Frontiers of Computer Science CN 10-1014/TP
DOI 10.1007/s11704-018-7285-8 ISSN 2095-2228
RESEARCH ARTICLE

GL-RF: A Reconciliation Framework for Label-free Entity Resolution

Yaoli XU[1,2], Zhanhuai LI[1,2], Qun CHEN[1,2], Fengfeng FAN[1,2]

1 School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072, China
2 Key Laboratory of Big Data Storage and Management, Northwestern Polytechnical University, Ministry of Industry and Information Technology, Xi'an 710072, China

Front. Comput. Sci., **Just Accepted Manuscript** • 10.1007/s11704-018-7285-8
http://journal.hep.com.cn on March 20, 2018

Front.Comput.Sci.
DOI

RESEARCH ARTICLE

GL-RF: A Reconciliation Framework for Label-free Entity Resolution

Yaoli XU [1,2], **Zhanhuai LI** [1,2], **Qun CHEN** (✉)[1,2], **Fengfeng FAN** [1,2]

1 School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, P.R. China, 710072
2 Key Laboratory of Big Data Storage and Management, Northwestern Polytechnical University, Ministry of Industry and Information Technology

Abstract Entity resolution (ER) plays a fundamental role in data integration and cleaning. Even though many approaches have been proposed for ER, their performance on the data sets with different characteristics may vary significantly. Given a practical ER workload, it remains very challenging to select an appropriate resolution technique while applying multiple techniques simultaneously would result in inconsistent resolution. In this paper, we study the problem of how to reconcile the inconsistent results of different label-free resolution techniques. We first propose a generic label-free reconciliation framework, denoted by GL-RF. The proposed framework does not require to manually label pairs, but reasons about the match status of the inconsistent pairs purely based on the implicit information contained in the consistent pairs. We then formalize the reconciliation problem and present an incremental K-neighbor influence algorithm. Finally, we empirically evaluate the performance of the proposed approach on the real data sets by a comparative study. Our extensive experiments show that GL-RF performs considerably better than the state-of-the-art alternatives.

Keywords data quality, data integration, entity resolution, inconsistent pairs, vector representation

1 Introduction

Entity resolution (ER), also referred to as record linkage and entity matching [1], is a long-standing challenge, existing in various data management systems, particularly data integration and cleaning systems. As so often happens, there exist multiple data sources which store duplicate real-world entity information in different descriptions, incurred by misspellings, typos, diverse name conventions, random usage of the abbreviation or full name, ongoing changes such as in DBpedia [2], and so forth. The purpose of ER is to determine whether two data records describe the same real-world entity.

The existing techniques for ER can be broadly categorized into learning-based [3, 4] and label-free [5–7]. The learning-based techniques require manually labeled training data while the label-free techniques do not have such requirement. The performance of the learning-based techniques to a large extent depends on the amount of available training data. Unfortunately, in practice, a new ER workload may come with a very limited amount of training data. In this scenario, the label-free techniques are usually more appropriate. Even though there are many label-free techniques proposed for ER, their performance on the data sets with different characteristics may vary significantly. A practitioner, provided with a new ER workload, may still encounter the challenging problem of how to choose an appropriate technique. It can be observed that in the absence of manually labeled training data, it is nearly impossible to pick the best performer among the existing techniques. A natural alternative is therefore to first simultaneously apply different techniques and then reconcile inconsistent results. Consider the following example:

Example 1 Two databases D_1 and D_2, as shown in Tables 1 and 2, consist of a wealth of literature records. Each record describes the title, authors and identity of an article. A pair consists of two records (i.e., one in D_1 and the other

Received Auguest 11, 2017; accepted March 20, 2018

E-mail: *yaolixu@mail.nwpu.edu.cn*

Table 1 Records on D_1

rID(D_1)	title	authors
$r_{1,1}$	An On-Line Self-Tuning Markov Histogram for XML Path Selectivity Estimation	L Lim, M Wang, S Padamanabhan, JS Vitter, RXPL
...	...	...
$r_{1,8}$	Loading a Cache with Query Results	LM Haas
$r_{1,9}$	An investigation into loading a cache with query results	L Haas, D Kossmann, I Ursu

Table 2 Records on D_2

rID(D_2)	title	authors
$r_{2,1}$	XPathLearner: An On-line Self-Tuning Markov Histogram for XML Path Selectivity Estimation	L Lim, M Wang, S Padmanabhan, J Vitter, R Parr
...	...	...
$r_{2,8}$	Loading a Cache with Query Results	L Haas, D Kossmann, I Ursu

Table 3 An Example of Inconsistent Pair Reconciliation Problem

pID	rID(D_1)	rID(D_2)	M_1	M_2	M_3	Golden
$p_{1,1}$	$r_{1,1}$	$r_{2,1}$	*P*	*P*	*P*	*P*
$p_{2,2}$	$r_{1,2}$	$r_{2,2}$	*P*	*P*	*P*	*P*
$p_{3,4}$	$r_{1,3}$	$r_{2,4}$	*P*	*P*	*P*	*N*
$p_{4,5}$	$r_{1,4}$	$r_{2,5}$	*N*	*N*	*N*	*N*
$p_{5,4}$	$r_{1,5}$	$r_{2,4}$	*N*	*N*	*N*	*N*
$p_{6,6}$	$r_{1,6}$	$r_{2,6}$	*N*	*N*	*N*	*P*
$p_{7,7}$	$r_{1,7}$	$r_{2,7}$	*P*	*N*	*P*	*P*
$p_{8,8}$	$r_{1,8}$	$r_{2,8}$	*P*	*N*	*N*	*P*
$p_{9,8}$	$r_{1,9}$	$r_{2,8}$	*N*	*P*	*P*	*N*
$p_{8,10}$	$r_{1,8}$	$r_{2,10}$	*N*	*N*	*P*	*N*
$p_{12,11}$	$r_{1,12}$	$r_{2,11}$	*N*	*P*	*N*	*N*
$p_{13,13}$	$r_{1,13}$	$r_{2,13}$	*P*	*P*	*N*	*N*

in D_2). Table 3 presents the resolution results of the twelve pairs solved by a rule-based method (M_1), a distance-based method (M_2), and a cluster-based method (M_3), where *P* indicates the *match*, *N* indicates the *non-match* and the values at the column of Golden indicate the ground-truth status. The rule-based method leverages a variety of rules derived from human experts or data sets, the distance-based method investigates how to propose an appropriate metric with certain threshold through statistical analyses, and the cluster-based method depends on an intuition that the more similar structure records have, the more likely these records depict the same real-world entity. It can be observed that M_1, M_2 and M_3 produce both consistent results (e.g., on the pairs, $p_{2,2}$ and $p_{3,4}$) and inconsistent results (e.g., on the pairs, $p_{8,8}$ and $p_{13,13}$). The performance of different techniques can also vary on different pair instances. For instance, M_1 correctly predicts the status of the pairs, $p_{8,8}$ and $p_{9,8}$, while M_3 fails at both of them. In contrast, at the pair of $p_{13,13}$, M_1 fails but M_3 succeeds.

In the example shown above, a consistent pair can be obviously labeled by the consensus resolution result. The problem of how to label the inconsistent pairs is however non-trivial and challenging. The naive approaches based on majority or weighted voting [8] usually have limited effectiveness in practical scenarios. In this paper, we propose a novel framework to reconcile the conflicting results of different label-free resolution techniques. The proposed framework reasons about the match status of inconsistent pairs purely based on consistent pairs. It thus has the attractive property of not requiring any manually labeled results. The major contributions of this paper can be summarized as follows:

1. We propose a novel reconciliation framework for label-free entity resolution. It can effectively take advantage of the hints implied by the consistent pairs to reason about the match status of the inconsistent pairs.
2. We investigate the inconsistent pair reconciliation problem and present a solution based on k-neighbor influence estimation. The solution represents a pair by a tf-idf vector and quantifies the influence of a reference pair over an inconsistent pair by their vector similarity.
3. We conduct extensive experiments on the real-world data sets to evaluate the performance of the proposed framework. Our experimental results show that it performs considerably better than the state-of-the-art alternatives on quality and its efficiency scales well with data size.

The rest of this paper is organized as follows: We review more related work in Section 2. We present an overview of the proposed framework in Section 3. We define the reconciliation problem and present the incremental k-neighbor influence algorithm in Section 4. We present our empirical evaluation results in Section 5. Finally, we conclude this paper with some thoughts on future work in Section 6.

2 Related Work

Previous researchers realized the importance of entity resolution (ER). Therefore, it has been extensively studied [9] and a variety of models or tools [1, 10–15] have been proposed.

Several studies focus on rule-based approaches that extract information from either data or ad-hoc experts. [16] defines Matching Dependencies (MDs), which tell us whether two records match or not, and provides a mechanism for inferring more MDs, according to a small set of MDs. [3, 4] can learn entity matching rules from the training set, with the aim of identifying records that depict the same real world entities. [3] proposed ER-rule, which servers for non-pairwise entity resolution approaches, and [4] defined the record-matching rule (RR), which can be applied by pairwise entity resolution approaches. We refer to label-free entity resolution methods as individual methods, so as to distinguish them from reconciliation methods. Our individual rule-based method employs matching rules, that are generated in the form of RR with a weight assigned to each record-matching rule.

There are several works [5–7] to identify matches using distances for the case where training data and domain knowledge are absent. The Footrule Distance was defined by [6], and was leveraged to identify the top-k tuples "most related" to a given tuple in the merging process of ranked lists of approximate match operations. [7] presented a distance-based measure, a weighted sum of the distances between attribute values, and then leveraged it to assist in a decision-theoretic model for identifying matches. [5] defined the centrifugal distance of a record pair to guide an Outlier-Detection based approach for entity matching. We select the representative work [5] as an individual method.

There are various unsupervised clustering techniques proposed in the machine learning community, such as K-Means clustering, Gaussian mixture model and their variants [17, 18]. Several researchers leverage these techniques to group record pairs into a match or non-match cluster. The Freely Extensible Biomedical Record Linkage system (Febrl) [10] provides the 'KMeans' and 'FarthestFirst' models. We select its 'KMeans' implementation as the representative work among cluster-based approaches.

Other literature is in pursuit of extra information such as Crowdsourcing [14] and the combination of human and machine wisdom [15] to improve the performance of entity resolution. However, among them, there is little endeavor to employ the results of off-the-shelf approaches.

There are several studies, with the aim of devising convenient frameworks. [10] offered Febrl, which contains a series of off-the-shelf record linkage techniques for practitioners, and provides a convenient GUI. [19] has proposed a framework for evaluating entity resolution (FEVER), which serves to automatically evaluate different entity resolution approaches under different configurations, in order to provide better parameter settings. In contrast, our framework is to cooperate with individual methods to improve the effectiveness of entity resolution.

Closer to our work are the ensemble models [17] in the machine learning community, such as AdaBoost, Random Forest and so forth. However, as these models require the presence of training data, it is impossible to apply these models for performing the entity matching task without sufficient training data. Our framework focuses on the case where training data is not accessible.

3 The Reconciliation Framework

In this section, we present a Generic Label-free Reconciliation Framework for entity resolution (GL-RF). For the sake of presentation, we introduce the used notations in Table 4.

GL-RF, as shown in Fig. 1, takes two relational tables, D_1 and D_2, and the label results of multiple resolution techniques, M1, M2 , . . . , Mn, as input. It divides the pair instances into two sets: (i) one consisting of consistent pairs, whose labels stemmed from individual techniques are the same, and (ii) the other consisting of the inconsistent pairs, whose labels stemmed from individual techniques are conflicting. The reconciliation component consists of three procedures: reference pair generation, pair vector representation and inconsistent pair reconciliation. In the rest of this section, we detail these three procedures in turn.

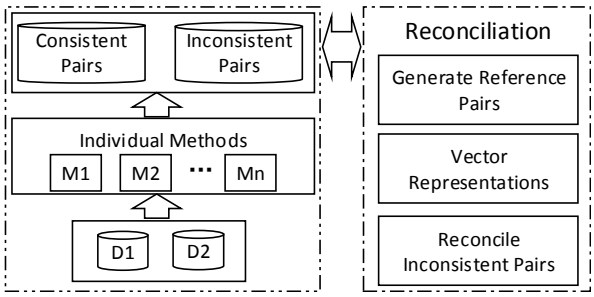


Fig. 1 Framework Overview

Notations. Given two relational tables (D_1, D_2), and individ-

Table 4 Summary of Notations

Symbols	Semantics
D_i	the i-th relational table that contains a set of records
$r_{i,j}$	the j-th record in D_i
M_k	the k-th individual method
$p_{i,j}$	a pair (i.e., a record pair) that contains two records (i.e., $r_{1,i} \in D_1$ and $r_{2,j} \in D_2$)
P (or N)	a match (or non-match) label
$P(\mathsf{M}_k)$	match pairs generated by M_k
$N(\mathsf{M}_k)$	non-match pairs generated by M_k
P^c	consistent pairs generated by $\mathsf{M}_1, \mathsf{M}_2, \ldots, \mathsf{M}_n$
P^{cp} (or P^{cn})	consistent positive (or negative) pairs
$P^{\bar{c}}$	inconsistent pairs generated by $\mathsf{M}_1, \mathsf{M}_2, \ldots, \mathsf{M}_n$
p_r	a reference pair
$R^p (or R^n)$	reference positive (or negative) pairs

ual methods (M_1, M_2, ..., M_n), the results of some method M_k are comprised of matched pairs $P(\mathsf{M}_k)$ and non-matched pairs $N(\mathsf{M}_k)$, where $P(\mathsf{M}_k)$ (or. $N(\mathsf{M}_k)$) can be produced explicitly (or implicitly). All pairs can be classified into sconsistent pairs P^c and inconsistent pairs $P^{\bar{c}}$. P^c can be further divided into the consistent positive pairs P^{cp} and the consistent negative pairs P^{cn}, according to whether consistent pairs are matched or non-matched.

Example 2 Consider two relational tables, D_1 and D_2, and $\mathsf{M}_k (1 \leq k \geq 3)$. Table 3 shows results of three individual methods for a given entity matching task. Based on the semantics of $P(\mathsf{M}_k)$ and $N(\mathsf{M}_k)$, we can derive the following: (i) $P(\mathsf{M}_1) = \{p_{1,1}, p_{2,2}, p_{3,4}, p_{7,7}, p_{8,8}, p_{13,13}\}$ and $N(\mathsf{M}_1) = \{p_{4,5}, p_{5,4}, p_{6,6}, p_{9,8}, p_{8,10}, p_{12,11}\}$. For M_1, the predicted labels of pairs in $P(\mathsf{M}_1)$ (resp. $N(\mathsf{M}_1)$) are P (resp. N). (ii) $p_{1,1}$ is a consistent positive pair, and $p_{4,5}$ is a consistent negative pair. Then we can derive $P^{cp} = \{p_{1,1}, p_{2,2}, p_{3,4}\}$, $P^{cn} = \{p_{4,5}, p_{5,4}, p_{6,6}\}$, and $P^c = P^{cp} \cup P^{cn}$. (iii) $p_{8,10}$ is an inconsistent pair, owing to $p_{8,10} \in P(\mathsf{M}_3)$ and $p_{8,10} \in N(\mathsf{M}_1)$. Further, our work focuses on how to reconcile pairs in $P^{\bar{c}}$ (i.e., $\{p_{7,7}, p_{8,8}, p_{9,8}, p_{8,10}, p_{12,11}, p_{13,13}\}$). □

Reference Pair Generation. In general, the set of reference pairs can be composed of all the consistent pairs. However, in practice, the set of consistent pairs is usually huge because it contains many non-matched pairs. It can be observed that the consistently non-matched pairs with very low similarity usually are far away from inconsistent pairs. Therefore, we propose to filter these pairs according to their similarity, and select its subset as reference negative pairs R^n. We regard P^{cp} as reference positive pairs R^p, since pairs in P^{cp} contain valuable information of matched pairs.

Pair Vector Representation. Pair vector was traditionally represented by the similarity of the attribute values of two relational records. However, the values of different attributes may contain common contents (e.g., both the name and description of a product contain its product number). Therefore, GL-RF represents a record by the tf-idf [20] values of its words (or phrases) and represents a pair by its two records' vector difference. Note that each dimension of a pair vector corresponds to a word or phrase. The tf-idf value of a word quantifies its significance for entity resolution. It is computed based on the entire data set. The vector representation for a given record r or a pair p is denoted as $v(r)$ or $v(p)$.

Example 3 In Tables 1 and 2, given $v(r_{1,1}) = [0.19, 0.0, \cdots, 0.28]$ and $v(r_{2,1}) = [0.19, 0.0, \cdots, 0.28]$, $v(p_{1,1})$ is $[0.0, 0.0, \cdots, 0.28]$ based on $v(p_{i,j}) = v(r_{1,i}) - v(r_{2,j})$.

Inconsistent Pair Reconciliation. In GL-RF, the reference set (a subset of consistent pairs) and the test set (consisting of inconsistent pairs) are not drawn from the same distribution. Therefore, the traditional learning-based approaches (e.g. SVM and Decision Tree) can not be applied for reconciliation, according to probably approximately correct learning [21], under which results of the learning theory were proved. We instead propose a K-neighbor Influence Algorithm (KIA). It essentially predicts the status of a pair by aggregating the influence of its neighbors.

4 An Algorithm for Reconciling Inconsistent Pairs

In this section, we propose our solution for the reconciliation problem, and present the K-neighbor Influence Algorithm (KIA). We also provide the complexity analysis of the proposed algorithm.

Given a set of consistent pairs, P^c, and a set of inconsistent pairs, $P^{\bar{c}}$, the problem is to predict whether an inconsistent pair $p^{\bar{c}}$ in $P^{\bar{c}}$ matches or not. We construct our solution based on the following observation: a pair can be considered to be a match pair if it is generally more similar to its top-k matched reference pairs than its top-k non-matched reference pairs, and vice versa.

We refer to a Gaussian function of the distance between an inconsistent pair vector, $v(p^{\bar{c}})$, and a reference pair vector, $v(p_r)$, as the influence of p_r over $p^{\bar{c}}$. The influence of p_r over $p^{\bar{c}}$ decreases with their increasing distance. Formally, we define the influence strength as follows.

$$IS(p_r, p^{\bar{c}}) = e^{-0.5 \cdot d(v(p^{\bar{c}}), v(p_r)) \cdot d(v(p^{\bar{c}}), v(p_r))} \quad (1)$$

where $v(p_r)$ and $v(p^{\bar{c}})$ are separately the vector representations of p_r and $p^{\bar{c}}$, and $d(v(p^{\bar{c}}), v(p_r))$ is the Euclidean dis-

tance between $v(p_r)$ and $v(p^{\bar{c}})$.

A positive (or negative) reference pair p_r^p(or p_r^n) provides extra evidence that its neighboring pair $p^{\bar{c}}$ is likely matched (or non-matched). We refer to this evidence as a signal label of p_r, denoted by $\iota(p_r)$. If $p_r \in R^P$, $\iota(p_r)$ is 1, otherwise $\iota(p_r)$ is -1.

We incorporate the vote information as a penalty factor for the influence strength, in order to avoid the mistake that a truly matched (or non-matched) pair with less (or more) votes is mistakenly considered as non-matched (or matched). For instance, if $p^{\bar{c}}$ achieves more votes, the inference strength from R^p should be with a little penalty. Given p_r and $p^{\bar{c}}$, we define the penalty factor, denoted by $PF(p_r, p^{\bar{c}})$, as the following.

$$PF(p_r, p^{\bar{c}}) = \begin{cases} 1 - P_{vote}(p^{\bar{c}}) & \iota(p_r) = 1 \\ P_{vote}(p^{\bar{c}}) & \iota(p_r) = -1 \end{cases} \quad (2)$$

where $\iota(p_r)$ is the signal label, and $P_{vote}(p^{\bar{c}})$ is the percentage of votes achieved by $p^{\bar{c}}$. The entity reconciliation information is represented by the product of $PF(p_r, p^{\bar{c}})$, $\iota(p_r)$ and $IS(p_r, p^{\bar{c}})$, as Eq. 3 shows.

$$ERI(p_r, p^{\bar{c}}) = PF(p_r, p^{\bar{c}}) \cdot \iota(p_r) \cdot IS(p_r, p^{\bar{c}}) \quad (3)$$

Reference positive pairs R^p can be considered to have a distribution of the Euclidean norm of their pair vectors. $p^{\bar{c}}$ tends to be matched, if it appears to be more similar to R^p. We leverage the novelty detection model [22] to achieve the distribution. Its core idea is to construct a rough, close frontier which describes the contour of the distribution of these existing observations (e.g., pairs in R^p). If a pair $p^{\bar{c}}$ is far from the frontier, then $p^{\bar{c}}$ is non-matched with a higher probability, and vice versa. For $p^{\bar{c}}$, this model outputs $D(p^{\bar{c}})$ (i.e., the distance between $p^{\bar{c}}$ and the frontier). $D(p^{\bar{c}})$ has the following semantics. (i) If $D(p^{\bar{c}})$ is negative (resp. positive), it means that $p^{\bar{c}}$ is likely to be non-matched (resp. matched). (ii) If $p^{\bar{c}}$ achieves a larger $|D(p^{\bar{c}})|$, the prediction has higher confidence.

Accumulative Entity Reconciliation Information. We define the reconciliation measure, as an accumulative entity reconciliation information $AERI(R, p^{\bar{c}})$, which is the combination of $D(p^{\bar{c}})$ and $ERI(R, p^{\bar{c}})$ according to Eq. 4.

$$AERI(R, p^{\bar{c}}) = (1 - \alpha) * n(D(p^{\bar{c}})) + \alpha * n(ERI(R, p^{\bar{c}})) \quad (4)$$

where $ERI(R, p^{\bar{c}})$ is the mean of $ERI(p_r, p^{\bar{c}})$ over $p_r \in R$, α is a weight which stands for the percentage of $ERI(R, p^{\bar{c}})$ in $AERI(R, p^{\bar{c}})$, and $n(x)$ is the normalization function which first changes all distances of inconsistent pairs between -10 and 10, and then transforms certain distance into a probability of 0 to 1.

Example 3 As shown in Table 3, $P^{\bar{c}}$ is $\{p_{7,7}, p_{8,8}, p_{9,8}, p_{8,10}, p_{12,11}, p_{13,13}\}$, and R is $\{p_{1,1}, p_{2,2}, p_{3,4}, p_{4,5}, p_{5,4}, p_{6,6}\}$. Given $d(v(p_{7,7}), v(p_{1,1})) = 1.32$ and $P_{vote}(p_{1,1}) = 0.67$, according to Eq. 1, the influence strength of $p_{1,1}$ over $p_{7,7}$ (i.e., $IS(p_{1,1}, p_{7,7})$) is 0.42. As $p_{1,1}$ is in R^p, we can obtain $\iota(p_{1,1}) = 1$, $PF(p_{7,7}, p_{1,1}) = 0.33$ and $ERI(p_{1,1}, p_{7,7}) = 0.14$, according to Eqs. 2–3. Given α= 52.5%, $n(D(p_{7,7})) = 0.8$, $ERI(p_{1,1}, p_{7,7}) = 0.14$, $ERI(p_{2,2}, p_{7,7}) = 0.26$, $ERI(p_{3,4}, p_{7,7}) = 0.3$, $ERI(p_{4,5}, p_{7,7}) = -0.35$, $ERI(p_{5,4}, p_{7,7}) = -0.39$, and $ERI(p_{6,6}, p_{7,7}) = -0.37$, we get $AERI(R, p_{7,7}) = 0.51$, according to Eq. 4. □

4.1 K-neighbor Influence Algorithm

Our reconciliation algorithm KIA is to deal with inconsistent pairs one by one, leveraging the reference pairs and previously resolved inconsistent pairs. It takes as input $P^{\bar{c}}$, the neighbor number of $p^{\bar{c}}$ k, the weight α, the ratio of non-matched to matched pairs in R Ratio, pair vectors $\mathbb{V}$, and the results of individual methods I. It returns a set of matched pairs M and a set of non-matched pairs $\overline{M}$. We present the procedure of KIA as follows.

As shown in Algorithm 1, KIA first initializes R by getRefPairs and a key-value structure $\mathbb{H}$ (lines 1–12). Then it evokes compDis to compute $D(p^{\bar{c}})$, and considers conflicted pairs to be non-matched according to a matching constraint (lines 14–16) through getConflictPairs. There are three types of ER [23]: Clean-Clean ER, Dirty-Clean ER and Dirty-Dirty ER. Our work focuses on Clean-Clean ER. It contains a matching constraint (i.e., each record $r_{1,i} \in D_1$ may matches one or zero record $r_{2,j} \in D_2$). And then, it resolves retained inconsistent pairs in $P^{\bar{c}}$ (lines 17–39) by evoking procedures doubleTopKNeighbor (line 20), compAERI (line 21), selectMaxAERI (line 24), isSatisCon (line 25), and relatedPairs (line 28) sequentially. It resolves at least one inconsistent pair in each iteration. It terminates when no inconsistent pair exists in $P^{\bar{c}}$. Finally, it outputs $\overline{M}$ and M.

Procedures. Now, we present related procedures as follows.

getConflictPairs Given R^p, $P^{\bar{c}}$, and I, it returns all conflicted pairs P^v. If records of p in $P^{\bar{c}}$, are involved in R^p, p is a conflicted pair.

doubleTopKNeighbor Given $p^{\bar{c}}$, k, $R^n \cup R^p$, and $\mathbb{H}$, it finds out the top k reference positive and negative pairs w.r.t. $p^{\bar{c}}$ (i.e., R_k^p and R_k^n), according to the Euclidean distances in $\mathbb{H}$. $\mathbb{H}$ is a key-value structure, which is computed via matrix op-

Algorithm 1: KIA

input : $P^{\bar{c}}$, k, α, Ratio $\mathbb{V}$, I
output: M, $\overline{M}$

1 $R \leftarrow$ getRefPairs(I, Ratio);
2 **for** $p_i \in R \cup P^{\bar{c}}$ **do**
3 **for** $p_j \in P^{\bar{c}}$ **do**
4 **if** $p_i \neq p_j$ **then**
5 $v(p_i) \leftarrow$ getPairVec($p_i, \mathbb{V}$);
6 $v(p_j) \leftarrow$ getPairVec($p_j, \mathbb{V}$);
7 $d \leftarrow \|v(p_i), v(p_j)\|$;
8 $\mathbb{H} \leftarrow \mathbb{H} \cup \{((p_i, p_j), d)\}$;
9 **end**
10 **end**
11 **end**
12 $R^n, R^p \leftarrow$ split(R) ;
13 $\mathbb{D} \leftarrow$ compDis($\mathbb{V}, R^p, P^{\bar{c}}$) ;
14 $P^v \leftarrow$ getConflictPairs($R^p, P^{\bar{c}}, I$) ;
15 $P^{\bar{c}} \leftarrow P^{\bar{c}} - P^v$;
16 $\overline{M} \leftarrow \overline{M} \cup P^v$;
17 **while** $P^{\bar{c}} \neq \emptyset$ **do**
18 $M_d \leftarrow \emptyset$;
19 **for** $p^{\bar{c}} \in P^{\bar{c}}$ **do**
20 $R_k^p, R_k^n \leftarrow$ doubleTopKNeighbor($p^{\bar{c}}, k, R^n \cup R^p, \mathbb{H}$) ;
21 $AERI \leftarrow$ compAERI($R_k^p, R_k^n, p^{\bar{c}}, \alpha, \mathbb{D}, I$) ;
22 $M_d \leftarrow M_d \cup \{(p^{\bar{c}}, AERI)\}$;
23 **end**
24 $p^*, AERI^* \leftarrow$ selectMaxAERI($M_d, P^{\bar{c}}$) ;
25 **if** $AERI^* > 0.5$ *and* isSatisCon(p^*, R^p) **then**
26 $M \leftarrow M \cup \{p^*\}$;
27 $R^p \leftarrow R^p \cup \{p^*\}$;
28 $C \leftarrow$ relatedPairs($p^*, P^{\bar{c}}$) ;
29 $\overline{M} \leftarrow \overline{M} \cup C$;
30 $R^n \leftarrow R^n \cup C$;
31 **else**
32 $\overline{M} \leftarrow \overline{M} \cup \{p^*\}$;
33 $R^n \leftarrow R^n \cup \{p^*\}$;
34 **end**
35 **if** $C \neq \emptyset$ **then**
36 $P^{\bar{c}} \leftarrow P^{\bar{c}} - C$;
37 **end**
38 $P^{\bar{c}} \leftarrow P^{\bar{c}} - \{p^*\}$;
39 **end**

erations and efficient in the query operation. Given p_i and p_j, their Euclidean distance is the value of the key (p_i, p_j) in $\mathbb{H}$.

compAERI Given R_k^p, R_k^n, $p^{\bar{c}}$, α, $\mathbb{D}$, and I, it is to compute $AERI(R_k^p \cup R_k^n, p^{\bar{c}})$ according to Eq. 4.

selectMaxAERI Given M_d which contains $AERI$ of inconsistent pairs in $P^{\bar{c}}$, it is to select an inconsistent pair p^*, which has the highest value of $AERI$ in $P^{\bar{c}}$ denoted by $AERI^*$.

isSatisCon Given p^* and R^p, if a pair p in R^p, involves any record of p^*, it returns $False$, and $True$ otherwise. $False$ means that it is unnecessary to deal with p^*, and p^* is non-matched, since one record of p^* has already achieved its match in p. $True$ means that there is no record involved in both R^p and p^*, so p^* needs to be processed.

relatedPairs Given p^* which is predicted to be matched, and all retained inconsistent pairs $P^{\bar{c}}$, it is to find out all related pairs throughout $P^{\bar{c}}$, that contain some overlapped record of p^*.

Example 4 As shown in Table 3, R^p is $\{p_{1,1}, p_{2,2}, p_{3,4}\}$, R^n is $\{p_{4,5}, p_{5,4}, p_{6,6}\}$, and $P^{\bar{c}}$ is $\{p_{7,7}, p_{8,8}, p_{9,8}, p_{8,10}, p_{12,11}, p_{13,13}\}$. After initialization, we get $\mathbb{H}$ and $P^v = \emptyset$. Consider $k = 3$ and $\alpha = 52.5\%$. In the first iteration, assuming that $p_{8,8}$ is with the highest value of $AERI$ (e.g., $AERI(R_3^p \cup R_3^n, p_{8,8}) = 0.58$), we can get $p^* = p_{8,8}$. As $p_{8,8}$ satisfies the constraint (isSatisCon$(p_{8,8}, R^p) = True$), we predict p^* to be matched. Then we compute related pairs w.r.t. p^* and get $C = \{p_{8,10}, p_{9,8}\}$, as compared with $p_{8,8} = (r_{1,8}, r_{2,8})$, $p_{8,10}$ contains the overlapped record $r_{1,8}$ and $p_{9,8}$ contains $r_{2,8}$. And then we predict pairs in C to be non-matched. Finally, we update $M = \{p_{8,8}\}$, $\overline{M} = \{p_{8,10}, p_{9,8}\}$, $R^p = \{p_{1,1}, p_{2,2}, p_{3,4}, p_{8,8}\}$, $R^n = \{p_{4,5}, p_{5,4}, p_{6,6}, p_{8,10}, p_{9,8}\}$, and $P^{\bar{c}} = \{p_{7,7}, p_{12,11}, p_{13,13}\}$. In the second iteration, assuming that $p_{7,7}$ is with the highest value of $AERI$ (e.g., $AERI(R_3^p \cup R_3^n, p_{7,7}) = 0.51$), we predict it to be matched according to $AERI(R_3^p \cup R_3^n, p_{7,7}) > 0.5$ and isSatisCon$(p_{7,7}, R^p) = True$. Because there is no conflict pair, we only update $M = \{p_{8,8}, p_{7,7}\}$, $R^p = \{p_{1,1}, p_{2,2}, p_{3,4}, p_{8,8}, p_{7,7}\}$ and $P^{\bar{c}} = \{p_{12,11}, p_{13,13}\}$. In the third iteration, assuming $p_{12,11}$ is with the highest value of $AERI$ (e.g., $AERI(R_3^p \cup R_3^n, p_{12,11}) = 0.49$), we predict it to be non-matched, since $AERI(R_3^p \cup R_3^n, p_{12,11}) \leq 0.5$. There is no necessary to compute the related pairs as it does not invoke any conflict. Finally, we update $\overline{M} = \{p_{9,8}, p_{8,10}, p_{12,11}\}$, $R^n = \{p_{4,5}, p_{5,4}, p_{6,6}, p_{8,10}, p_{9,8}, p_{12,11}\}$, and $P^{\bar{c}} = \{p_{13,13}\}$. In the final iteration, assuming $AERI(R_3^p \cup R_3^n, p_{13,13}) = 0.44$ and isSatisCon$(p_{13,13}, R^p) = True$, we predict it to be non-matched, and update $\overline{M} = \{p_{9,8}, p_{8,10}, p_{12,11}, p_{13,13}\}$, $R^n = \{p_{4,5}, p_{5,4}, p_{6,6}, p_{8,10}, p_{9,8}, p_{12,11}\}$, and $P^{\bar{c}} = \emptyset$. As $P^{\bar{c}}$ is empty, KIA terminates, and we get $M = \{p_{8,8}, p_{7,7}\}$ and $\overline{M} = \{p_{12,11}, p_{8,10}, p_{9,8}, p_{13,13}\}$ □

Theorem 1 Given inconsistent pairs $P^{\bar{c}}$ and reference pairs R, Algorithm KIA runs in $O(|R \cup P^{\bar{c}}| \cdot |P^{\bar{c}}| + |P^{\bar{c}}|^2)$ time.

Proof As shown in Algorithm 1, the time-consuming operations are (i) computing $AERI$ of $p^{\bar{c}}$, which mainly includes the distance calculation (line 7), and (ii) the search operation (line 20) which aims to find R_k^p and R_k^n. For ease of discus-

sion, we view each distance calculation as a cost unit, and each search operation as the same cost unit. According to lines 2–11, the distance calculation takes $O(|R \cup P^{\bar{c}}| \cdot |P^{\bar{c}}|)$ times. From lines 17–39, there are at most $|P^{\bar{c}}|$ iterations, as there is at least one inconsistent pair to be resolved during each iteration. The time required for line 20 is $O(|P^{\bar{c}}|^2)$. Thus, the time complexity of KIA is $O(|R \cup P^{\bar{c}}| \cdot |P^{\bar{c}}| + |P^{\bar{c}}|^2)$. As once an inconsistent pair is resolved to be matched, the related pairs are generated and $P^{\bar{c}}$ becomes smaller. In practice, the time for line 20 is far less than $|P^{\bar{c}}|^2$. □

5 Experimental Results

In this section, we present comprehensive experiments using four real-world data sets, to demonstrate the effectiveness in terms of Recall, Precision, and F-score and scalability of our strategy. We evaluated (i) the effectiveness of our incremental entity resolution approach (KIA), comparing with individual and baseline methods, (ii) the impact of different weights, the ratio of non-matched to matched pairs among reference pairs, the number of neighbor reference pairs w.r.t. inconsistent pairs, and (iii) the scalability of KIA with the data size.

5.1 Experimental Setting

Environment. We used ubuntu 16.04 64 bit as the operating system and python 2.7 as the runtime environment. All experiments were conducted on a machine with a 2.50GHz Intel(R) Core(TM) i7-4710MQ processor and 16 GB of RAM.

Evaluation Measures. We use Recall, Precision, and F-score to measure the effectiveness of our method. The ground truth is composed of all truly matched pairs, denoted as $pairs_T$. The positive pairs are those predicted to be matched, denoted as $pairs_P$. The truly positive pairs are kind of pairs, which are correctly predicted to be matched, denoted as $pairs_{TP}$.

Recall is the ratio of truly positive pairs to the truly matched pairs, i.e., $\text{Recall} = \frac{\#pairs_{TP}}{\#pairs_T}$.

Precision is the ratio of the truly positive pairs to the positive pairs, i.e., $\text{Precision} = \frac{\#pairs_{TP}}{\#pairs_P}$.

F-score, short for the balanced F-score, is the harmonic mean of Recall and Precision, i.e., $\text{F-score} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Precision} + \text{Recall}}$.

Notice that, we define the global (or local) evaluation measures (i.e., the aforementioned measures on the entire data (or inconsistent pairs)), so as to facilitate comparison. The main reason is that individual and baseline methods have different objectives. Individual methods aim to conduct an entity resolution task from scratch, so it is convenient to employ the global ones in comparison with these methods. While the baseline methods and KIA only resolve inconsistent pairs, so it is fair to evaluate their effectiveness on the local ones. We applied the local ones in most cases.

Data Sets. We conducted comprehensive experiments using four real-world data sets [1] in bibliographic or e-commerce domains, exploited in existing literature [2, 5, 23]. We describe these data sets as follows.

1. AB, short for Abt-Buy, consists of two relational tables (i.e., Abt and Buy). Both of them present elaborate product information (e.g., name, description, and so on). We only leveraged name and description.
2. AG, short for Amazon-GoogleProducts, consists of two relational tables (i.e., Amazon and GoogleProducts). Like AB, it contains product information, and two attributes (i.e., name and description) are employed.
3. DA, short for DBLP-ACM, consists of two relational tables (i.e., DBLP and ACM). Both of them are bibliographic and involve four attributes (i.e., title, authors, venue, and year). We only leveraged authors and title.
4. DS, short for DBLP-Scholar, consists of two relational tables (i.e., DBLP and Scholar). Like DA, it contains the same attributes and we only leveraged authors and title.

Algorithms. We compared our method with individual methods (i.e., distance-based [5], rule-based, and cluster-based [10] methods). The distance-based method (Distance) is a state-of-the-art entity resolution algorithm and we fortunately got its source code. Its basic idea is first to map each pair into a point of the feature space based on similarities on matching fields, and then compute the centrifuge distance of pairs, rank pairs by their distances, and finally regard pairs that satisfy the matching constraint as matched ones. We also generalize the record-matching rule (RR) proposed by [4], through assigning a weight to each record matching rule, denoted as gRR. These weights are specified by domain experts in advance to quantify the importance of gRR. We implemented a rule-based method (Rule), which computes the summed weight of gRRs that each candidate pair follows, and then judges pairs that satisfy the matching constraint as matched ones. The cluster-based method (Cluster) is offered by Febrl [10]. Its basic idea is to convert each pair into a vector using field comparison functions, and then exploit a K-Means

[1] These data sets are available at `http://dbs.uni-leipzig.de/en/research/projects/object_matching/fever/benchmark_datasets_for_entity_resolution`.

Table 5 Comparison with Individual Methods (top values are shown in bold)

Method	Recall(%)				Precision(%)				F-score(%)			
	AB	AG	DA	DS	AB	AG	DA	DS	AB	AG	DA	DS
Distance	**79.03**	**64.23**	98.02	95.36	80.73	61.31	95.78	87.61	79.87	**62.77**	96.90	91.32
Rule	74.84	58.15	97.84	93.15	76.73	59.06	95.52	85.58	75.77	58.60	96.67	89.21
Cluster	34.55	26.69	97.75	87.71	33.81	23.32	85.39	65.75	34.17	24.89	91.15	75.16
KIA	78.21	61.38	**98.29**	**96.04**	**88.00**	**63.99**	**97.81**	**91.64**	**82.82**	62.66	**98.05**	**93.79**

method for clustering pairs into match/non-match groups.

We also compared with three baseline algorithms: the majority voting strategy (MVote), the weighted voting strategy (WVote), and a learning-based approach (Learning). MVote [8] is the most popular method. It is to calculate the voting number of a pair, and makes a conclusion that a pair is matched if the pair obtains more than half of the entire votes. WVote takes the voting weight into consideration. It prefers to extract matched pairs with the highest summed weight of votes. Learning uses the support vector classifier (SVC). It first regards consistent pairs as the training set, and inconsistent pairs as the test set, and then classifies inconsistent pairs into match/non-match groups. As AB, AG, DA and DS are Clean-Clean data sets, for fair comparison, we impose the matching constraint on the output of baseline methods.

5.2 Experimental Results

We performed six sets of experiments and reported the experimental results: (i) in the first set of experiments, we presented the effectiveness of our approach KIA, comparing to individual methods (i.e., Distance, Rule and Cluster); (ii) we evaluated our algorithm, comparing with traditional baseline approaches (i.e., MVote, WVote and Learning) in the second set of experiments; (iii) in the third set of experiments, we analyzed the influence of different weights on the effectiveness of KIA; (iv) we also assessed the effect of the neighbor number k (resp. the ratio of non-matched to matched pairs among reference pairs Ratio) on the quality of our approach in the fourth (resp. fifth) set of experiments; and (v) in the last set of experiments, we presented the scalability of KIA with both the number of reference pairs and that of inconsistent pairs.

5.2.1 Comparison with Individual Methods

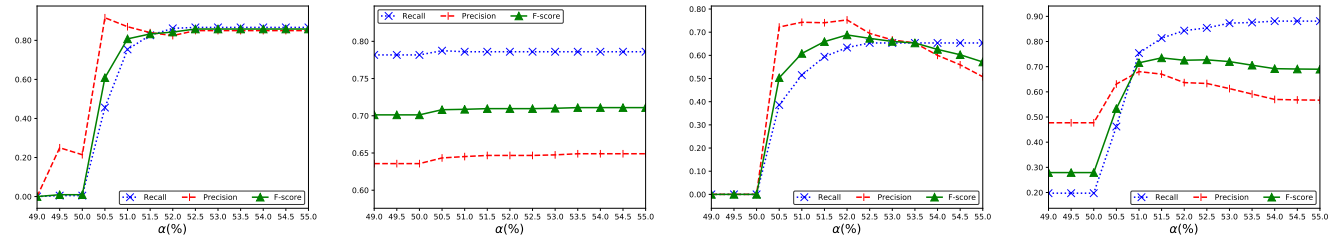
Among the first set of experiments, we show that our approach KIA can improve the quality of individual methods, by taking reference pairs R into consideration. For KIA, we assigned 1 to Ratio, 3 to k, 52.5% to α, and leveraged three individual methods. The experimental results on AB, AG, DA and DS have been reported in Table 5. We make the following observations. (i) KIA can improve Precision, with a slight decline in Recall or F-score on occasion. For AB, Precision of KIA was improved by 54.19%, 11.27% and 7.27%, compared with that of Cluster, Rule and Distance. On AG, Precision of KIA was 40.67%, 4.93% and 2.68% more than that of Cluster, Rule and Distance respectively. On DA, the improvement reached 12.42%, 2.29% and 2.03% in terms of Precision. Similarly for DS, the improvement reached 25.89%, 6.06% and 4.03%. Compared with Recall and F-score of Distance, those of KIA slightly decreased over AG, because the entity match task on AG is quite challenging. The improvement on DA is limited because all of individual methods on DA have already achieved very high Precision, which verifies KIA can maintain the results of individual methods. (ii) KIA can improve individual algorithms in terms of F-score and Recall in most cases, even in contrast to Distance with the best effectiveness among individual methods. Although there is slight decline of Recall over AB in contrast to that of Distance, F-score of KIA is still improved. All in all, KIA not only significantly outperforms individual methods in terms of Precision, but also improves their F-score and Recall in most cases.

5.2.2 Comparison with Baseline Methods

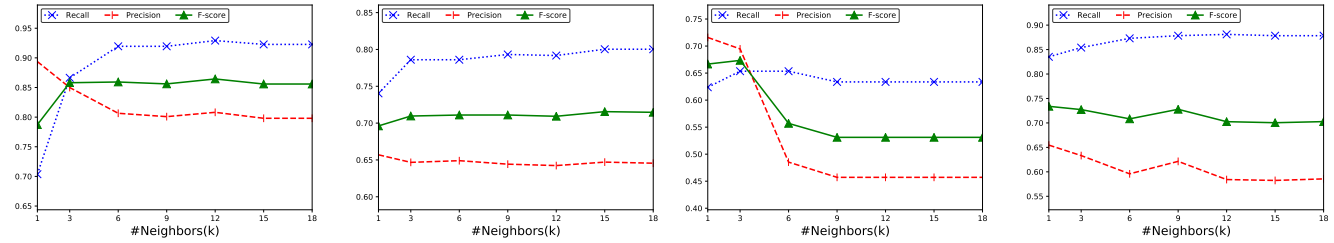
To the best of our knowledge, we are the first to study how to combine off-the-shelf methods to enhance their performance via an unified framework for entity resolution. So, in the second set of experiments, we considered MVote, WVote and Learning as baseline methods, and contrasted their effectiveness with that of KIA. The experimental results on AB, AG, DA and DS, as shown in Table 6, have uncovered the following. (i) KIA significantly outperforms these baseline methods in either Precision or Recall. Compared to Precision of MVote, WVote and Learning, that of KIA on AB (resp. DA) increased by 2.5% (resp. 14.27%), 5.21% (resp. 7.74%), and 28.21% (resp. 32.22%). Similarly, for AG (resp. DS), Recall of KIA increased by 7.84% (resp. 4.87%), 9.98% (resp. 11.36%), and 76.75% (resp. 8.92%). (ii) KIA may occasionally have a lower Recall or Precision than baseline methods. For instance, on DA, Learning had a higher Recall,

Table 6 Comparison with Baseline Methods (top values are shown in bold)

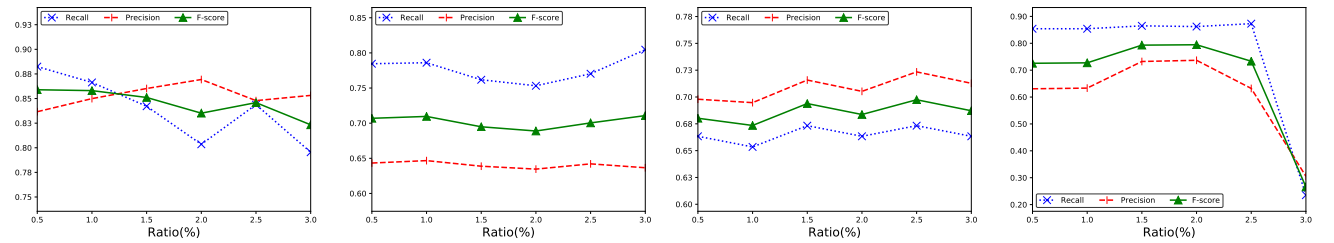
Method	Recall(%)				Precision(%)				F-score(%)			
	AB	AG	DA	DS	AB	AG	DA	DS	AB	AG	DA	DS
MVote	84.22	70.76	68.32	80.54	82.49	**66.58**	55.20	61.57	83.35	68.60	61.06	69.79
WVote	81.32	68.62	49.50	74.05	79.78	63.04	61.73	**68.33**	80.54	65.71	54.95	71.08
Learning	68.12	1.85	**75.25**	76.49	56.78	19.70	37.25	38.66	61.93	3.39	49.84	51.36
KIA	**86.63**	**78.60**	65.35	**85.41**	**84.99**	64.67	**69.47**	63.33	**85.81**	**70.96**	**67.35**	**72.73**



(a) Effect of Different Weights on AB (b) Effect of Different Weights on AG (c) Effect of Different Weights on DA (d) Effect of Different Weights on DS



(e) Effect of #Neighbors (k) on AB (f) Effect of #Neighbors (k) on AG (g) Effect of #Neighbors (k) on DA (h) Effect of #Neighbors (k) on DS



(i) Effect of Ratio on AB (j) Effect of Ratio on AG (k) Effect of Ratio on DA (l) Effect of Ratio on DS

Fig. 2 Effect of Different Weights (α), #Neighbors (k) and Ratio

while on AG, MVote had a higher Precision. However, all of F-score on data sets increased.

5.2.3 Effect of Different Weights

We conducted a set of experiments to analyze the effect of different weights on KIA, and reported the results in Figs. 2(a)–2(d). We varied the weight α from 49.0% to 55.0%. We observed the following. (i) On most of data sets, the effectiveness of our method increased as α increased. As Figs. 2(a), 2(c) and 2(d) shown, Recall, Precision and F-score increased in the beginning. However, for AG they went smoothly with subtle variations in Fig. 2(b). (ii) When α went around 52.5%, our method achieved a better performance in most cases, and we regarded it as our default setting. For example, on DA and DS, our method achieved better results with α = 52.5%, and there were decreases in terms of Precision and F-score, when α was greater than 52.5% in Figs. 2(c) and 2(d).

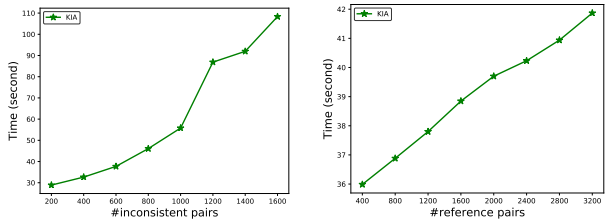
5.2.4 Effect of the Neighbor Number

In this set of experiments, we evaluated the impact of the neighbor number w.r.t. an inconsistent pair on our proposed method (KIA). As Figs. 2(e)–2(h) described, with the neighbor number (k) varying from 1 to 18, the results of KIA show the following observations. (i) Experimentally, setting k to 3 is better than others, and we adopted it as our default setting. There are two reasons for this. Firstly, letting k = 1 is unstable. F-scores of AB, AG, and DA in the setting with k = 1,

were lower than those in other settings (e.g., k = 3) , while F-score of DS with k = 1 was higher than that of k = 3. Secondly, other settings (e.g., k = 12 on AB) may obtain certain better F-score. However, their increments are quite limited, and the computation cost is higher than that of the case with k = 3, as the more neighbors mean that the more calculations are required. (ii) Utilizing more neighbors does not necessarily bring more remarkable benefits, and sometimes evaluation measures may decrease as k increases. For instance, Fig 2(e) and 2(f) demonstrated that the measures (Recall, Precision, and F-score) of KIA, might increase slightly, but did not obtain obvious improvement, with k varying from 3 to 18. In some case, both F-score and Precision on DA (or DS) had slight decline as shown in Figs. 2(g) and 2(h).

5.2.5 Effect of different Ratio

In this set of experiments, we evaluated the impact of Ratio on KIA and varied it from from 0.5 to 3.0. As Figs. 2(i)–2(l) illustrated, the experimental results reveal the following. (i) All of measures (e.g., Recall) vary markedly on different data set. For example, on DS, they first increased as ratio increased, and then decreased quickly, while on AG, they went smoothly. (ii) KIA with Ratio = 1 achieves better effectiveness for most data sets, and Ratio = 1 was our default setting. The reason was the following. Firstly, when Ratio was 1, KIA always performed as well as Ratio = 0.5, but R was balanced. Secondly, other settings (e.g., Ratio = 2.5 on DA) may bring some improvement, but they also incur additional costs such as more distance computation.



(a) Different $|P^{\bar{c}}|$ (|R| = 600) (b) Different |R| ($|P^{\bar{c}}|$=600)

Fig. 3 Scalability on DS

5.2.6 Scalability Evaluation

The last set of experiments showed the scalability of KIA with both the number of reference pairs, denoted as $|R|$, and the number of inconsistent pairs, denoted as $|P^{\bar{c}}|$. In the first case, we fixed $|R|$ = 600, and varied $|P^{\bar{c}}|$ from 200 to 1,600, as shown in Fig 3(a). In the second case, we fixed $|P^{\bar{c}}|$ = 600, and varied $|R|$ from 400 to 3,200, as shown in Fig 3(b). The scalability evaluation on DS shows the following. (i) Our approach scales quadratically with the number of inconsistent pairs. (ii) Our approach approximately scales linearly with the number of reference pairs. Once an inconsistent pair is resolved, some related pairs are removed, and the execution time is reduced. However, the number of related pairs depends on $P^{\bar{c}}$ and R, so there are some variations in Figs. 3(a)–3(b). All of observations are consistent with Theorem 1.

6 Conclusion and Future work

In this paper, we made the first step to combine a variety of individual methods to improve the efficiency of ER tasks. We proposed an unified generic framework to employ the consistent pairs, in which we implemented an incremental entity reconciliation algorithm to address all inconsistent pairs. The extensive experiments have shown that our method outperforms individual methods and baseline methods. Notice that, our method is suitable for the case where quality is more urgent than efficiency, because it achieves higher quality at the expense of a little efficiency.

For future work, we would like to extend our approach to support more kind of constraints and provide more efficient algorithms. We also want to extend the proposed method to meet the need of some applications that cover multiple types of entities.

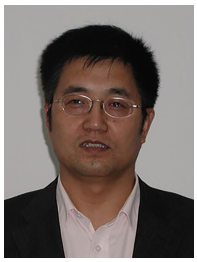
Acknowledgements We thank Murtadha Ahmed, Yiyi Li, Ping Zhong, YanyanWang, and Jing Su for their invaluable suggestions. This work was supported by the Ministry of Science and Technology of china under Grant No.2016YFB1000703, and the National Natural Science Foundation of China under Grant No.61732014, No.61332006, No.61472321, No.61502390, and No.61672432.

References

1. Pradap Konda, Sanjib Das, Paul Suganthan G. C., AnHai Doan, Adel Ardalan, Jeffrey R. Ballard, Han Li, Fatemah Panahi, Haojun Zhang, Jeffrey F. Naughton, Shishir Prasad, Ganesh Krishnan, Rohit Deep, and Vijay Raghavendra. Magellan: Toward building entity matching management systems. *PVLDB*, 9(12):1197–1208, 2016.
2. Giovanni Simonini, Sonia Bergamaschi, and H. V. Jagadish. BLAST: a loosely schema-aware meta-blocking approach for entity resolution. *PVLDB*, 9(12):1173–1184, 2016.
3. Lingli Li, Jianzhong Li, and Hong Gao. Rule-based method for entity resolution. *IEEE Trans. Knowl. Data Eng.*, 27(1):250–263, 2015.
4. Jiannan Wang, Guoliang Li, Jeffrey Xu Yu, and Jianhua Feng. Entity matching: How similar is similar. *PVLDB*, 4(10):622–633, 2011.

5. Fengfeng Fan, Zhanhuai Li, Qun Chen, and Hailong Liu. An outlier-detection based approach for automatic entity matching. *Chinese Journal of Computers*, 40(10):2197–2211, 2017.
6. Sudipto Guha, Nick Koudas, Amit Marathe, and Divesh Srivastava. Merging the results of approximate match operations. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*, pages 636–647, 2004.
7. Debabrata Dey, Sumit Sarkar, and Prabuddha De. Entity matching in heterogeneous databases: A distance based decision model. In *Thirty-First Annual Hawaii International Conference on System Sciences, Kohala Coast, Hawaii, USA, January 6-9, 1998*, pages 305–313, 1998.
8. Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.
9. Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.
10. Peter Christen. Febrl: A freely available record linkage system with a graphical user interface. In *Proceedings of the Second Australasian Workshop on Health Data and Knowledge Management - Volume 80*, HDKM '08, pages 17–25, Darlinghurst, Australia, Australia, 2008. Australian Computer Society, Inc.
11. Steven Euijong Whang and Hector Garcia-Molina. Joint entity resolution on multiple datasets. *VLDB J.*, 22(6):773–795, 2013.
12. Hanna Köpcke, Andreas Thor, and Erhard Rahm. Evaluation of entity resolution approaches on real-world match problems. *PVLDB*, 3(1):484–493, 2010.
13. Chenchen Sun, Derong Shen, Yue Kou, Tiezheng Nie, and Ge Yu. GBJER: A graph-based model for joint entity resolution. In *Database Systems for Advanced Applications - 20th International Conference, DASFAA 2015, Hanoi, Vietnam, April 20-23, 2015, Proceedings, Part I*, pages 458–473, 2015.
14. Chengliang Chai, Guoliang Li, Jian Li, Dong Deng, and Jianhua Feng. Cost-effective crowdsourced entity resolution: A partial-order approach. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 969–984, 2016.
15. Zhaoqiang Chen, Qun Chen, and Zhanhuai Li. A human-and-machine cooperative framework for entity resolution with quality guarantees. In *33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017*, pages 1405–1406, 2017.
16. Wenfei Fan, Xibei Jia, Jianzhong Li, and Shuai Ma. Reasoning about record matching rules. *PVLDB*, 2(1):407–418, 2009.
17. Zhi-Hua Zhou. *Machine learning*. Tsinghua University Press, 2016.
18. Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.
19. Hanna Köpcke, Andreas Thor, and Erhard Rahm. Comparative evaluation of entity resolution approaches with FEVER. *PVLDB*, 2(2):1574–1577, 2009.
20. J.D. Rajaraman, A.; Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2011.
21. Tom M Mitchell et al. Machine learning. wcb, 1997.
22. Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alexander J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.
23. George Papadakis, Georgia Koutrika, Themis Palpanas, and Wolfgang Nejdl. Meta-blocking: Taking entity resolutionto the next level. *IEEE Trans. Knowl. Data Eng.*, 26(8):1946–1960, 2014.

Yaoli XU is a PhD candidate at the School of Computer Science and Engineering, Northwestern Polytechnical University (NWPU), China. She received her M.S. degree from NWPU. Her research interests include data quality, data cleaning and entity resolution.



Zhanhuai LI is a professor and doctoral supervisor at the School of Computer Science and Engineering, Northwestern Polytechnical University (NWPU), China. He is a senior member of China Computer Federation. His research interests include data management, big data analysis and data quality.

Qun CHEN is a professor and doctoral supervisor at the School of Computer Science and Engineering, Northwestern Polytechnical University, China. He received his PhD degree from National University of Singapore. His research interests include data management, big data analysis and data quality.

Fengfeng FAN is a PhD candidate at the School of Computer Science and Engineering, Northwestern Polytechnical University (NWPU), China. He received his B.S. and M.S. degrees from Northwestern Polytechnical University. His research interests include data quality and entity resolution.