity from one or more data sources. Given two data sources, each pair of records can be classified into one of two classes: linked and non-linked. Table 1 shows a simple example of RL. The table contains records from two bibliographic data sources, viz., DBLP and ACM digital library. The aim is to identify those pairs of records referring to the same publications, which in this case are (ACM1, DB1) and (ACM2, DB2). Any other pairs of records should be identified as non-linked. If records have error-free and unique identifiers, such as social security numbers, RL is a straightforward process that can be easily performed by the standard database join operation. In many practical scenarios, however, such a unique identifier does not exist and the linkage process needs to be performed by approximate matching of the corresponding fields of two records. It is also notable that the same data can be represented in different ways in different data sources due to factors such as different conventions, typographical errors, missing and out of date values. This makes similarity matching and aggregation of similarity scores to perform record linkage, a challenging task.

Table 1: An Example of RL.

ID	Title	Authors	Venue
ACM1	A compact B-tree	Peter Bumbulis, Ivan T. Bowman	International Conference on Management of Data
ACM2	A theory of redo recovery	David Lomet, Mark Tuttle	International Conference on Management of Data
DB1	A compact B-tree	Peter Bumbulis, Ivan Bowman	SIGMOD Conference
DB2	A theory of redo recovery	Mark R. Tuttle, David B. Lomet	SIGMOD Conference
DB3	The nimble integration engine	Denise Draper Alon Y. Halevy Daniel S. Weld	SIGMOD Conference

Efficiency and Effectiveness in RL: The problem of RL can be seen as comprising two main fields of research which are: (1) developing time-efficient algorithms for RL [1] and (2) efforts on developing techniques for effective link discovery [2], [3], [4]. The former focuses on improving the turnaround time for record linkage through heuristically avoiding comparison between records that hold a low apriori chance of getting linked. The space of candidate record pairs for record linkage is evidently quadratic in the size of the datasets. This quadratic space is often pruned out through indexing and filtering, collectively referred to as blocking methodologies. The latter field of research, that towards effective RL, focuses on the orthogonal problem of accurately determining which among compared candidate pairs are to be labelled as linked/non-linked. In this work, we will focus on the second research problem, which is the development of models for accurately determining the linkage status of record pairs.

Effective Record Linkage: Techniques for effective RL may be seen as comprising two major streams, based on whether labelled data is exploited for the task. The large majority of techniques for effective RL rely on the usage of a training dataset comprising record pairs that are labelled as linked/non-linked to learn a classifier, thus treating it as a supervised learning problem. In these techniques, each pair of records is represented as a similarity vector representing a set of numeric similarities, each calculated with a similarity measure on a pair of field values of the two records. The task of RL is then considered as a binary classification problem over similarity vectors [2]. The second catergory address RL in the absence of training data. Traditionally, this task has been addressed by replacing training data with assistance from a domain expert, who would handcraft bespoke domain-specific rules that aid determining the linkage likelihood of a candidate record pair [3, 4]. Drawing up rules for record linkage requires deep topical expertise in the domain, an impractical or costly proposition in many scenarios. This makes unsupervised machine learning for RL, the task that targets to tackle RL without the aid of training data or a domain expert, a promising avenue of research in RL. It is notable that unsupervised machine learning for RL is much more challenging than the supervised or expert-assisted variants; this explains the relative dearth of research.

Our Contribution: In this paper, we address the problem of unsupervised RL and propose a novel method for unsupervised scoring of record pairs modeling the task as an optimization problem. *Our core idea is that a good record linkage method would be able to make conclusive decisions on the linkage of most pairs of records, if not all; we look for methods that can achieve conclusive decisions, while staying strictly within the*

space of RL models that make linkage decisions on a weighted sum aggregate of similarity scores. Accordingly, we outline a model for the ambiguity of record linkage, and progressively refine the weightings associated with similarity measures in the direction of reducing overall ambiguity. Through an empirical analysis over multiple real-world datasets, we illustrate that our method is able to outperform existing methods.

Related Work

We briefly survey recent RL methods under two separate heads.

Semi-supervised Record Linkage. In semi-supervised learning, a small set of labeled instances and a large set of unlabeled instances are used in the training process. A popular approach to semi-supervised RL is that using active learning (AL) [5]. AL identifies highly informative instances for manual labeling that are later used for training classification models. In [6] the instances that are not assigned to the same class by majority of the classifiers are selected for manual labeling. A different approach, where a set of similarity vectors are ranked and those in the middle (ambiguous) region are selected for manual labeling, is proposed in [7]. In the work presented in [8], all the record pairs are clustered by their similarity vectors and randomly selected similarity vectors from each cluster are selected for manual labeling. Depending on the output of the manual labeling, similarity vectors in each cluster are automatically labeled as linked or non-linked, or the cluster is further divided into sub-clusters. The system reported in [9] takes as input a small set of training examples, referred to as seeds, to initially train the classification model, which is then used over unseen data.

Unsupervised Record Linkage. In [2], k-means clustering is used to predict the status of a small set of similarity vectors (seeds). Following this, the seeds are used as training set for a supervised learning algorithm. Automatic seed selection, referred to as nearest based, was applied with the self-training process in [10]. In [11] an entity matching algorithm is proposed, which allows to identify best k results for a user-specific scoring function. Unsupervised approaches to RL based on maximizing the value of pseudo f-measure were investigated in [12], [13]. Pseudo f-measure, an unsupervised variant of the f-measure, is formulated using the assumption that while different records often represent the same entity in different repositories, distinct record within one dataset is expected to denote distinct entity. It can be calculated using sets of unlabeled records. The idea is to find the decision rule for record matching which maximizes the value of the pseudo f-measure applying genetic programming [12] or hierarchical search [13]. In more recent work the authors proposed to address the problem of unsupervised record linkage using graphical models [14] and multi view ensemble self-learning [15].

Discussion. While semi-supervised learning significantly reduces the number of manually labeled examples required for generating a classification model, it still requires a certain amount of human input in the training process. Methods that require labeled data for RL are not applicable in many real-world situations; in particular for privacy preserving RL, where the data is private and confidential [16]. While unsupervised methods such as [10] do not require any labeled data, there is much gap to close between them and supervised methods in terms of accuracy.

Problem Definition

Consider a dataset of relational records $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$ where each record comprises values it takes for attributes from a schema $\mathcal{A} = [a_1, a_2, \ldots, a_m]$. Accordingly, we can represent a record r_i as $[r_{i1}, r_{i2}, \ldots, r_{im}]$ where r_{ij} is the value that the i^{th} record takes, for the j^{th} attribute in the schema. For each attribute $a_j \in \mathcal{A}$, we use $\mathcal{S}_j$ to denote the set of similarity measures that are available for the attribute. Examples of common similarity measures include Jaccard, and inverses of edit-distance, or L_1 and L_2 distances. Thus, $\mathcal{S}_j : dom(a_j) \times dom(a_j) \rightarrow \mathbb{R}$, where $dom(a_j)$ denotes the domain of the attribute a_j. Here, we address the task of *unsupervised record linkage scoring*, that of leveraging $\mathcal{R}$ and $\mathcal{S}_j$s to learn a scoring method for pairs from $\mathcal{R}$, the score quantifying the likelihood that both records relate to the same entity. Notationally:

$$[\mathcal{R}, \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}] \xRightarrow[\text{Learning}]{\text{Unsupervised}} RLS : \mathcal{R} \times \mathcal{R} \rightarrow \mathbb{R} \tag{1}$$

Thus, $RLS(r_i, r_j)$ would be a numeric score directly related to the likelihood that r_i and r_j relate to the same entity. A few points are in order; first, unlike the bulk of literature in record linkage that address the linkage problem in the presence of labeled information [17] - i.e., pairs of records that are known to be linked or not-linked - we make use of no such labeled information, and thus address the unsupervised problem. Second, in the interest of retaining generality, we do not necessitate that the scoring by RLS needs to be in $[0, 1]$, or have a probabilistic or possibilistic semantics since that would require corresponding semantics on the similarity measures as well. In other words, for an accurate estimate of RLS, we only expect that pairs that are scored higher are more likely to be linked to the same entity than those are scored lower, i.e., that the relative ordering of pairs on RLS scores is meaningful. Third, *record linkage scoring* is a direct building block for the record linkage problem of classifying pairs of records as either linked or not linked. Applying an appropriate threshold to the RLS scores would yield an intuitive solution to the record linkage problem; one with the pairs that score above the threshold marked as linked, and others as not-linked.

Evaluating Record Linkage Scoring

As is the case with any unsupervised machine learning task, we would like to evaluate the quality of RLS against gold-standard labeled data. This is done by checking the relative ordering between record pairs which are known to be 'linked' and pairs that are known to be 'not linked'. With a threshold on RLS scores yielding a record linkage method, the precision, recall and f-measure on the linked and non-linked classes can be measured on varying values of the threshold. However, most record linkage datasets are very unbalanced [17] with a much larger fraction of unlinked records (recall that this was also the case even in the small example outlined in Table 1). This lopsided distribution makes it easier for RLS methods to achieve high precision and recall on the unlinked class. Thus, rank-aware measures that can incentivize RLS methods that put linked record pairs at the top of the ordering would help better evaluate the quality of RLS methods. Accordingly, we outline two simple evaluation measures below. Let

$\mathcal{L}$ and $\mathcal{U}$ be the set of labeled data comprising linked record pairs and unlinked pairs respectively. Our evaluation measure is then:

$$AR\mathcal{L}(RLS, \mathcal{L}, \mathcal{U}) = average\{Rank_{\mathcal{L},\mathcal{U}}(RLS, l) | l \in \mathcal{L}\}$$

$$MR\mathcal{L}(RLS, \mathcal{L}, \mathcal{U}) = median\{Rank_{\mathcal{L},\mathcal{U}}(RLS, l) | l \in \mathcal{L}\}$$

where $Rank_{\mathcal{L},\mathcal{U}}(RLS, l)$ denotes the rank of the record pair $l \in \mathcal{L}$ in the decreasing RLS-score ordering of record pairs in $(\mathcal{L} \cup \mathcal{U})$. Since our gold standard labellings may not be comprehensive, $(\mathcal{L} \cup \mathcal{U}) \subseteq \mathcal{R} \times \mathcal{R}$. Thus, $AR\mathcal{L}(\ldots)$ and $MR\mathcal{L}(\ldots)$ measures the mean and median ranks of the record pairs in $\mathcal{L}$ in the RLS score ordering. Since we would like to see the record pairs in $\mathcal{L}$ at the top of the ordering, numerically lower values of $AR\mathcal{L}$ and $MR\mathcal{L}$ are desirable. These metrics differ in their character in that $AR\mathcal{L}$ is affected by all changes in orderings of record-pairs, whereas $MR\mathcal{L}$ is less sensitive to outliers at the ends of the ordering.

Our Method

The Scoring Formulation

Consider a record pair $p_{xy} = (r_x, r_y)$ where $\{r_x, r_y\} \subseteq \mathcal{R}$. The similarity between the two records in p_{xy} can be measured along each of the m attributes in $\mathcal{A}$. Further, for each attribute $a_j \in \mathcal{A}$, similarity between the records in p_{xy} can be measured using each of the similarity measures in $\mathcal{S}_j$. Thus, there are $\sum_{j=1}^{m} |\mathcal{S}_j|$ signals of similarity that are available for each record pair, across the similarity measures. Given a set of weights to associate with each of these similarity signals, collectively denoted as $\mathcal{W}$, our method aggregates these similarities using a linear aggregation (weighted sum) into a single value yielding the RLS scores:

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

with $S(r_x.a_j, r_y.a_j)$ denoting the similarity measure between the values for attribute a_j on r_x and r_y, as measured using the similarity measure $S \in \mathcal{S}_j$. The square of the weights, $w_{jS} \in \mathcal{W}$, is used in the formulation for optimization convenience to automatically disallow negative weights. Thus, the crux of our method is in learning the set of weights, $\mathcal{W}$, so that the $RLS_{\mathcal{W}}$ scores estimate the linkage likelihood effectively.

Developing the Objective Function

Our interest is in ensuring that the set of weights, $\mathcal{W}$, leads to an $RLS_{\mathcal{W}}$ scoring that is similar to an "ideal" RLS scoring. Within our unsupervised setting, the notion of ideal-ness needs to be outlined without the luxury of knowing information such as the balance of the $\mathcal{L}$-$\mathcal{U}$ split in $\mathcal{R}$. Thus, we choose to go with a simple goal motivated by *unambiguity* - we would like to learn an estimate of $\mathcal{W}$ such that the resultant weighted-sum based $RLS_{\mathcal{W}}$ scoring can decidedly determine whether each pair of records from $\mathcal{R}$ is to be linked or not. In other words, for every pair of records, we would like the

$RLS_{\mathcal{W}}$ scoring to be either close to the lower extreme (unlinked) or close to the higher extreme (linked), avoiding the (ambiguous) bay between the extremes as much as possible. Since it is evidently impractical to enforce this strictly for all record pairs in $\mathcal{R}$ for such a $\mathcal{W}$ may not even exist, we use an approach of iteratively optimizing an objective function to progressively refine $\mathcal{W}$ in the direction of lesser overall ambiguity.

We now outline an objective function. Consider a record-pair $p_{xy} = (r_x, r_y)$ and an estimate of weights $\mathcal{W}$, the ambiguity for p_{xy} may be modeled as follows:

$$AMB_{\mathcal{W}}(r_x, r_y) \quad = \quad min\big\{RLS_{\mathcal{W}}(r_x, r_y) \;-\; \rho, \tau \;-\; RLS_{\mathcal{W}}(r_x, r_y)\big\} \quad (3)$$

where the $RLS_{\mathcal{W}}$ scores for all record-pairs in $\mathcal{R}$ reside in $[\rho, \tau]$, ρ being the lower extreme and τ being the upper extreme for the $RLS_{\mathcal{W}}$ scores. Informally, $AMB_{\mathcal{W}}(r_x, r_y)$ measures the extent to which the RLS score computed using $\mathcal{W}$ deviates from either ends. The min aggregation ensures that scores in the extremes, i.e., both $RLS_{\mathcal{W}}(r_x, r_y) = \rho$ and $RLS_{\mathcal{W}}(r_x, r_y) = \tau$, would bring the ambiguity score down to zero. This is a desirable condition since these extreme scorings indicate that $RLS_{\mathcal{W}}$ is in no way uncertain about the linkage status for the record pairs in p_{xy}. Analogously, as the $RLS_{\mathcal{W}}$ score moves into the gulf between the extremes, the $AMB_{\mathcal{W}}$ correspondingly goes up.

In most practical cases, similarity functions return non-negative values since negative values for similarity do not make much practical sense. This non-negativity assumption makes 0.0 an intuitive lower bound (ρ) for $RLS_{\mathcal{W}}$ scores. The upper extrem for the $RLS_{\mathcal{W}}$ scores (τ) is set as 1. Incorporating it, we refine the ambiguity notion as:

$$AMB_{\mathcal{W}}(r_x, r_y) \quad = \quad min\big\{RLS_{\mathcal{W}}(r_x, r_y), \tau \;-\; RLS_{\mathcal{W}}(r_x, r_y)\big\} \quad (4)$$

We would like to aggregate this across record-pairs in $\mathcal{R}$ to arrive at a notion of overall ambiguity of $\mathcal{W}$ as follows:

$$AMB_{\mathcal{W}}(\mathcal{R}) \quad = \quad \sum_{\{r_x, r_y \in \mathcal{R}, x \neq y\}} min\big\{RLS_{\mathcal{W}}(r_x, r_y), \tau \;-\; RLS_{\mathcal{W}}(r_x, r_y)\big\} \quad (5)$$

Connecting back to our original goal of reducing ambiguity, our task is simply to learn a set of weights $\mathcal{W}$ that minimizes the overall ambiguity across the dataset.

$$\mathcal{W}^* = \arg\min_{\mathcal{W}} \;\; AMB_{\mathcal{W}}(\mathcal{R}) \quad (6)$$

Our intent in the optimization approach is to ensure that the initial estimates of $\mathcal{W}$ are re-balanced over iterations to orient them towards those similarity signals that can play a role in reducing overall ambiguity. Specifically, all the weights increasing (or decreasing) together do not benefit us much since that would mostly change the range of $RLS_{\mathcal{W}}$ rather than altering the ordering among pairs. In order to be robust and to avoid such cases, we use an add-to-one constraint in our optimization approach.

$$\sum_{a_j \in \mathcal{A}} \sum_{S \in \mathcal{S}_j} w_{jS} = 1 \quad (7)$$

This enforces that all the weights in $\mathcal{W}$ sum up to 1.0. It may be noted that the optimization problem in Eq 6, in combination with the constraint in Eq. 7, while simple to

state, involves searching over all possible settings of $\mathcal{W}$ such that its components add up to 1.0. This is evidently a massive search space, making brute force search impossible. In the next section, we will outline a gradient co-ordinate descent formulation.

Optimization Formulation

The objective to be minimized (from Eq. 6) can be written by expanding $RLS_{\mathcal{W}}$:

$$\sum_{\{r_x,r_y\in\mathcal{R},x\neq y\}} min\{\sum_{a_j\in\mathcal{A}}\sum_{S\in\mathcal{S}_j} w_{jS}^2 \times S_{xyj}, \tau - \sum_{a_j\in\mathcal{A}}\sum_{S\in\mathcal{S}_j} w_{jS}^2 \times S_{xyj}\} \quad (8)$$

where S_{xyj} is a shorthand for $S(r_x.a_j, r_y.a_j)$. The min aggregation function, being not differentiable, does not easily yield to optimization. We observe that exponentiation can be used to approximate the min aggregation (similar to another approximation [18]).

$$min\{a,b\} \approx \frac{1}{\phi} log\Big(exp(\phi a) + exp(\phi b)\Big) \quad (9)$$

This approximation holds for high negative values of ϕ for any numbers a and b. In this paper we set $\phi = -50$. The inner multiplication of a and b separately with ϕ enables spacing out the two terms due to the large numeric value of ϕ; observe that $|a-b| << |\phi a - \phi b|$. For cases where $a > b$ holds, ϕa would be much smaller than ϕb, given that ϕ is a large negative value. Consequently, $exp(\phi a)$ would be much lesser than $exp(\phi b)$, making their sum much closer to the latter than the former. Thus, the log of their sum would be closer to ϕb, making the entire term in the RHS of Eq. 9 a good approximation of b. It is notable that this approximation works reasonably well when $a = b$ too, since the RHS would reduce to $a + log(2)/\phi$; the second term is a very small term, due to having a numerically large ϕ in the denominator, thus yielding a good approximation for min. We simply apply this approximation to re-write our objective:

$$\frac{1}{\phi}\sum_{\{r_x,r_y\in\mathcal{R},x\neq y\}} log\Bigg(exp\Big[\phi\sum_{a_j\in\mathcal{A}}\sum_{S\in\mathcal{S}_j} w_{jS}^2 S_{xyj}\Big] + exp\Big[\phi\Big(\tau - \sum_{a_j\in\mathcal{A}}\sum_{S\in\mathcal{S}_j} w_{jS}^2 S_{xyj}\Big)\Big]\Bigg) \quad (10)$$

Towards optimizing this more convenient and differentiable objective function, we adopt a gradient descent approach, optimizing for one variable within $\mathcal{W}$, at a time. Consider the variable $w_{j'S'}$; the partial derivative, $\frac{\partial AMB_{\mathcal{W}}(\mathcal{R})}{\partial w_{j'S'}}$ is then

$$\sum_{\{r_x,r_y\in\mathcal{R},x\neq y\}} \frac{2w_{j'S'}S'_{xyj'}\Big(exp\Big[\phi\sum_{a_j\in\mathcal{A}}\sum_{S\in\mathcal{S}_j} w_{jS}^2 S_{xyj}\Big] - exp\Big[\phi\Big(\tau - \sum_{a_j\in\mathcal{A}}\sum_{S\in\mathcal{S}_j} w_{jS}^2 S_{xyj}\Big)\Big]\Big)}{\Big(exp\Big[\phi\sum_{a_j\in\mathcal{A}}\sum_{S\in\mathcal{S}_j} w_{jS}^2 S_{xyj}\Big] + exp\Big[\phi\Big(\tau - \sum_{a_j\in\mathcal{A}}\sum_{S\in\mathcal{S}_j} w_{jS}^2 S_{xyj}\Big)\Big]\Big)} \quad (11)$$

The update for $w_{j'S'}$ follows gradient descent[1], using a learning rate, μ.

[1] https://en.wikipedia.org/wiki/Gradient_descent

$$w_{j'S'} = w_{j'S'} - \mu \times \frac{\partial AMB_{\mathcal{W}}(\mathcal{R})}{\partial w_{j'S'}} \tag{12}$$

Algorithm 1: Our Unsupervised Record Linkage Scoring Method

input : Set of Records $\mathcal{R}$ and similarity functions $\{\ldots, \mathcal{S}_j, \ldots\}$
output : Set of weights $\mathcal{W} = \{\ldots, w_{jS}, \ldots\}$ to define the scoring $RLS_{\mathcal{W}}$

1 Initialize all w_{jS}s uniformly satisfying $\sum_{a_j \in \mathcal{A}} \sum_{S \in \mathcal{S}_j} w_{jS} = 1$;
2 While (iterations limit not reached and not converged)
3 Initialize a new set of weights $\mathcal{W}'$
4 $Sum_{\mathcal{W}'} \leftarrow 0$;
5 $\forall\, w_{jS} \in \mathcal{W}$
6 $w'_{jS} \leftarrow w_{jS} - \mu \times \frac{\partial AMB_{\mathcal{W}}(\mathcal{R})}{\partial w_{jS}}$;
7 $Sum_{\mathcal{W}'} = Sum_{\mathcal{W}'} + w'_{jS}$;
8 $\forall\, w'_{jS} \in \mathcal{W}$
9 $w'_{jS} \leftarrow \frac{w'_{jS}}{Sum_{\mathcal{W}'}}$;
10 $\mathcal{W} \leftarrow \mathcal{W}'$;
11 Output $\mathcal{W}$;

Overall Approach

Our iterative approach targets arriving at a good estimate of $\mathcal{W}$ by updating each w_{jS} in turn using the values of $\mathcal{W}$ from the previous iteration, followed by re-normalizing them to sum up to 1.0 within each iteration. This approach is outlined in Algorithm 1. Line 6 denotes the gradient descent update, whereas Line 9 performs the normalization. **Complexity:** The update in Eq. 12 has terms for each record pair to be evaluated hidden within the slope term, i.e., $\frac{\partial AMB_{\mathcal{W}}(\mathcal{R})}{\partial w_{jS}}$, and needs to be run for each similarity measure for each attribute (notice the iteration over w_{jS}s). From a computational perspective, consider the construction of the slope term (Ref. Eq. 12); it may be observed that the inner term in the numerator (difference between exponentiated terms) and the inner term in the denominator (sum of exponentiated terms) are both independent of j' and S', and thus, can be computed once per record pair. This makes the full complexity $\mathcal{O}(\sum_{j=1}^{m} |\mathcal{S}_j| \times \mathcal{P})$ per iteration where $\mathcal{P}$ is the number of record pairs. It may be noted that the $\sum_{j=1}^{m} |\mathcal{S}_j|$ term is small, there being only a handful of attributes and a handful of similarity measures, making the complexity largely dependent on the size of $\mathcal{P}$. We will show later that our method stabilizes to reasonable accuracy in 100s of iterations in our experimental section. Coming to $\mathcal{P}$, though the number of possible record pairs in $\mathcal{R}$ is quadratic in $|\mathcal{R}|$, typical record linkage scenarios use efficient blocking strategies to rule out a large fraction of record pairs from being considered for linkage determination making $\mathcal{P} << |\mathcal{R}|^2$. Usage of better blocking strategies would benefit our method since they reduce $\mathcal{P}$, leading to our method running faster.

Experiments and Results

Experimental Setup

In the experiments, we use gold standard linkage labellings to evaluate the methods. It may be emphasized here that the gold standard labellings were used only for the evaluation purposes. All experiments were run on a workstation with Intel(R) Core(TM) i7-4790 CPU @ 3,60 GHz processor, 16 GB (RAM) and 64-bit Windows 7. As with a typical RL approach, we perform blocking as the first step to reduce the number of record pairs for each of the RL methods. Any blocking method could be employed, blocking being orthogonal to the task we evaluate; we used the recently proposed unsupervised blocking scheme learner [19] in our evaluation.

Baselines. As baselines for our method, we used two unsupervised RL methods [12, 13] based on the pseudo f-measure. To compare the two methods against our approach we used the similarity scoring configurations output by each of the method to rank all record pairs. Following this, we use $AR\mathcal{L}$ and $MR\mathcal{L}$ to evaluate each of the rankings. All baseline parameters were set to the values recommended in respective papers.

Data. The experiments were conducted with three real world datasets commonly used for evaluating RL methods: Restaurant, Cora and ACM-DBLP. The Restaurant dataset contains 864 restaurant records (with 112 pairs of matching records), each with five fields, including name, address, city, phone and type. The Cora dataset is a collection of 1,295 (with 14,184 pairs of matching records) citations to computer science papers. Each citation is represented by 4 fields (author, title, venue, year). The ACM-DBLP is a bibliographic datasets of Computer Science bibliography records represented by four attributes (author, title, venue, year). The total number of entity pairs is 6,001,104.

Parametrization of the algorithms. One parameter needs to be set to run our method, which is the number of iterations. The reported results were obtained for number of iterations equals to 200 for each of the 3 datasets. For each of the evaluated methods we applied five commonly used similarity measures for RL, namely Jaro [20], Jaro-Winkler [21], Jaccard [22], Q-Gram [23], and Levenshtein edit distance [24].

Comparative evaluation of record pairs ordering

Effectiveness. Table 2 lists the results of the comparative evaluation of our method against the genetic algorithm and linear classifiers baselines, among the latest methods for unsupervised ML-based record linkage. For each of the evaluation measures, $AR\mathcal{L}$ and $MR\mathcal{L}$, lower values are better, as seen in Section 1. Recall measures the fraction of correctly linked record pairs among the top $|\mathcal{L}|$ pairs; thus, this is the recall of the method if the top-$\mathcal{L}$ pairs according to the scoring were output as the result set. We additionally have also included the values of the measures that can be achieved by a perfect scoring, one that puts all linked record pairs at the top, followed by the unliked pairs; this implicitly has a recall of 1.0. This enables understanding the gap between our method and the best possible scoring. It can be observed that our method obtained better result than two baseline methods for Cora and DBLP-ACM datasets. For the Restaurant dataset, our method was outperformed by the genetic algorithm based approach, which is able to navigate the search space using the randomized approach effectively for the

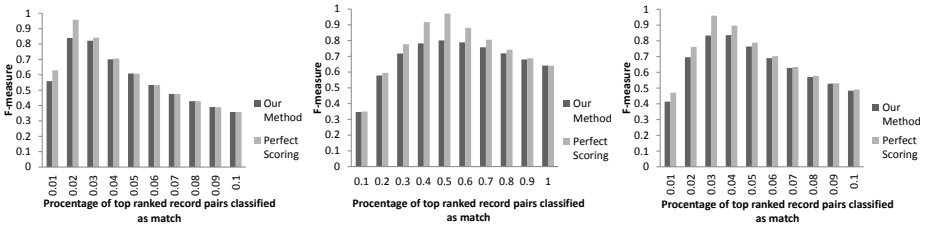
Table 2: Evaluation over Rank-aware Metrics and Recall (best numbers highlighted).

	Genetic algorithm			Linear classifier			Our method			Perfect scoring	
	AR$\mathcal{L}$	MR$\mathcal{L}$	Rec.	AR$\mathcal{L}$	MR$\mathcal{L}$	Rec.	AR$\mathcal{L}$	MR$\mathcal{L}$	Rec.	AR$\mathcal{L}$	MR$\mathcal{L}$
Restaurant	**59.6**	**56**	**0.92**	79.42	67	0.72	65.2	61	0.89	55.5	55.5
Cora	8514.5	7006.5	0.79	9396.2	8858.5	0.75	**8077.3**	**6836.5**	**0.82**	6756	6756
DBLP-ACM	1475.2	1444.5	0.65	1769.6	1009.5	0.84	**1100**	**926.5**	**0.85**	851	851

Table 3: The execution time: *hours : minutes : seconds* (best numbers highlighted).

	Genetic algorithm	Linear classifier	Our method
Restaurant	00:19:42	30:26:06	**00:15:07**
Cora	17:34:01	22:55:34	**00:07:27**
DBLP-ACM	30:26:47	72:00:00 +	**02:14:58**

small dataset. However, as expected, our method, due to it's rather highly focused search along the space of solutions, is able to achieve better effectiveness over large datasets (and their correspondingly larger solution spaces). This character is well pronounced in the large DBLP-ACM dataset where our method is able to make massive improvements.
Efficiency. For each of the evaluated methods we measure their execution time, baselines implemented according to our best understanding from the respective papers. The run times of each of the methods are reported in table 3. We can observe that our method was able to perform the linkage process significantly faster.



(a) Restaurant (b) Cora (c) DBLP-ACM

Fig. 1: F-measure obtained with different values of cut off point.

Further Analysis of Our Method

F-Measure at different Cut-offs. For usage of our record linkage scoring method in a practical scenario, one would need to apply a cut-off point in the ranked list, so that record pairs above that cut-off could be regarded as linked, and those below may be considered as unlinked. The recall (for the linked class) reported in the previous section

Table 4: Evaluation of Resilience to Ambigious Similarities.

	With one HAS			With two HASs			Without HAS		
	AR$\mathcal{L}$	MR$\mathcal{L}$	Rec.	AR$\mathcal{L}$	MR$\mathcal{L}$	Rec.	AR$\mathcal{L}$	MR$\mathcal{L}$	Rec.
Restaurant	65.5	62	0.88	65.5	62	0.88	65.2	61	0.89
Cora	8073.8	6837.5	0.82	8169.7	6939.5	0.81	8077.3	6836.5	0.82
DBLP-ACM	1101.8	927	0.85	1099	926	0.85	1100	926.5	0.85

is precisely equal to the recall when the cut-off point is chosen after $\mathcal{L}$ records; at this cut-off, the precision and recall are equivalent due to the usage of the same denominator. However, information about $\mathcal{L}$ is part of the gold-standard data and is not available within a realistic record linkage setting. Thus, we evaluate the recall, precision and F-measure for the linked class over varying cut-offs, to illustrate the effectiveness trends of our methods at varying cut-offs. We plot the F-measure achieved by our method against that of the perfect ordering for each of the datasets in Figure 1, with the choice of the percentage of all record pairs in the dataset used for the cut-off point indicated in the X-axis. The range of cut-off points studied differ for the Cora dataset due to the higher ratio of matching records. Our method is seen to trail the perfect scoring quite closely in some parts of the space, with the gap widening, though not significantly, at other parts. This further illustrates the effectiveness of our method.

Resilience to Noisy Similarity Measures: We now study the resilience of our method to *highly ambigious* similarity measures (abbreviated to HASs), those that hold no utility in separating the linked and unlinked record pairs. We conduct this study through the usage of similarity measures that simply sample from a normal distribution centered at 0.5, the value midway between the linkage favoring extreme of 0.0 and the other extreme of 1.0; we use a standard deviation of 0.2. Table 4 lists the results of our method when one or two HASs are added to the dataset. The results show that the effectiveness deteriorations of our methods in the presence of HASs are decidedly miniscule.

Conclusions

In this paper we addressed the problem of unsupervised RL and proposed a novel approach to RL, which models the task as an optimization problem. Our optimization formulation searches for RL methods that use a weighted sum scoring to determine linkages between records, favoring methods that are less ambiguous overall, in the linkage decisions they make. Our experimental results indicate that our method is highly effective in making accurate linkage decisions, while also being orders of magnitude faster than existing approaches, especially on large datasets. We also illustrated that our method is fairly accurate as an RL method at different cut-off points, and that our optimization approach is exceedingly robust to noisy similarity measures.

References

1. Steorts, R.C., Ventura, S.L., Sadinle, M., Fienberg, S.E.: A comparison of blocking methods for record linkage. In: International Conference on Privacy in Statistical Databases, Springer

(2014) 253–268
2. Elfeky, M.G., Verykios, V.S., Elmagarmid, A.K.: Tailor: A record linkage toolbox. In: Data Engineering, 2002. Proceedings. 18th International Conference on, IEEE (2002) 17–28
3. Isele, R., Bizer, C.: Learning expressive linkage rules using genetic programming. Proceedings of the VLDB Endowment **5**(11) (2012) 1638–1649
4. Wang, J., Li, G., Yu, J.X., Feng, J.: Entity matching: How similar is similar. Proceedings of the VLDB Endowment **4**(10) (2011) 622–633
5. Arasu, A., Gotz, M., Kaushik, R.: On active learning of record matching packages. In: SIGMOD, ACM (2010) 783–794
6. Sarawagi, S., Bhamidipaty, A.: Interactive deduplication using active learning. In: SIGKDD, ACM (2002) 269–278
7. Cohen, W.W., Richman, J.: Learning to match and cluster large high-dimensional data sets for data integration. In: SIGKDD, ACM (2002) 475–480
8. Wang, Q., Vatsalan, D., Christen, P.: Efficient interactive training selection for large-scale entity resolution. In: PAKDD, Springer (2015) 562–573
9. Kejriwal, M., Miranker, D.P.: Semi-supervised instance matching using boosted classifiers. In: European Semantic Web Conference, Springer (2015) 388–402
10. Christen, P.: Automatic record linkage using seeded nearest neighbour and support vector machine classification. In: SIGKDD, ACM (2008) 151–159
11. Lee, S., Lee, J., Hwang, S.w.: Efficient entity matching using materialized lists. Information Sciences **261** (2014) 170–184
12. Nikolov, A., dAquin, M., Motta, E.: Unsupervised learning of link discovery configuration. In: Extended Semantic Web Conference, Springer (2012) 119–133
13. Ngomo, A.C.N., Lyko, K.: Unsupervised learning of link specifications: deterministic vs. non-deterministic. In: Proceedings of the 8th International Conference on Ontology Matching-Volume 1111, CEUR-WS. org (2013) 25–36
14. Steorts, R.C., Hall, R., Fienberg, S.E.: A bayesian approach to graphical record linkage and deduplication. Journal of the American Statistical Association **111**(516) (2016) 1660–1672
15. Jurek, A., Hong, J., Chi, Y., Liu, W.: A novel ensemble learning approach to unsupervised record linkage. Information Systems **71** (2017) 40–54
16. Vatsalan, D., Christen, P., Verykios, V.S.: A taxonomy of privacy-preserving record linkage techniques. Information Systems **38**(6) (2013) 946–969
17. Christen, P.: Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection. Springer Science Business Media (2012)
18. P, D.: Mixkmeans: Clustering question-answer archives. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP. (2016) 1576–1585
19. Kejriwal, M., Miranker, D.P.: An unsupervised algorithm for learning blocking schemes. In: Data Mining (ICDM), 2013 IEEE 13th International Conference on, EEE (2013) 340–349
20. Jaro, M.A.: Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. Journal of the American Statistical Association **84**(406) (1989) 414–420
21. Winkler, W.E.: String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. (1990) 354–359
22. Jaccard, P.: Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. Bull. Soc. Vaud. Sci. Nat. **37** (1901) 241–272
23. Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review **5**(1) (2001) 3–55
24. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. Volume 10. (1966) 707–710