SCHOOL OF COMPUTER SCIENCE, NORTHWESTERN POLYTECHNICAL UNIVERSITY

r-HUMO: A Risk-aware Human-Machine Cooperation Framework for Entity Resolution with Quality Guarantees (Technical Report)

Boyi Hou, Qun Chen, Zhaoqiang Chen, Youcef Nafa, Zhanhuai Li

Oct 6, 2018

Even though many approaches have been proposed for entity resolution (ER), it remains very challenging to enforce quality guarantees. To this end, we propose a *r*isk-aware HUman-Machine cOoperation framework for ER, denoted by *r*-HUMO. Built on the existing HUMO framework, *r*-HUMO similarly enforces both precision and recall guarantees by partitioning an ER workload between the human and the machine. However, *r*-HUMO is the first solution that optimizes the process of human workload selection from a risk perspective. It iteratively selects human workload by real-time risk analysis based on the human-labeled results as well as the pre-specified machine metric. In this paper, we first introduce the *r*-HUMO framework and then present the risk model to prioritize the instances for manual inspection. Finally, we empirically evaluate *r*-HUMO's performance on real data. Our extensive experiments show that *r*-HUMO is effective in enforcing quality guarantees, and compared with the state-of-the-art alternatives, it can achieve desired quality control with reduced human cost.

1 INTRODUCTION

Entity resolution (ER) usually refers to identifying the relational records that correspond to the same real-world entity. A challenging task due to incomplete and dirty data, ER has been extensively studied in the literature [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. Unfortunately, it remains very challenging to enforce quality guarantees on ER. The approach based on active learning [12, 13] can maximize recall while ensuring a pre-specified precision level. More recently, a HUman-Machine cOoperation framework [14, 15], denoted by HUMO, has been proposed to enforce more comprehensive quality guarantees at both precision and recall fronts. HUMO enables a flexible mechanism for quality control by partitioning an ER workload between the human and the machine. It automatically labels easy instances by the machine while assigning more challenging ones to the human. For instance, given a metric of record pair similarity, the pairs with high or low similarities can be automatically labeled by the machine with high accuracy. However, the pairs with medium similarities may require human inspection because labeling them either way by machine would introduce considerable errors. The optimization objective of HUMO is to minimize the required human cost given the user-specified precision and recall levels.

HUMO measures the hardness of an ER instance pair by a pre-specified machine metric and performs human workload selection in batch mode. It first groups the pairs into subsets by their metric values and then assigns the subsets between the human and the machine. As a result, all the pairs with similar metric values in a subset would be either automatically labeled by the machine or manually labeled by the human. However, it can be observed that due to the limitation of machine metrics, even though two pairs have similar metric values, their risks of being mislabeled by the machine may be vastly different.

In this paper, we investigate the problem of workload partition between the human and the machine from a risk perspective. Since human workload selection can be performed in an interactive manner, human input, which consists of the human-labeled pairs in our example, can be naturally used for risk analysis to prioritize pairs for human inspection. Our idea is to iteratively pick up more risky pairs from a given subset of pairs for human inspection such that the remaining pairs in the subset can achieve overall higher machine-labeling accuracy. With human effort spent on more risky pairs, the required human cost for quality guarantees can be effectively reduced. As HUMO, the proposed risk-aware framework, *r*-HUMO, is to some extent motivated by the success of the existing crowdsourcing solutions for ER [16, 17, 18, 19]. The work on crowdsourcing ER focused on how to make the human work effectively and efficiently on a given workload. HUMO and *r*-HUMO instead investigate how to partition a workload between the human and the machine such that a user-specified quality requirement can be met.

The major contributions of this paper can be summarized as follows:

1. We propose a risk-aware human-machine cooperation framework for ER, *r*-HUMO, which can enforce both precision and recall guarantees. It is the first solution that optimizes the process of human workload selection from a risk perspective.

2. We propose the technique of risk analysis for iterative human workload selection. We present the risk model based on modern portfolio investment theory to prioritize ER pairs for human inspection;

3. We conduct an empirical study on the performance of *r*-HUMO by extensive experiments on real data. Our experimental results show that *r*-HUMO is effective in enforcing quality guarantees, and compared with the state-of-the-art alternatives, it can achieve desired quality control with reduced human cost.

The rest of this paper is organized as follows: Section 2 reviews more related work. Section 3 introduces the problem and briefly describes the existing HUMO framework. Section 4 presents the *r*-HUMO framework. Section 5 describes the technique of risk analysis. Section 6 presents our empirical evaluation results. Finally, Section 7 concludes this paper with some thoughts on future work.

2 Related Work

As a classical problem in the area of data quality, entity resolution has been extensively studied in the literature [1, 2]. It can be performed based on rules [4, 5, 6], probabilistic theory [3, 20] or machine learning [7, 8, 12, 13]. Unfortunately, it remains very challenging to enforce quality guarantees on ER.

The approach based on active learning [12, 13] has been proposed to enforce the precision guarantee on ER. The authors of [12] proposed a technique that can optimize recall while ensuring a pre-specified precision level. The authors of [13] proposed an improved algorithm to approximately maximize recall under the precision constraint. Compared with the work of [12], its major advantage is better label complexity. However, these techniques share the same classification paradigm with the traditional machine learning algorithms; hence they can not enforce comprehensive quality guarantees specified at both precision and recall fronts.

The progressive paradigm for ER [21, 22] has also been proposed for the application scenario in which ER should be processed efficiently but does not necessarily require to generate high-quality results. Taking a pay-as-you-go approach, it studied how to maximize quality given a pre-specified resolution budget. In [21], the authors proposed several concrete ways of constructing resolution "hints" that can then be used by a variety of existing ER algorithms as a guidance for which entities to resolve first. In [22], the authors studied the more complicated problem of relational ER, in which a resolution of some entities might influence the resolution of other entities. A similar iterative algorithm, SiGMa, was proposed in [23]. It can leverage both the structure information and the string similarity measures to resolve entity alignment across different knowledge bases. There also exist some interactive systems [26, 34] that take advantage of knowledge bases or specific user input to achieve improved efficiency and quality for data cleaning. Unfortunately, these techniques have been built on machine computation; hence they can not be applied to enforce quality guarantees either.

It has been well recognized that pure machine algorithms may not be able to produce satisfactory results in many practical scenarios [17]. Therefore, many researchers [9, 16, 18, 19, 24, 25, 26, 27, 28, 29, 30, 31] have studied how to crowdsource an ER workload. In [18], the authors studied how to generate Human Intelligence Tasks (HIT), and how to incrementally select the instance pairs for human inspection such that the required human cost can be minimized. In [28], the authors focused on how to select the most beneficial questions for the human in terms of expected accuracy. More recently, the authors of [16] proposed a cost-effective framework that employs the partial order relationship on instance pairs to reduce the number of asked pairs. Similarly, the authors in [31] provided a solution to take advantage of both pairwise and multi-item interfaces in a crowdsourcing setting. The authors of [32] studied how to balance cost and quality in crowdsourcing. Considering the diverse accuracies of workers across tasks, the authors of [33] proposed an adaptive crowdsourcing framework that assigns the tasks based on worker accuracy estimation. In [27], the authors proposed an online crowdsourcing platform based on oracle. While these researchers addressed the challenges specific to crowdsourcing, we instead investigate a different problem in this paper: how to partition a workload between the human and the machine such that a user-specified quality requirement can be met. Since the workload assigned to the human by *r*-HUMO can be naturally processed by crowdsourcing, our work can be considered orthogonal to the existing work on crowdsourcing. It is interesting to investigate how to seamlessly integrate a crowdsourcing platform into *r*-HUMO in future work.

The *r*-HUMO framework is built on the recently proposed HUMO framework [14, 15], which can enforce quality guarantees at both precision and recall fronts. The general idea of HUMO and *r*-HUMO was similar to the Fellegi-Sunter theory of record linking [3], which also proposed to divide an ER workload into three parts based on match probability. HUMO however proposed the effective algorithms to divide an ER workload and estimate the match probability of machine workload for the quality guarantees specified at both precision and recall fronts. The *r*-HUMO framework represents a major step forward in that it is the first solution to optimize the process of human workload selection from a risk perspective. Instead of selecting human workload in batch mode purely based on a pre-specified machine metric as HUMO does, *r*-HUMO performs real-time risk analysis on the manually labeled results for the purpose of reducing the required human cost.

3 Preliminaries

3.1 Problem Definition

Entity resolution reasons about whether two records are equivalent. Two records are deemed to be equivalent if and only if they correspond to the same real-world entity. We call a pair an *equivalent* pair if and only its two records are equivalent; otherwise, it is called an *inequivalent* pair. An ER solution labels each pair in a workload as *matching* or *unmatching*. As usual, we measure the quality of an ER solution by the metrics of precision and recall. Precision denotes the fraction of equivalent pairs among the pairs labeled as *matching*, while recall denotes the fraction of correctly labeled equivalent pairs among all the equivalent pairs.

Formally, we denote the ground-truth labeling solution of D by $\hat{L}$, $\hat{L} = \{\hat{l}_1, \hat{l}_2, \cdots, \hat{l}_n\}$, in which $\hat{l}_i = 1$ if the corresponding records in the pair of d_i are equivalent and $\hat{l}_i = 0$ otherwise. Given a labeling solution L, we use D_{tp} to denote its set of true positive pairs, $D_{tp} = \{d_i | \hat{l}_i = 1 \wedge l_i = 1\}$, D_{fp} its set of false positive pairs, $D_{fp} = \{d_i | \hat{l}_i = 0 \wedge l_i = 1\}$, and D_{fn} its set of false negative pairs, $D_{fn} = \{d_i | \hat{l}_i = 1 \wedge l_i = 0\}$. Based on the definitions of D_{tp}, D_{fp} and D_{fn}, the achieved precision level of L can be represented by

$$\textit{precision(D,L)} = \frac{|D_{tp}|}{|D_{tp}| + |D_{fp}|}. \quad (1)$$

Similarly, the achieved recall level of L can be represented by

$$\textit{recall(D,L)} = \frac{|D_{tp}|}{|D_{tp}| + |D_{fn}|}. \quad (2)$$

Table 1: Frequently Used Notations.

Notation	Description
D	an ER workload consisting of instance pairs
D_i, D_+, D_-, D_H	subsets of D
S, S_i	a labeling solution for D
d, d_i	an instance pair in D
$TN(D_i)$	the total number of pairs in D_i
$EN(D_i)$	the number of equivalent pairs in D_i
$EP(D_i)$	the proportion of equivalent pairs in D_i
f, f_i	a feature of instance pair
F, F_i	a feature set
D_f	the set of instance pairs with the feature f

For presentation simplicity, we summarize the frequently used notations in Table 1. Formally, we define the problem of entity resolution with quality guarantees [14, 15] as follows:

Definition 1 ***[Entity Resolution with Quality Guarantees].*** *Given a set of instance pairs, $D = \{d_1, d_2, \cdots, d_n\}$, the problem of entity resolution with quality guarantees is to give a labeling solution S for D such that with the confidence level of θ, $precision(D, S) \geq \alpha$ and $recall(D, S) \geq \beta$, in which α and β denote the user-specified precision and recall levels respectively.*

3.2 THE HUMO FRAMEWORK

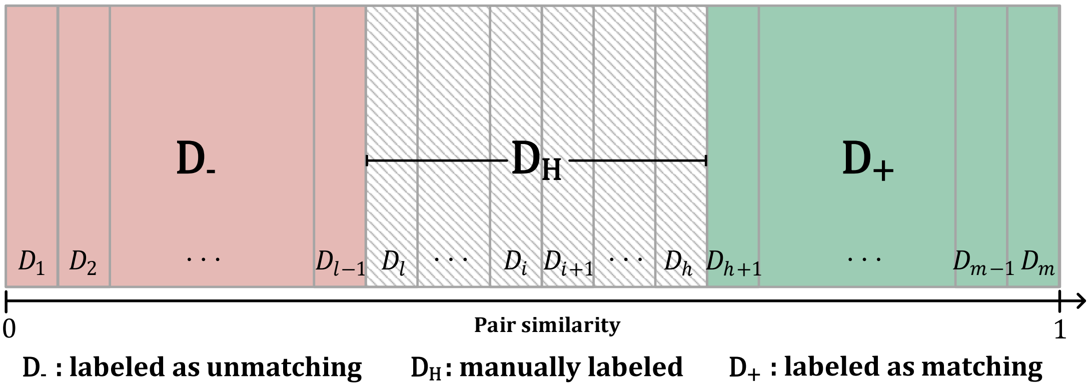


Figure 1: HUMO Framework.

The HUMO framework is shown in Fig. 1. Given a workload, D, HUMO first groups its pairs into unit subsets (denoted by D_i in the figure) by a machine metric (e.g., pair similarity or match probability), and then partitions the unit subsets into three disjoint sets, D_-, D_H and D_+. HUMO assumes that the given machine metric satisfies the monotonicity of precision, which statistically states that the higher (or lower) metric values a set of pairs have, the more probably they are equivalent pairs. HUMO enforces the precision and recall guarantees by automatically labeling D_- and D_+ as *unmatching* and *matching* respectively, and assigning D_H to the human for manual inspection. The monotonicity assumption of precision underlies the effectiveness of HUMO's workload partitioning strategy between the human and the machine. However, HUMO never need to expect that the monotonicity assumption can be *strictly* satisfied on real data. Instead, it only assumes that provided with a reasonable machine metric, monotonicity of precision is usually a statistical trend on real data. It is also worthy to point out that in a similar way, the monotonicity assumption of precision underlies the effectiveness of the existing machine classification metrics for ER. HUMO is effective provided that the given machine metric satisfies the monotonicity assumption of precision. However, for presentation simplicity, we use pair similarity as the example of machine metric in this paper.

Given an ER workload, D, the quality of a HUMO solution, S, can be estimated by reasoning about the lower and upper bounds of the number of equivalent pairs in D_-, D_H and D_+. In Figure. 1, the lower bound of the achieved precision level can be represented by

$$precision_L(D,S) = \frac{EN_L(D_+) + EN_L(D_H)}{TN(D_+) + TN(D_H)}, \tag{3}$$

in which $TN(\cdot)$ denotes the total number of pairs in a set and $EN_L(\cdot)$ denotes the lower bound of the total number of equivalent pairs in a set. Similarly, the lower bound of the achieved recall level can be represented by

$$recall_L(D,S) = \frac{EN_L(D_+) + EN_L(D_H)}{EN_L(D_+) + EN_L(D_H) + EN_U(D_-)}, \tag{4}$$

in which $EN_U(\cdot)$ denotes the upper bound of the total number of equivalent pairs in a set. *In this paper, for the sake of presentation simplicity, we assume that the pairs in D_H can be manually labeled with 100% accuracy. However, it is worthy to point out that the effectiveness of HUMO does not depend on the 100%-accuracy assumption. It can actually work properly provided that quality guarantees can be enforced on D_H. In the case that human errors are introduced in D_H, the lower bounds of the achieved precision and recall can be estimated based on Eq. 3 and Eq. 4 respectively.* Nonetheless, under the assumption that the human performs better than the machine in resolution quality, the best quality guarantees HUMO can achieve are no better than the performance of the human on D_H.

As human work is usually more expensive than machine computation, HUMO aims to minimize the workload in D_H while guaranteeing resolution quality. By quantifying human cost by the number of instance pairs in D_H, we define the optimization problem of HUMO as follows [14, 15]:

Definition 2 ***[Minimizing Human Cost in HUMO].*** *Given an ER workload, D, a confidence level θ, a precision level α and a recall level β, the optimization problem of HUMO is represented by*

$$\begin{aligned} &\arg\min_S |D_H(S)| \\ &subject \quad to \quad P(precision(D,S) \geq \alpha) \geq \theta, \\ &\qquad\qquad\qquad P(recall(D,S) \geq \beta) \geq \theta, \end{aligned} \tag{5}$$

in which S denotes a labeling solution, $D_H(S)$ denotes the set of instance pairs assigned to the human by S, $precision(D, S)$ denotes the achieved precision level of S, and $recall(D, S)$ denotes the achieved recall level of S.

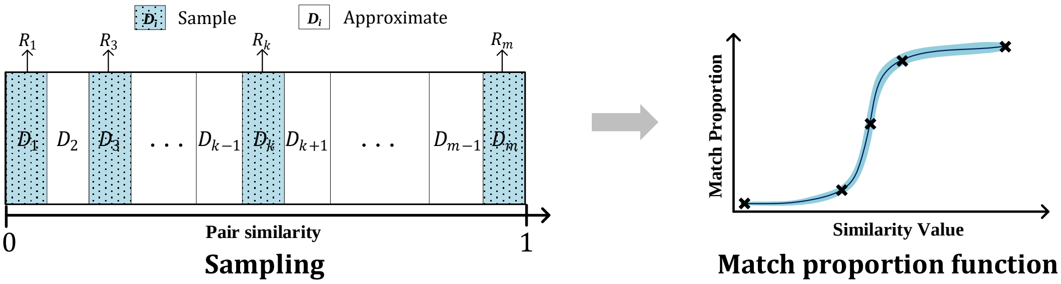


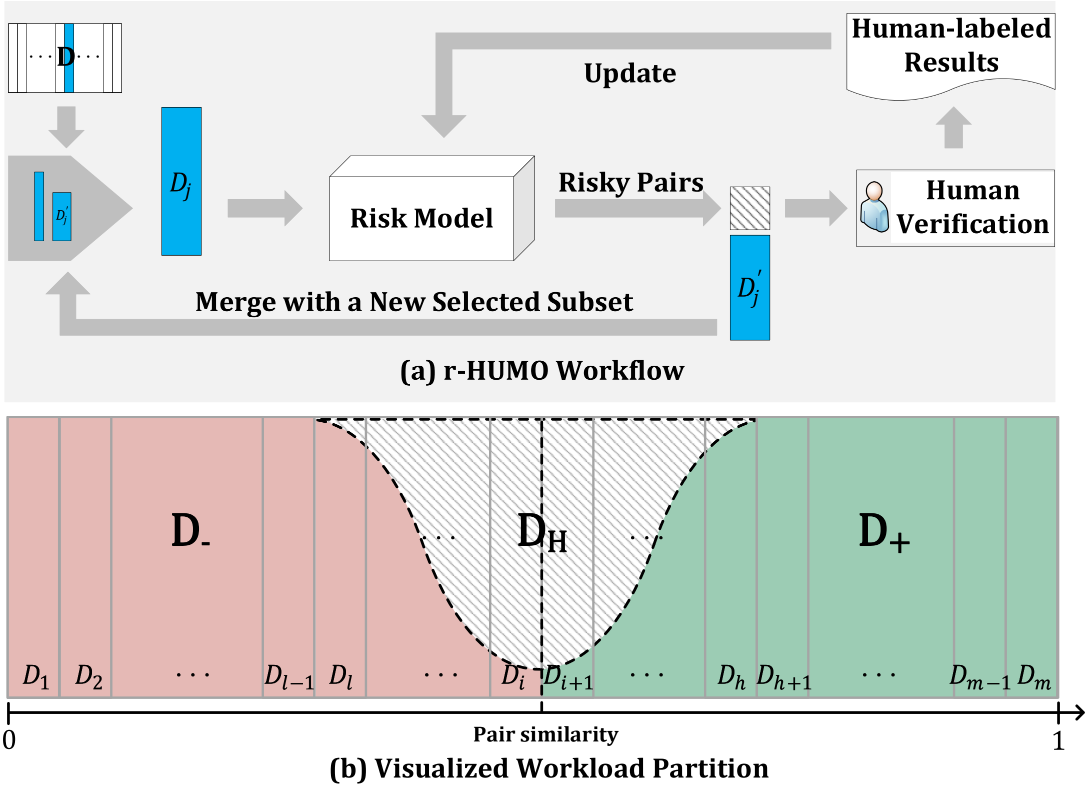
Figure 2: Process of GPR.

The optimization problem as defined in Eq. 5 is challenging because the proportions of equivalent pairs in D_+ and D_- are unknown, thus need to be estimated. There exist two types of approaches to minimize the size of D_H: one purely based on the monotonicity assumption of precision and the other one based on sampling [14, 15]. They estimate equivalence proportion based on different assumptions. Between them, the sampling-based approach has been empirically shown to have superior performance. It first estimates the equivalence proportions of the unit subsets by sampling, and then identifies the minimal workload of D_H by reasoning about the numbers of equivalent pairs in D_- and D_+. The equivalence proportion of a unit subset, D_i, can be directly estimated by sampling or approximated by Gaussian Process Regression (GPR) [35]. The process of GPR is shown in Fig. 2. Assuming that the equivalence proportions of all the unit subsets have a joint Gaussian distribution, GPR can approximate their equivalence proportions by sampling only a fraction of them. Based on GPR approximation, HUMO estimates the lower and upper bounds of the numbers of equivalent pairs in D_- and D_+ by aggregating their corresponding Gaussian distributions. It can therefore iteratively optimize the lower and upper bounds of D_H. More technical details of HUMO can be found at [14, 15].

4 THE *r*-HUMO FRAMEWORK

The *r*-HUMO framework consists of two processes, human workload selection and risk analysis. The process of human workload selection picks out the pairs for manual inspection from a set of candidate pairs; the process of risk analysis estimates pair risk based on the human-labeled results. The procedure is invoked iteratively until the user-specified quality requirement is met. After each iteration, the set of candidate pairs is updated and pair risk is also re-estimated based on the updated set of human-labeled results. The *r*-HUMO framework and its workflow are shown in Fig. 3.

In the rest of this section, we first describe the basic real-time manner of human workload selection provided with a risk model, and then an alternative batch manner, which can significantly reduce the frequency of human-machine interaction. Finally, we present the technical details of quality assurance. However, the technique of risk analysis will be presented in the following section.



D
D_j
D_j'
Update
Human-labeled Results
Risk Model
Risky Pairs
Human Verification
Merge with a New Selected Subset
(a) r-HUMO Workflow
D-
DH
D+
D_1 D_2 ... D_{l-1} D_l ... D_i D_{i+1} ... D_h D_{h+1} ... D_{m-1} D_m
0
Pair similarity
1
(b) Visualized Workload Partition

Figure 3: *r*-HUMO Framework.

4.1 Real-time Human Workload Selection

Suppose that D has been divided into m unit subsets with increasing metric values of pair similarity, $D = \{D_1, D_2, ..., D_m\}$. Initially, we set $D_H = \emptyset$, $D_- = \{D_1, \dots, D_i\}$, and $D_+ = \{D_{i+1}, D_{i+2}, \dots, D_m\}$. Since the pairs in D_- would be automatically labeled as *unmatching*, the equivalence proportions of the subsets in D_- are expected to be less than 0.5; similarly, the equivalence proportions of the subsets in D_+ are expected to be larger or equal to 0.5.

Similar to HUMO, *r*-HUMO alternately selects the pairs in D_- and D_+ for manual inspection to enforce precision and recall guarantees. Note that compared with first working on D_- and then on D_+, working alternately on D- and D+ would result in the human-labeled pairs with a wider variety of machine metric values between iterations. Risk analysis based on the human-labeled results could therefore be less biased. In the rest of this subsection, we first describe the processes of pair selection on D_- and D_+, and then present the algorithm to enforce quality guarantees based on them.

Pair Selection in D_-. According to the monotonicity assumption of precision, the pairs in D_i

(the rightmost subset in D_-) have higher probabilities of being *equivalent* than any other subset in D_-. Accordingly, they are at the highest risk to be mislabeled by the machine. Therefore, *r*-HUMO sets the initial set of candidate pairs, denoted by D'_-, to be D_i, or $D'_- = D_i$. It then iteratively selects the pairs in D'_- in a risk-wise decreasing order for human inspection. We define the Marginal Equivalence Proportion (MEP) of selection by

$$MEP(D'_-) = \frac{dM}{dN}, \tag{6}$$

in which the variables N and M represent the number of inspected pairs and the number of equivalent pairs among the inspected pairs respectively, and the differential operator "*d*" represents the increment of the variables M and N in a period of human inspection. It can be observed that if risk estimation is effective, $MEP(D'_-)$ would decrease as the selection proceeds. For simplicity of presentation, we denote the expected equivalence proportion of a subset D_i by $EP(D_i)$. Iterative pair selection on D'_- would stop once either of the two following conditions is satisfied:

1. The expected equivalence proportion of the remaining pairs in D'_- falls below the expected equivalence proportion of the rightmost uninspected unit subset adjacent to D'_- in D_-. Denoting the rightmost unit subset by D_j, we can specify the condition by $EP(D'_-) < EP(D_j)$;
2. The marginal equivalence proportion of selection, $MEP(D'_-)$, falls below the expected equivalence proportion of the remaining pairs in D'_-, or $MEP(D'_-) < EP(D'_-)$.

It can be observed that if the first condition is triggered, it means that the pairs mislabeled by the machine can be more easily found in the unit subset D_j instead of D'_-. If the second condition is triggered, it means that risk analysis on D'_- has become ineffectual; all the remaining pairs in D'_- should therefore be either automatically labeled by the machine or manually labeled by the human. In both cases, *r*-HUMO would merge the current candidate set and the rightmost uninspected unit subset adjacent to D'_-, D_j, to constitute a new candidate set, $D'_- = D'_- \cup D_j$. It would then re-estimate pair risk based on the updated human-labeled results and begin a new pair pick-out iteration on the new D'_-. To handle the case when pair selection stops too early, *r*-HUMO sets a threshold for the number of inspected pairs in each iteration. If the number of pairs chosen for inspection is less than the threshold number, it would assign all the remaining pairs in the rightmost unit subset in the candidate set D'_- to the human.

Pair Selection in D_+. The process of human workload selection in D_+ is similar. We denote the candidate set considered for manual inspection in D_+ by D'_+. *r*-HUMO iteratively selects the pairs in D'_+ in a risk-wise decreasing order for human inspection. Since the pairs with lower similarities are at higher risk to be mislabeled by the machine, D'_+ is initially set to be the leftmost unit subset in D_+, or $D'_+ = D_{i+1}$ in Fig. 3. For simplicity of presentation, we denote the leftmost uninspected unit subset adjacent to D'_+ in D_+ by D_k. Iterative pair selection in D'_+ would stop once either of the two following conditions is satisfied:

1. The expected equivalence proportion of the remaining pairs in D'_+ exceeds the expected equivalence proportion of the leftmost uninspected unit subset adjacent to D'_+ in D_+, or $EP(D'_+) > EP(D_k)$;

2. The marginal equivalence proportion of pair selection in D'_+ exceeds the expected equivalence proportion of the remaining pairs in D'_+, or $MEP(D'_+) > EP(D'_+)$.

In both cases, *r*-HUMO would merge D'_+ and D_k to constitute a new candidate set, $D'_+ = D'_+ \cup D_k$. It would then re-estimate pair risk based on the updated human-labeled results and begin a new iteration of pair pick-out on the new D'_+. Similar to the case of D_-, *r*-HUMO sets a lower threshold for the number of inspected pairs in each iteration.

Algorithm. The process of human workload selection alternately selects the pairs in D_- and D_+ for manual inspection. Note that manual pair inspection in D_- would elevate both precision and recall levels. In comparison, manual pair inspection in D_+ could only elevate precision level. The process of real-time human workload selection is sketched in Algorithm 1.

Algorithm 1: Real-time Human Workload Selection in *r*-HUMO.

```
while $recall_L < \beta$ or $precision_L < \alpha$ do
    if $recall_L < \beta$ then
        Iteratively select pairs in $D'_-$ until one of the stop conditions is triggered;
        Re-estimate pair risk;
    end
    if $precision_L < \alpha$ then
        Iteratively select pair in $D'_+$ until one of the stop conditions is triggered;
        Re-estimate pair risk;
    end
end
return $D_H$.
```

4.2 Batch Human Workload Selection

In real-time human workload selection, given a set of candidate pairs, *r*-HUMO iteratively selects the riskiest pair for manual inspection, and updates in real time the equivalence proportion expectation of the remaining candidate pairs and marginal equivalence proportion of selection, based on the human label, to guide the next selection. In other words, it needs to wait for human labeling result until it can generate the next task for human inspection. This setting may be impractical in a real human-machine cooperation environment, as workers do not always respond in a real-time manner. Therefore, in this subsection, we propose a slightly batch version of *r*-HUMO, which allow human to inspect multiple pairs at each iteration.

The idea of batch selection is to predict the least number of pairs that need to be manually inspected such that either of the two stop conditions can be satisfied. We take pair selection in D_- as example. The case for D_+ is similar, thus omitted here. Consider the first condition $EP(D'_-) < EP(D_j)$, which specifies that the expected equivalence proportion of the remaining pairs in the candidate set D'_- falls below the expected equivalence proportion of the rightmost uninspected unit subset adjacent to D'_- in D_-. With the assumption that the marginal equivalence proportion of selection in D'_- decreases monotonously as the selection proceeds, the minimal equivalence proportion of the remaining pairs in D'_{-} after x pairs are selected for manual inspection can be represented by

$$EP_L = \frac{EP(D'_{-}) \cdot n - MEP(D'_{-}) \cdot x}{n - x}, \tag{7}$$

in which n represents the total number of pairs in the original D'_{-}, and $EP(D'_{-})$ represents its expected equivalence proportion. As a result, the least number of pairs, which need to be manually inspected such that the first condition can be satisfied, can be represented by

$$N_1 = \frac{n \cdot (EP(D'_{-}) - EP(D_j))}{MEP(D'_{-}) - EP(D_j)}. \tag{8}$$

Now we consider the second condition, $MEP(D'_{-}) < EP(D'_{-})$. Suppose that the current value of the marginal equivalence proportion is denoted by $MEP(D'_{-})$, which is represented by $\frac{m'}{n'}$, and the current value of the equivalence proportion is denoted by $EP(D'_{-})$. In the worst case that the following x manually inspected pairs are all *inequivalent* pairs, the minimal value of the latest marginal equivalence proportion can be represented by $\frac{m'}{n'+x}$; at the same time, the maximal value of the latest equivalence proportion can be represented by $\frac{EP(D'_{-}) \cdot n}{n-x}$. Therefore, the least number of pairs, which need to be manually inspected for the second condition to be satisfied, can be represented by

$$N_2 = \frac{m' \cdot n - EP(D'_{-}) \cdot n' \cdot n}{m' + EP(D'_{-}) \cdot n}. \tag{9}$$

In summary, at each interaction, r-HUMO can select at least min$\{N_1, N_2\}$ pairs in D'_{-} for manual inspection before either of the two stop conditions can be satisfied.

4.3 Quality Assurance

As in HUMO, the lower bounds of the achieved precision and recall levels of an *r*-HUMO solution are represented by Eq. 3 and Eq. 4 respectively. In this subsection, we present the GPR process to approximate the equivalence proportions of unit subsets and then describe how to compute the lower and upper bounds of the number of equivalent pairs in a given set based on GPR estimates.

GPR Approximation. As HUMO, *r*-HUMO samples a fraction of unit subsets to estimate their equivalence proportions and then uses GPR to approximate the equivalence proportions of other unit subsets. Suppose that k unit subsets are sampled, their observed equivalence proportions are denoted by $\mathsf{R} = [\mathsf{R}_1, \mathsf{R}_2, \ldots, \mathsf{R}_k]^T$. For each unit subset, we also denote its average pair similarity by v_i. Accordingly, we denote the average pair similarities of the k sampled subsets by $V = [v_1, v_2, \ldots, v_k]^T$. Given a new unit subset D_* and its average pair similarity value v_*, GPR assumes that the random variables of $[V^T, v_*]^T$ satisfy a joint Gaussian distribution, which is represented by

$$\begin{bmatrix} V \\ v_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K}(V,V) & \mathbf{K}(V,v_*) \\ \mathbf{K}(v_*,V) & \mathbf{K}(v_*,v_*) \end{bmatrix}\right), \tag{10}$$

in which $\mathbf{K}(\cdot,\cdot)$ represents the covariance matrix. The details of how to compute the covariance matrix can be found in [35]. Based on Eq. 10, the distribution of the equivalence proportion of S_*, R_*, can be represented by the following Gaussian function

$$\mathsf{R}_* \sim N\left(\bar{\mathsf{R}}_*, \sigma^2_{\mathsf{R}_*}\right), \tag{11}$$

in which the mean of R_*, $\bar{\mathsf{R}}_*$, can be represented by

$$\bar{\mathsf{R}}_* = \mathbf{K}(v_*, V) \cdot \mathbf{K}^{-1}(V, V) \cdot \mathsf{R}, \tag{12}$$

and the variance of R_*, $\sigma^2_{\mathsf{R}_*}$, can be represented by

$$\sigma^2_{\mathsf{R}_*} = \mathbf{K}(v_*, v_*) - \mathbf{K}(v_*, V) \cdot \mathbf{K}^{-1}(V, V) \cdot \mathbf{K}(V, v_*). \tag{13}$$

Bound Estimation. Provided with the Gaussian distributions of the equivalence proportions of unit subsets, the number of equivalent pairs in any given set consisting of multiple unit subsets can be estimated by aggregating the distributions of unit subsets. Suppose that the pair set of D_* consists of t unit subsets, $D_* = D^1_* \cup D^2_* \cup \ldots D^t_*$, the total number of pairs in D^i_* $(1 \leq i \leq t)$ is denoted by n^i_*, and the average similarity value of the pairs in D^i_* by v^i_*. Then, the total number of equivalent pairs in D_*, denoted by m_*, can be represented by

$$\mathsf{m}_* \sim N\left(\bar{\mathsf{m}}_*, \sigma^2_{\mathsf{m}_*}\right), \tag{14}$$

in which the mean, $\bar{\mathsf{m}}_*$, can be represented by

$$\bar{m}_* = \sum_{i=1}^{t} n^i_* \cdot \bar{\mathsf{R}}^i_*, \tag{15}$$

and the variance, $\sigma^2_{\mathsf{m}_*}$, can be represented by

$$\sigma^2_{\mathsf{m}_*} = \sum_{1\leq i\leq t, 1\leq j\leq t} n^i_* \cdot n^j_* \cdot cov(v^i_*, v^j_*), \tag{16}$$

in which $cov(v^i_*, v^j_*)$ is the covariance between the two estimates.

In *r*-HUMO, some pairs in a unit subset may be inspected by the human while others are automatically labeled by the machine. In other words, D_H, D_- and D_+ may contain a fraction of the pairs in a unit subset. Consider a pair set consisting of t subsets, $D_*{}' = D^{1'}_* \cup D^{2'}_* \cup \ldots D^{t}_*{}'$, in which $D^{i}_*{}'$ denotes the set of remaining pairs in D^i_* with some of the pairs selected for manual inspection and $D^{i}_*{}' \neq \emptyset$. Suppose that there are r equivalent pairs among all the pairs inspected by the human in D_*, or the pairs in $D_* - D_*{}'$. On the number of equivalent pairs in $D_*{}'$, we have Lemma 1, whose proof is straightforward, thus omitted here.

Lemma 1 *Provided that the number of equivalent pairs in D_* can be represented by the Gaussian function of $N(\bar{m}_*, \sigma^2_{m_*})$, the number of equivalent pairs in D'_*, m'_*, can thus be represented by the Gaussian function of*

$$m'_* \sim N(\bar{m}_* - r, \sigma^2_{m_*}). \tag{17}$$

Proof 1 *Suppose that some pairs in D_*^i have been selected for manual verification and k_i pairs among them are matching pairs. The expectation of the number of matching pairs among the remaining pairs in D_*^i, m_*^i, can be represented by*

$$E({m_*^i}') = E(m_*^i - k_i) = E(m_*^i) - k_i. \tag{18}$$

Accordingly, the expectation of the total number of matching pairs in D'_* can be represented by

$$\begin{aligned} & E({m_*}') = \sum_i E({m_*^i}') \\ & = \sum_i E(m_*^i) - \sum_i k_i \\ & = \bar{m}_* - k. \end{aligned} \tag{19}$$

The covariance between ${m_*^i}'$ and ${m_*^j}'$ can also be represented by

$$\begin{aligned} & cov({m_*^i}', {m_*^j}') = cov(m_*^i - k_i, m_*^j - k_j) \\ & = E[(m_*^i - k_i - E(m_*^i - k_i))(m_*^j - k_j - E(m_*^j - k_j))] \\ & = E[(m_*^i - E(m_*^i))(m_*^j - E(m_*^j))] \\ & = cov(m_*^i, m_*^j). \end{aligned} \tag{20}$$

Therefore, the aggregated covariance of ${D_*}'$ remains the same as that of D_*.

Note that the correctness of Lemma 1 depends on the non-emptiness of unit subsets ${D_*^i}'$. If all the pairs of a unit subset have been chosen for manual inspection and it becomes empty, its estimation covariance with any other estimate on other unit subsets would become zero.

Finally, given the confidence level of θ, the lower and upper bounds of the number of equivalent pairs in a subset D_* can be represented by

$$[\bar{m}_* - \mathcal{Z}_{(1-\theta)} \cdot \sigma_{m_*}, \bar{m}_* + \mathcal{Z}_{(1-\theta)} \cdot \sigma_{m_*}], \tag{21}$$

in which $\mathcal{Z}_{(1-\theta)}$ is the $(1 - \frac{1-\theta}{2})$ point of *standard normal distribution*.

5 Risk Analysis

Risk analysis of *r*-HUMO is performed on the human-labeled results. Given a candidate pair set, *r*-HUMO iteratively selects the most risky pairs in it for manual inspection. It can be said that the performance of *r*-HUMO depends on the effectiveness of risk analysis. Motivated by its success in modern portfolio investment theory [36, 37, 38], *r*-HUMO employs the metric of Conditional Value at Risk (CVaR) to measure the risk of pairs being mislabeled by the machine.

In the portfolio risk theory, given the confidence level of θ, CVaR is defined, in a conservative way, to be the expected loss incurred in the $1 - \theta$ worst case. Formally, given the loss function $z(X) \in L^p(F)$ of a portfolio X and the confidence level of θ, the metric of CVaR is defined as

$$CVaR_\theta(X) = \frac{1}{1-\theta} \int_0^{1-\theta} VaR_{1-\gamma}(X) d\gamma, \tag{22}$$

where $VaR_{1-\gamma}(X)$ represents the minimum loss incurred in the γ worst case and can be formally represented by

$$VaR_{1-\gamma}(X) = \inf\{z_* : P(z(X) \geq z_*) \leq \gamma\}. \tag{23}$$

According to CVaR's definition, risk measurement requires the labeled pair's potential loss estimation. Intuitively, this loss refers to the probability of the pair's label being incorrect. As typical in CVaR evaluation, *r*-HUMO represents the equivalence probability of a pair by a Gaussian distribution and estimates the potential loss based on it. It considers a pair as a portfolio consisting of multiple stocks. Each stock corresponds to a feature of the pair and its loss corresponds to its corresponding feature's equivalence or inequivalence probability. In the rest of this section, we first describe how to extract features from human-labeled pairs, and then present the metric of risk measurement and analyze its complexity.

5.1 Feature Extraction

For general purpose, the features used for risk analysis should have the following three desirable properties: (1) they could be easily extracted from the human-labeled pairs; (2) they should be evidential, or indicative of the equivalence status of a pair; (3) they should be to a large extent independent of the machine metric used in ordering pairs in the first place. It can be observed that in *r*-HUMO, the pairs are generally chosen into the candidate set in the order dictated by a pre-specified machine metric. Therefore, the features independent of the machine metric would be more effective than the non-independent ones in differentiating the pairs with similar metric values in terms of mislabeling risk.

r-HUMO extracts two types of features from the human-labeled pairs, $Same(t_i)$ and $Diff(t_i)$, in which t_i represents a token, $Same(t_i)$ means that t_i appears in both records in a pair, and $Diff(t_i)$ means that t_i appears in one and only one record in a pair. It can be observed that these two features are evidential and easily extractable. Moreover, they were not used in the existing classification metrics for ER. Our risk model assumes that the equivalence probability of a feature satisfies a normal distribution. Given a feature f and a set of human-labeled pairs with the feature f, D_f, the expectation of the equivalence probability of f can be represented by

$$E(f) = \frac{|D_{f+}|}{|D_f|}, \tag{24}$$

in which D_{f+} denotes the set of equivalent pairs in D_f. Its variance can also be represented by

$$V(f) = \frac{1}{|D_f| - 1} \sum_{d_i \in D_f} (L(d_i) - E(f))^2, \tag{25}$$

in which $L(d_i)$ denotes the manual label of a pair d_i in D_f, $L(d_i) = 1$ if d_i is labeled as *matching* and $L(d_i) = 0$ if d_i is labeled as *unmatching*.

5.2 Risk Measurement

In this subsection, we first propose the risk model for the case that features are independent; we then describe how to handle the more complicated case where the features are not independent.

According to the theory of portfolio investment, a pair's equivalence probability distribution can be represented by the weighted linear combination of the distributions of its features. Therefore, provided with the Gaussian distributions of features, the equivalence probability expectation of a pair d, can be represented by

$$E(d) = \sum_{f_i \in F_d} w_d(f_i) \cdot E(f_i), \tag{26}$$

in which F_d denotes the set of features contained in d, and $w_d(f_i)$ denotes the weight of f_i in d. Its variance can also be represented by

$$V(d) = \sum_{f_i \in F_d} w_d(f_i)^2 \cdot V(f_i). \tag{27}$$

The weight of the feature f_i in d is defined as

$$w_d(f_i) = \frac{w(f_i)}{\sum\limits_{f_j \in F_d} w(f_j)}, \tag{28}$$

where $w(f_i)$ denotes the absolute weight of the feature f_i.

In Eq. 28, the absolute feature weights can be simply set to 1: each feature is equally powerful in predicting a given pair's equivalence probability. In most practical scenarios, this assumption may not hold true. Therefore, *r*-HUMO uses the concept of information value [39, 40] to determine feature weight.

The metric of information value has been successfully used to estimate the predictive power of a categorical evidence or attribute in classification problems [39, 40]. *r*-HUMO regards each feature as a categorical evidence. Given a feature f, it defines its weight of evidence by

$$WoE(f) = \ln(\frac{|D_f^-|/|D_h^-|}{|D_f^+|/|D_h^+|}) \tag{29}$$

, in which D_h^+ and D_h^- denotes the set of equivalent pairs and the set of inequivalent pairs in the human-labeled results respectively, and D_f^+ (D_f^-) denotes the set of equivalent pairs (inequivalent pairs respectively) in D_f. The information value of the feature f can then be defined as

$$\begin{aligned} IV(f) &= (\tfrac{|D_f^-|}{|D_h^-|} - \tfrac{|D_f^+|}{|D_h^+|}) \cdot WoE(f) \\ &= (\tfrac{|D_f^-|}{|D_h^-|} - \tfrac{|D_f^+|}{|D_h^+|}) \cdot \ln(\tfrac{|D_f^-|/|D_h^-|}{|D_f^+|/|D_h^+|}). \end{aligned} \tag{30}$$

Given a pair d, we denote its equivalence probability by x, and the probability density function and cumulative distribution function of x by $pdf_d(x)$ and $cdf_d(x)$ respectively. Suppose that d is originally labeled by the machine as *unmatching*. Then, the probability of p being mislabeled by the machine is equal to x. Accordingly, the worst-case loss of d corresponds to the case that x is maximal. Therefore, given the confidence level of θ, the CVaR of d is the expectation of

$z = x$ in the $1-\theta$ cases where x is from ${cdf_d}^{-1}(\theta)$ to $+\infty$. Formally, the CVaR risk of a pair d with the machine label of *unmatching* can be estimated by

$$CVaR_\theta(d) = \frac{1}{1-\theta} \int\limits_{{cdf_d}^{-1}(\theta)}^{+\infty} pdf_d(x) \cdot xdx. \tag{31}$$

Otherwise, d is originally labeled by machine as *matching*. Then the potential loss of d being mislabeled by machine is equal to $1-x$. Therefore, the CVaR risk of a pair d with the machine label of *matching* can be estimated by

$$CVaR_\theta(d) = \frac{1}{1-\theta} \int\limits_{-\infty}^{{cdf_d}^{-1}(1-\theta)} pdf_d(x) \cdot (1-x)dx. \tag{32}$$

The above-described process assumes that the extracted features are independent. Unfortunately, this assumption may not hold true in practice. In the case that the features contained by a pair d are *not* independent, *r*-HUMO again borrows the idea of modern portfolio investment theory and introduces the covariances between features into the process of risk measurement.

Handling Feature Dependency. The above-described process assumes that the extracted features are independent. Unfortunately, this assumption may not hold true in practice. In the case that the features contained by a pair d are *not* independent, *r*-HUMO again borrows the idea of modern portfolio investment theory and introduces the covariances between features into the process of risk measurement. We represent the variance of the equivalence probability of a given pair, d, by

$$V(d) = \sum_{f_i \in F_d} \sum_{f_j \in F_d} w_d(f_i) w_d(f_j) \cdot cov(f_i, f_j). \tag{33}$$

in which F_d denotes the set of features contained in d, $w_d(f_i)$ denotes the weight of f_i in d, and $cov(f_i, f_j)$ denotes the covariance between the equivalence probability of f_i and f_j.

Given two features f_i and f_j, their covariance, $cov(f_i, f_j)$, can be represented by

$$E(P(+|f_i) \cdot P(+|f_j)) - E(P(+|f_i)) \cdot E(P(+|f_j)) \tag{34}$$

, in which $P(+|f_i)$ denotes the equivalence probability of the feature f_i. We can estimate $E(P(+|f_i))$ by

$$E(P(+|f_i)) = \frac{|D^+_{f_i}|}{|D_{f_i}|} \tag{35}$$

, in which D_{f_i} denotes the set of manually-labeled pairs containing the feature f_i and $D^+_{f_i}$ denote the set of equivalent pairs in D_{f_i}. Similarly, we estimate $E(P(+|f_i) \cdot P(+|f_j))$ based on the set of manually-labeled pairs containing both f_i and f_j, denoted by $D_(f_i, f_j)$, by

$$E(P(+|f_i) \cdot P(+|f_j)) = (\frac{|D^+_{(f_i,f_j)}|}{|D_{(f_i,f_j)}|})^2 \tag{36}$$

, in which $D^{+}_{(f_i,f_j)}$ denotes the set of equivalent pairs in $D_{(f_i,f_j)}$.

It can be observed that two features are correlated by co-occurrence in a pair. Accordingly, *r*-HUMO defines a correlation factor between two features by feature co-occurrence, and estimates their covariance if and only if their correlation factor exceeds a pre-defined threshold. We define the correlation factor between two features, f_i and f_j, by

$$CoF(f_i, f_j) = \frac{P(f_i, f_j)}{P(f_i) \cdot P(f_j)} \tag{37}$$

, in which $P(f_i)$ denotes the probability that a pair in D contains the feature of f_i, and $P(f_i, f_j)$ the probability that a pair contains both f_i and f_j. In practical implementation, the threshold of $CoF(f_i, f_j)$ can be set to be equal or bigger than 1 (e.g., 10).

5.3 Complexity Analysis

It can be observed that in each iteration, the total frequency of feature occurrence is bound by $\boldsymbol{O}(l \cdot n)$, in which n denotes the total number of pairs in a workload and l denotes the maximal number of tokens in a pair. As a result, the time complexity of computing the feature distributions in each iteration is bounded by $\boldsymbol{O}(m \cdot n)$. Accordingly, without considering feature dependency, the time complexity of computing the CVaR risk of the candidate pairs in each iteration is also bounded by $\boldsymbol{O}(l \cdot n)$. Since the number of iterations is at most $\boldsymbol{O}(n)$, the total time complexity of risk analysis can be represented by $\boldsymbol{O}(l \cdot n^2)$. Therefore, we have Theorem. 1, whose proof follows naturally from the above analysis.

Theorem 1 *The space and time complexities of risk analysis without considering feature dependency can be represented by* $\mathbf{O}(l \cdot n)$ *and* $\mathbf{O}(l \cdot n^2)$ *respectively, in which n denotes the total number of pairs in a workload and l denotes the maximal number of tokens in a pair.*

Now we analyze the space and time complexity of computing feature dependency. It can be observed that the total number of pairs of interdependent features is bounded by $\boldsymbol{O}(m^2 \cdot n)$. It follows that the space complexity of computing feature dependency can be represented by $\boldsymbol{O}(m^2 \cdot n)$. The time complexity of computing their covariances is also bounded by $\boldsymbol{O}(m^2 \cdot n)$. Since feature covariances can be computed incrementally at each new iteration, the total time complexity for computing feature dependency can be represented by $\boldsymbol{O}(m^2 \cdot n^2)$.

Proof 2 *The maximum number of human-labeled pairs and candidate pairs are the number of all pairs n, and the maximum number of iterations is also n when extreme case occurs that* r-*HUMO updates risk model every pair verification and assigns all the pairs for human verification. Assuming that the maximum number of tokens contained in an entity record is constant c, then the maximum number of human labeled results as samples of all the features is $2c \cdot n$ in total: Consider that we have assigns all the n pairs for human verification; if there are no same features between any 2 pairs, then each feature will have only 1 sampled result, and the total number of these results is $1 \cdot 2c \cdot n = 2c \cdot n$; once there is a feature appears 1 time more among these pairs, the feature will have 1 more sampled results, while the number of total different features of the pairs will decrease 1, therefore, the number of human labeled results of all the* *features stays the same as $2c \cdot n$; and so on, the maximum number of human labeled results of all the features always keeps $2c \cdot n$ in total.*

The time complexity of r-*HUMO consists of computing on many iterations, and for each iteration, the computing mainly consists of 2 parts: (1) the estimation of expectations and variances of the distributions of features, taking $O(2c \cdot n) = O(n)$ time on processing the human labeled results of all the features; (2) the estimation of risks of candidate pairs, taking $2c \cdot O(n) = O(n)$ time on processing the distributions of all candidate pairs; (3) sorting the candidate pairs in the decreasing order of risk, taking $O(n \cdot log(n))$ time. Therefore, for n iterations, the time complexity of* r-*HUMO is $O(n^2 \cdot log(n))$.*

The space complexity of r-*HUMO consists of storage mainly consists of 2 parts: (1) the features and their expectations and variances of the distributions; (2) the candidate pairs and their risks. Each part needs $O(n)$ space. Therefore, the space complexity of* r-*HUMO is $O(n)$.*

6 Experimental Study

In this section, we empirically evaluate the performance of *r*-HUMO on real data by comparative study. We compare *r*-HUMO with the state-of-the-art alternative HUMO [14, 15], which can enforce both precision and recall, as well as two baselines. Note that most of existing ER techniques can not enforce the quality guarantees measured by precision and recall. Their comparative performance evaluation is therefore beyond the scope of this paper. However, we also compare r-HUMO with the active learning-based approach (denoted by ACTL) [12], which can at least enforce precision. ACTL can maximize recall while ensuring a pre-specified precision level. It estimates the achieved precision level of a labeling solution by sampling. As a result, ACTL also requires manual inspection. We compare *r*-HUMO with ACTL on the achieved quality and the required manual cost.

The rest of this section is organized as follows: Subsection 6.1 describes the experimental setup. Subsection 6.2 evaluates quality guarantee of *r*-HUMO. Subsection 6.3 compares *r*-HUMO with HUMO. Subsection 6.4 compares *r*-HUMO with two baselines. Subsection 6.5 compares *r*-HUMO with ACTL. Subsection 6.6 evaluates the efficacy of batch human workload selection. Finally, Subsection 6.9 evaluates the efficiency and scalability of *r*-HUMO. Subsection. 6.7 evaluates the effectiveness of the proposed risk model. Subsection 6.8 evaluates the effectiveness of feature weighting in risk measurement.

6.1 Experimental Setup

Our evaluation is conducted on the three real datasets, whose details are described as follows:

- DBLP-Scholar[1] (denoted by DS): The DS dataset contains 2616 publication entities from DBLP and 64263 publication entities from Google Scholar. The experiments match the DBLP entries with the Scholar entries. After blocking, the DS workload has 100077 pairs and 5267 among them are match pairs.

[1] available at https://dbs.uni-leipzig.de/file/DBLP-Scholar.zip

- Abt-Buy[2] (denoted by AB): The AB dataset contains 1081 product entities from Abt.com and 1092 product entities from Buy.com. The experiments match the Abt entries with the Buy entries. After blocking, the AB workload has 313040 pairs and 1085 among them are match pairs.
- Songs[3] (denoted by SG): The SG dataset contains 1000000 song entities, some of which refer to the same songs. The experiments match the song entries in the same table. After blocking, the SG workload contains 289893 pairs and 13756 among them are match pairs.

Our empirical study uses pair similarity as the machine metric. It computes pair similarity by aggregating attribute similarities with weights [14, 15]. Specifically, on the DS dataset, Jaccard similarity of the attributes *title* and *authors*, and Jaro-Winkler distance of the attribute *venue* are used; on the AB dataset, Jaccard similarity of the attributes *product name* and *product description* are used; on the SG dataset, Jaccard similarity of the attributes *song title*, *release information* and *artist name*, Jaro-Winkler distance of the attributes *song title* and *release information*, and number similarity of the attributes *duration*, *artist familiarity*, *artist hotness* and *year* are used. The weight of each attribute is determined by the number of its distinct values. As in [12, 15], we use the blocking technique to filter the instance pairs unlikely to match. Specifically, the DS workload contains the instance pairs whose aggregated similarity values are no less than 0.2. Similarly, the aggregated similarity value thresholds for the AB and SG workloads are set to 0.05 and 0.2. After blocking, the DS workload has 100077 pairs, 5267 among them are equivalent pairs; the AB workload has 313040 pairs, 1085 among them are equivalent pairs; the SG workload has 289893 pairs and 13756 among them are equivalent pairs.

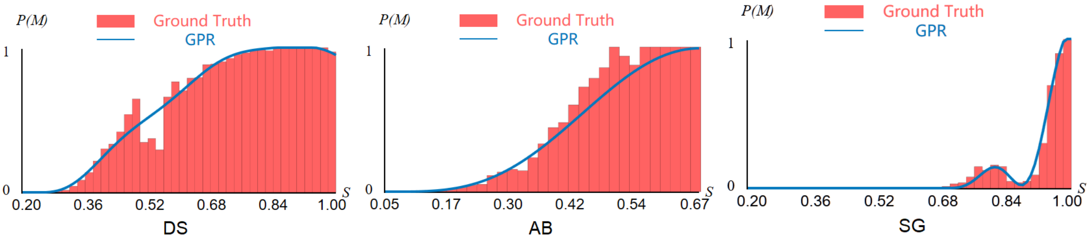


Figure 4: the Equivalence Proportions of Unit Subsets with regard to Pair Similarity.

As in the HUMO implementation, our *r*-HUMO implementation partitions an ER workload into disjoint unit subsets, each of which contains the same number of instance pairs. The number of instance pairs contained by each subset is set to 200. The equivalence proportions of the uit subsets with regard to pair similarity on the three workloads are presented in Figure. 4. It can be observed that on all the three test datasets, monotonicity of precision is a general trend. Specifically, on the SG dataset, which to the largest extent violates the assumption among them, monotonicity is not satisfied only in a small range of similarity value, between 0.70 and 0.82.

To balance the sampling cost and the accuracy of equivalence proportion approximation, as in the HUMO implementation [14, 15], *r*-HUMO sets both lower and upper limits on sampling

[2] available at https://dbs.uni-leipzig.de/file/Abt-Buy.zip
[3] available at http://pages.cs.wisc.edu/~anhai/data/falcon_data/songs

cost, which is measured by the proportion of sampled unit subsets among all the subsets. In our experiments, the range of the sampling proportion is set between 3% and 5%. We observe that considering feature dependency in risk analysis for *r*-HUMO can only marginally improve the performance on the test workloads. We therefore report the results of *r*-HUMO without considering feature dependency.

Note that in *r*-HUMO, different runs may generate different labeling solutions due to sampling randomness. For each experiment, we therefore run the program 20 times on each workload and report the averaged result. In our experiments, we have the ground-truth labels of all the test pairs. The ground-truth labels are originally hidden; whenever manual inspection is called, they are provided to the program.

6.2 Quality Enforcement

In the experiments, we specify 7 scales of quality requirement, whose precision and recall are set at different levels at 0.8, 0.825, 0.85, 0.875, 0.9, 0.925 and 0.95 respectively. The confidence level on quality guarantee is set at 0.9.

Table 2: Evaluation of Quality Enforcement.

Dataset	Required Quality	Achieved Quality		
	α=β	α	β	SR(%)
DS	0.825	0.9079	0.8459	100
	0.850	0.9098	0.8657	100
	0.875	0.9124	0.8904	100
	0.900	0.9248	0.9150	100
	0.925	0.9497	0.9391	100
	0.950	0.9748	0.9628	95
AB	0.800	0.9629	0.8546	100
	0.825	0.9630	0.8589	100
	0.850	0.9635	0.8718	100
	0.875	0.9643	0.8920	100
	0.900	0.9651	0.9112	100
	0.925	0.9671	0.9398	100
	0.950	0.9693	0.9508	90
SG	0.800	0.9663	0.8957	90
	0.825	0.9671	0.9201	90
	0.850	0.9678	0.9417	90
	0.875	0.9683	0.9581	90
	0.900	0.9686	0.9660	90
	0.925	0.9687	0.9694	90
	0.950	0.9746	0.9708	90

The detailed evaluation results are presented in Table 2, in which α denotes precision, β denotes recall, and SR denotes success rate (the percentage of the successful runs achieving required quality levels among all the runs). On DS, the initial machine labeling solution achieves precision and recall levels above 0.8; we therefore do not report the results in the table. It can be observed that *r*-HUMO is effective in enforcing quality guarantees. On all the three workloads, the achieved quality levels are considerably above the required levels in most cases and the achieved success rates consistently exceed the confidence level of 0.9.

6.3 *r*-HUMO VS HUMO

We compare the manual cost consumed by *r*-HUMO and HUMO given the same quality requirement. Bear in mind that the manual cost includes both the sampling cost and the cost of manually inspecting the pairs in D_H. Since *r*-HUMO and HUMO consume the same amount of sampling cost in each run, we compare the size of D_H (excluding sampled pairs). On all the three datasets, the consumed sampling cost is very close to the upper limit of 5% in most runs; averagely, the sampling cost of DS, AB and SG are 4.93%, 4.94% and 4.96% respectively.

The comparative results of *r*-HUMO and HUMO are presented in Table 3. It can be observed that given the same quality requirement, *r*-HUMO consistently consumes less human cost than HUMO and the cost difference between them is considerable in most cases. *r*-HUMO iteratively selects a few mislabeled pairs from a set of mostly correctly labeled pairs. With the tighter quality requirement, the task becomes more challenging because it has to identify the mislabeled pairs in the unit subsets with increasingly low mislabeling proportion. In other words, in *r*-HUMO, the required human cost for finding out a fixed number of mislabeled pairs would increase with the decreasing equivalence proportion. Accordingly, its advantage over HUMO would gradually become smaller as pair selection proceeds. As a result, as shown in Table 3, the performance difference between *r*-HUMO and HUMO narrows down as quality requirement is enhanced.

Table 3: Performance Comparison between *r*-HUMO and HUMO.

Data set	Required Quality	Size of D_H (excl. samples)		
	α=β	HUMO	*r*-HUMO	Reduction(%)
DS	0.825	108	**37**	65.74
	0.850	463	**151**	67.39
	0.875	927	**302**	67.42
	0.900	1255	**538**	57.13
	0.925	1852	**967**	47.79
	0.950	2802	**1786**	36.26
AB	0.800	4628	**4144**	10.46
	0.825	6056	**5664**	6.47
	0.850	7894	**7179**	9.06
	0.875	10239	**8328**	18.66
	0.900	13495	**10662**	20.99
	0.925	28273	**27380**	3.16
	0.950	34649	**34136**	1.48
SG	0.800	62940	**28007**	55.50
	0.825	66672	**34615**	48.08
	0.850	68598	**43055**	37.24
	0.875	71667	**59654**	16.76
	0.900	75698	**74156**	2.04
	0.925	83156	**81903**	1.51
	0.950	89742	**88190**	1.73

The results reported in Table 3 are based on the GPR approximation that uses the samples, which consists of only a small portion (less than 5%) of all the pairs in a workload. However, the results of GPR approximation may not be accurate under many circumstances. The estimation accuracy of GPR approximation may significantly affect the computation of D_H's boundaries. To further validate the effectiveness of *r*-HUMO's pair prioritization strategy, we also execute *r*-HUMO and HUMO with the extreme assumption that the estimations of subset equivalence proportions are exactly right. In the new experimental setting, both *r*-HUMO and HUMO are given the exact number of equivalent pairs in each unit subset (instead of GPR estimation) beforehand as well as the same quality requirement. The comparative results on the three workloads are presented in Table 4. It can be observed that *r*-HUMO consistently outperforms HUMO in all the test cases, and compared with the previous setting of GPR approximation, *r*-HUMO outperforms HUMO by more considerable margins. *Even though the results reported in Table 4 are unrealistic in practical scenarios, they do illustrate the efficacy of r-HUMO's pair prioritization strategy and demonstrate that the performance advantage of r-HUMO over HUMO can increase with the tightness of GPR approximation.*

Note that the bounds of precision and recall are estimated by aggregating the GPR approximations over all the unit subsets. Accordingly, small variations on the equivalence proportions of unit subsets can lead to comparatively large variations on the estimated precision and recall. Pair selection based on GPR approximation is therefore usually very conservative, resulting in human labeling cost much more than what is necessary for quality guarantee. In Table 2, we can observe that most precision and recall achieved by r-HUMO exceed the required levels by considerable margins. If provided with ground-truth equivalence proportions on all the unit subsets, precision and recall could be estimated with certainty. The achieved precision and recall level of r-HUMO would therefore be much closer to the required levels. Since the efficacy of risk analysis decreases with the required quality, the cost reductions reported in Table 4 are more considerable than those reported in Table 3. However, it does not mean that GPR approximation is ineffective. The accuracy of the bounds established by GPR approximation is necessary for quality guarantee. As shown in Figure 4, the GPR approximation is generally accurate on all the three test datasets. The success of r-HUMO in quality guarantee, as shown in Table 3, also validates the effectiveness of the GPR approximation.

Table 4: *r*-HUMO vs HUMO with Ground-Truth Equivalence Proportions.

Data set	Required Quality	Size of D_H (excl. samples)		
	α=β	HUMO	*r*-HUMO	Reduction(%)
DS	0.850	443	**78**	82.39
	0.875	949	**221**	76.71
	0.900	1080	**373**	65.46
	0.925	1574	**586**	62.77
	0.950	2545	**1020**	59.92
AB	0.800	2519	**953**	62.17
	0.825	3997	**1359**	66.00
	0.850	5971	**2041**	65.82
	0.875	9617	**2899**	69.86
	0.900	12671	**4045**	68.08
	0.925	19466	**14628**	24.85
	0.950	33226	**32020**	3.77
SG	0.800	62298	**12796**	79.46
	0.825	65101	**14052**	78.42
	0.850	67983	**15505**	77.19
	0.875	71006	**17558**	75.27
	0.900	72820	**19584**	73.11
	0.925	75640	**23693**	68.68
	0.950	80869	**30287**	62.55

6.4 *r*-HUMO VS BASELINES

To further validate the efficacy of the proposed risk analysis technique, we also compare its performance with two baseline alternatives for pair selection, the native random strategy (denoted by Rand) and the simple strategy based on the distance from the center of the scale of a workload (denoted by CoS). In the experiments, the center of the scale is computed based on attribute similarities. Given the same GPR approximation result, we measure the achieved precision and recall of different strategies with the same amount of human cost budget. We set the cost budget as the number of manually inspected pairs required by *r*-HUMO to enforce the specified quality guarantees. The detailed comparative results are presented in Table 5. It can be observed that given the same cost budget, *r*-HUMO achieves considerably better quality than both Rand and CoS. These experimental results show that *r*-HUMO is considerably more accurate in picking out the mislabeled pairs than Rand and CoS. They validate the efficacy of the proposed risk analysis technique.

Table 5: *r*-HUMO vs Two Baselines.

Dataset	Cost	*r*-HUMO		CoS		Rand	
		α	β	α	β	α	β
DS	37	0.9079	0.8459	0.9075	0.8407	0.9075	0.8407
	151	0.9098	0.8657	0.9075	0.8407	0.9075	0.8407
	302	0.9124	0.8904	0.9076	0.8409	0.9075	0.8408
	538	0.9248	0.9150	0.9076	0.8409	0.9075	0.8408
	967	0.9497	0.9391	0.9076	0.8409	0.9075	0.8408
	1786	0.9748	0.9628	0.9076	0.8409	0.9076	0.8409
AB	4144	0.9629	0.8546	0.9475	0.6571	0.9475	0.6574
	5644	0.9630	0.8589	0.9475	0.6571	0.9475	0.6576
	7179	0.9635	0.8718	0.9475	0.6571	0.9475	0.6576
	8328	0.9643	0.8920	0.9475	0.6571	0.9475	0.6576
	10662	0.9651	0.9112	0.9475	0.6571	0.9475	0.6578
	27380	0.9671	0.9398	0.9476	0.6589	0.9476	0.6587
	34136	0.9693	0.9508	0.9476	0.6589	0.9477	0.6594
SG	28007	0.9663	0.8957	0.9177	0.3381	0.9179	0.3390
	34615	0.9671	0.9201	0.9177	0.3381	0.9180	0.3394
	43055	0.9678	0.9417	0.9177	0.3382	0.9181	0.3399
	59654	0.9683	0.9581	0.9179	0.3387	0.9184	0.3412
	74156	0.9686	0.9660	0.9180	0.3393	0.9188	0.3429
	81903	0.9687	0.9694	0.9180	0.3394	0.9190	0.3442
	88190	0.9746	0.9708	0.9181	0.3399	0.9193	0.3454

6.5 *r*-HUMO vs ACTL

Table 6: *r*-HUMO vs ACTL on Recall given the Same Precision.

Data	Required	Achieved Recall		$\psi(\%)$		$\frac{\Delta\psi}{100\cdot\Delta Recall}$
set	Precision	*r*-HUMO	ACTL	*r*-HUMO	ACTL	
DS	0.825	0.8459	0.8176	4.97	3.46	0.5335
	0.850	0.8657	0.7999	5.08	3.70	0.2097
	0.875	0.8904	0.8000	5.23	2.98	0.2489
	0.900	0.9150	0.7662	5.47	3.17	0.1546
	0.925	0.9391	0.7557	5.90	3.83	0.1129
	0.950	0.9628	0.7273	6.71	2.56	0.1762
AB	0.800	0.8546	0.1558	6.26	0.29	0.0854
	0.825	0.8589	0.1395	6.75	0.28	0.0899
	0.850	0.8718	0.1578	7.23	0.28	0.0973
	0.875	0.8920	0.1152	7.60	0.27	0.0944
	0.900	0.9112	0.1152	8.35	0.29	0.1013
	0.925	0.9398	0.0857	13.69	0.19	0.1581
	0.950	0.9508	0.0857	15.85	0.19	0.1810
SG	0.800	0.8957	0.3337	14.62	0.30	0.2548
	0.825	0.9201	0.3284	16.90	0.27	0.2811
	0.850	0.9417	0.3271	19.81	0.34	0.3168
	0.875	0.9581	0.3259	25.54	0.42	0.3973
	0.900	0.9660	0.3107	30.54	0.54	0.4578
	0.925	0.9694	0.2530	33.21	0.42	0.4577
	0.950	0.9708	0.2469	35.38	0.38	0.4835

In this subsection, we compare *r*-HUMO with the active learning based (ACTL) alternative. We have implemented of both of the techniques proposed in [12] and [13] respectively. Our experiments showed that they perform similarly on the achieved quality and required manual work. Here, we present the comparative evaluation results between *r*-HUMO and the technique proposed in [12]. As [12], we employ Jaccard similarity, edit distance and number similarity on attributes used in Subsection 6.1 as the similarity space for ACTL. On DS, the used attributes are *title* and *authors*; on AB, they are *product name* and *product description*; and on SG, they are *song title*, *release information*, *artist name*, *duration*, *artist familiarity*, *artist hotness* and *year*. ACTL uses sampling to estimate the achieved precision level of a given classification solution; therefore it also requires manual work.

In our experiments, the required precision and recall levels are set to be the same for *r*-HUMO. Considering that ACTL can not enforce recall level; at each given precision level, we record *r*-HUMO and ACTL's performance difference on the achieved recall and the consumed human cost. The detailed comparison results between *r*-HUMO and ACTL are presented in Table 6, in which ψ represents the percentage of manual work, and Δ denotes the performance difference between the two methods on a specified metric. It can be observed that the achieved recall level by ACTL generally decreases with the specified precision level. In all the test cases, *r*-HUMO achieves higher recall levels than ACTL. We also record the additional human cost required by *r*-HUMO for the absolute recall improvement of 1% over ACTL (at the last columns of Table 6. It can be observed that, with both precision and recall set at the high level of 0.95, the cost is as low as 0.1762% on DS, 0.1810% on AB and 0.4835% on SG.

Table 7: *r*-HUMO vs ACTL on F1 given the Same Precision.

Data set	Required Precision	Achieved F1		$\psi(\%)$		$\frac{\Delta\psi}{100\cdot\Delta F1}$
		r-HUMO	ACTL	*r*-HUMO	ACTL	
DS	0.825	0.8758	0.8057	4.97	3.46	0.2154
	0.850	0.8872	0.8067	5.08	3.70	0.1714
	0.875	0.9013	0.8130	5.23	2.98	0.2548
	0.900	0.9199	0.8187	5.47	3.17	0.2273
	0.925	0.9444	0.8220	5.90	3.83	0.1691
	0.950	0.9688	0.8161	6.71	2.56	0.2718
AB	0.800	0.9055	0.2626	6.26	0.29	0.0929
	0.825	0.9080	0.2408	6.75	0.28	0.0970
	0.850	0.9154	0.2653	7.23	0.28	0.1069
	0.875	0.9267	0.2053	7.60	0.27	0.1016
	0.900	0.9374	0.2053	8.35	0.29	0.1101
	0.925	0.9533	0.1575	13.69	0.19	0.1696
	0.950	0.9600	0.1575	15.85	0.19	0.1951
SG	0.800	0.9297	0.4807	14.62	0.30	0.3189
	0.825	0.9430	0.4790	16.90	0.27	0.3584
	0.850	0.9546	0.4780	19.81	0.34	0.4085
	0.875	0.9632	0.4769	25.54	0.42	0.5166
	0.900	0.9673	0.4629	30.54	0.54	0.5948
	0.925	0.9690	0.4013	33.21	0.42	0.5776
	0.950	0.9727	0.3939	35.38	0.38	0.6047

It can be observed that given the same precision requirement, ACTL and *r*-HUMO might actually achieve different precision levels. Therefore, we also compare their actual performance on the F1 metric and record the additional human cost required by *r*-HUMO for the absolute F1 improvement of 1% over ACTL. The detailed results are presented in Table 7. Similar to what was observed in Table 6, the additional human cost generally increases with the specified precision level. On DS, the additional human cost of *r*-HUMO for 1% increase in F1 score is maxed at 0.2718%. On AB and SG, it is as low as 0.1951% and 0.6047% respectively. Along with the results presented in Table 6, these results clearly demonstrate that compared with ACTL, *r*-HUMO can effectively improve the resolution quality with reasonable ROI in terms of human cost.

6.6 *r*-HUMO: REAL-TIME VS BATCH MODE

In this subsection, we compare the performance of *r*-HUMO in the real-time and batch-mode settings. Given the same GPR approximations, we compare the performance by the frequency of required human and machine interaction and the total amount of required manual work (the size of D_H excluding sampled pairs) as well as the achieved quality. The detailed evaluation results are presented in Table 8 and Table 9. It can be observed that the batch mode achieves very similar performance to the real-time mode in terms of resolution quality and human cost, while significantly reducing the frequency of required interactions (up to 90+%). These experimental results clearly validate the efficacy of the proposed batch mechanism.

Table 8: *r*-HUMO: Real-time vs Batch on Human Cost.

Data	Required Quality	Size of D_H (excl. samples)		Interaction Frequency
set	α=β	Real Time	Batch	Reduction(%)
DS	0.825	**24**	**24**	90.17
	0.850	**150**	**150**	91.10
	0.875	**325**	326	91.28
	0.900	**638**	640	91.26
	0.925	**1132**	1134	91.04
	0.950	**2074**	2118	91.15
AB	0.800	**4049**	4089	83.19
	0.825	**5646**	5668	81.27
	0.850	7195	**7156**	78.09
	0.875	**8323**	8335	74.50
	0.900	**10410**	10427	70.04
	0.925	**25615**	25634	57.35
	0.950	**34547**	34557	44.59
SG	0.800	**24222**	24233	89.39
	0.825	**26980**	27012	90.31
	0.850	**32121**	32176	91.67
	0.875	**43078**	43428	93.33
	0.900	**56651**	56847	94.54
	0.925	**76098**	76449	95.81
	0.950	**84894**	85030	96.01

Table 9: *r*-HUMO: Real-time vs Batch on Achieved Quality.

Dataset	Required	Achieved Quality			
	Quality	Real Time		Batch Manner	
	α=β	α	β	α	β
DS	0.825	0.8986	0.8537	0.8986	0.8537
	0.850	0.9009	0.8759	0.9009	0.8759
	0.875	0.9058	0.9008	0.9059	0.9010
	0.900	0.9253	0.9257	0.9256	0.9261
	0.925	0.9503	0.9506	0.9511	0.9505
	0.950	0.9755	0.9694	0.9766	0.9692
AB	0.800	0.9620	0.8540	0.9619	0.8533
	0.825	0.9622	0.8587	0.9622	0.8584
	0.850	0.9627	0.8712	0.9627	0.8713
	0.875	0.9635	0.8908	0.9635	0.8909
	0.900	0.9643	0.9105	0.9643	0.9106
	0.925	0.9671	0.9398	0.9671	0.9398
	0.950	0.9693	0.9508	0.9693	0.9508
SG	0.800	0.9667	0.8703	0.9667	0.8703
	0.825	0.9676	0.8967	0.9676	0.8967
	0.850	0.9685	0.9232	0.9685	0.9232
	0.875	0.9693	0.9490	0.9693	0.9486
	0.900	0.9698	0.9663	0.9698	0.9659
	0.925	0.9734	0.9726	0.9735	0.9731
	0.950	0.9835	0.9732	0.9836	0.9738

6.7 Risk Model Effectiveness

In order to evaluate the effectiveness of the CVaR risk model proposed for *r*-HUMO, we compare its performance to that of the expectation loss (EL) risk model in this subsection. The EL model simply computes the mislabeled risk of a pair as the expectation of its mislabeled probability. Therefore, the EL model estimates the risk of a pair p with the machine label of *unmatching* as

$$EL(p) = E(x) \tag{38}$$

, and the risk of a pair p with the machine label of *matching* as

$$EL(p) = 1 - E(x) \tag{39}$$

, where $E(x)$ stands for the estimated probability expectation of p being matching based on feature distributions.

Table 10: Risk Model Evaluation.

Dataset	Required Quality	Size of D_H (excl. samples)	
	α=β	EL	CVaR
DS	0.825	**37**	**37**
	0.850	**151**	**151**
	0.875	305	**302**
	0.900	548	**538**
	0.925	989	**967**
	0.950	1814	**1786**
AB	0.800	4167	**4144**
	0.825	5706	**5664**
	0.850	7327	**7179**
	0.875	8472	**8328**
	0.900	**10140**	10662
	0.925	**27378**	27380
	0.950	**34124**	34136
SG	0.800	29603	**28007**
	0.825	37036	**34615**
	0.850	46462	**43055**
	0.875	62291	**59654**
	0.900	76462	**74156**
	0.925	84050	**81903**
	0.950	89710	**88190**

Table 11: Risk Model Evaluation with Ground-Truth Match Proportions.

Dataset	Required Quality	Size of D_H (excl. samples)	
	α=β	EL	CVaR
DS	0.850	79	**78**
	0.875	**221**	**221**
	0.900	381	**373**
	0.925	595	**586**
	0.950	1031	**1020**
AB	0.800	1066	**953**
	0.825	1540	**1359**
	0.850	2219	**2041**
	0.875	2961	**2899**
	0.900	4107	**4045**
	0.925	16917	**14628**
	0.950	32134	**32020**
SG	0.800	12978	**12796**
	0.825	14324	**14052**
	0.850	16033	**15505**
	0.875	18513	**17558**
	0.900	20826	**19584**
	0.925	26571	**23693**
	0.950	36577	**30287**

The comparative results of the CVaR and EL risk models on the three workloads are presented in Table 10. Their comparative performance provided with ground-truth match proportions are also presented in Table 11. In all the tables, the cost corresponding to the better performance is emphasized in **bold**. It can be observed that given the same quality requirement, the CVaR risk model requires less human cost than the EL model on most of the test cases (in the cases of SG, the margins are considerable); in the cases where it performs worse, their cost difference is only marginal. EL selects pairs on their incorrect probability on average, while CVaR remains risk-averse to select pairs based on the cases that are most probable to be incorrect. Our experimental results show that for the complicated and challenging ER workloads, a pair is usually not "lucky" enough to be correctly labeled by chance on some optimistic cases, and even not on average. It is thus sensible that we remain conservative and critical, to consider the target pairs with a more prudent care on those worst cases that could occur with a certain probability.

6.8 Effectiveness of Feature Weighting

Table 12: Evaluation of Feature Weighting.

Dataset	Required Quality	Size of D_H (excl. samples)	
	α=β	EW	IV
DS	0.825	**37**	**37**
	0.850	152	**151**
	0.875	303	**302**
	0.900	540	**538**
	0.925	980	**967**
	0.950	1817	**1786**
AB	0.800	4465	**4144**
	0.825	5714	**5664**
	0.850	7586	**7179**
	0.875	9085	**8328**
	0.900	11403	**10662**
	0.925	27403	**27380**
	0.950	34159	**34136**
SG	0.800	25230	**19062**
	0.825	26920	**20587**
	0.850	29480	**22807**
	0.875	33604	**26025**
	0.900	35126	**30470**
	0.925	43277	**36368**
	0.950	82078	**81603**

In this subsection, we evaluate the effectiveness of *r*-HUMO's feature weighting strategy based on information value (IV). We compare it with the simple equally-weighting alternative (EW). The comparative results on the three workloads are presented in Table 12. Their comparative performance provided with ground-truth equivalence proportions are also presented in Table 13. It is clear that IV consistently outperforms EW, and in some cases, their performance margins are considerable. Our experimental results demonstrate the effectiveness of the feature weighting strategy based on information value.

Table 13: Evaluation of Feature Weighting with Ground-Truth Equivalence Proportions.

Dataset	Required Quality	Size of D_H (excl. samples)	
	α=β	EW	IV
DS	0.850	**78**	**78**
	0.875	**221**	**221**
	0.900	376	**373**
	0.925	591	**586**
	0.950	1026	**1020**
AB	0.800	1222	**953**
	0.825	1921	**1359**
	0.850	2911	**2041**
	0.875	4082	**2899**
	0.900	7067	**4045**
	0.925	17940	**14628**
	0.950	32025	**32020**
SG	0.800	13494	**12796**
	0.825	14801	**14052**
	0.850	16318	**15505**
	0.875	18553	**17558**
	0.900	20655	**19584**
	0.925	24754	**23693**
	0.950	32138	**30287**

6.9 Efficiency and Scalability

In this subsection, we evaluate the efficiency and scalability of our *r*-HUMO implementation on different data scales. We perform random sampling on the DS dataset to generate the test workloads with different data scales. We measure the efficiency by the consumed run time.

The evaluation results are presented in Figure. 5. It can be observed that as data scale increases, the run time increases polynomially as dictated by the complexity analysis results. As expected, the run time increases more dramatically with data scale as the quality requirement becomes more strict.

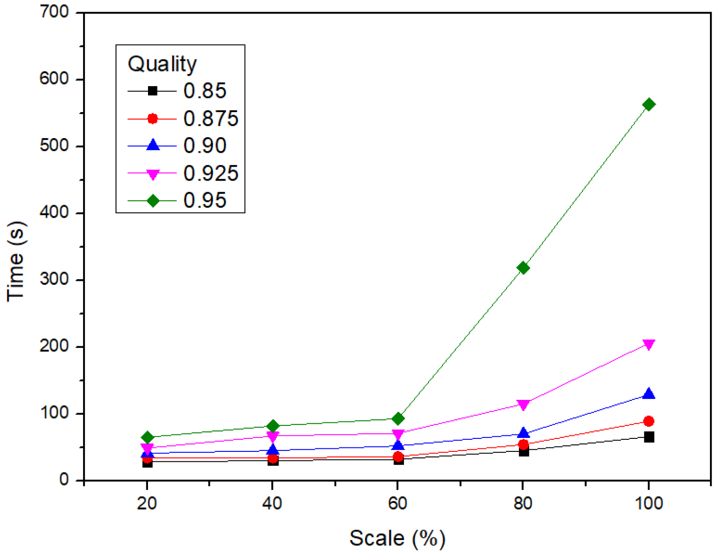


Figure 5: Evaluation of *r*-HUMO Efficiency and Scalability on DS.

7 Conclusion

In this paper, we have proposed a risk-aware human-machine cooperation framework, *r*-HUMO, for entity resolution with quality guarantees. Different from the existing HUMO framework, *r*-HUMO takes advantage of the manually-labeled results to measure the risk of pairs being mislabeled by machine, thus can effectively reduce required manual work. Our extensive experiments on real data have also validated the efficacy of *r*-HUMO.

For large workload, crowdsourcing may be the only feasible solution for human verification. It is interesting to integrate *r*-HUMO into the existing crowdsourcing platforms in future work. On the crowdsourcing platforms, monetary cost may be a more appropriate metric of human cost than the number of manually inspected pairs used in this paper. On the other hand, re-training the machine learning algorithm after each iteration of manual labeling can usually improve the overall performance of human and machine cooperation. It is a challenging task and deserves an independent investigation in future work.

References

[1] P. Christen, "*Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection,"* Springer Publishing Company, Incorporated*, 2012.*

[2] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 1–16, 2007.*

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

[4] *W. Fan, X. Jia, J. Li, and S. Ma, "Reasoning about record matching rules,"* Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 407–418, 2009.*

[5] *L. Li, J. Li, and H. Gao, "Rule-based method for entity resolution,"* IEEE Transactions On Knowledge And Data Engineering*, vol. 27, no. 1, pp. 250–263, 2015.*

[6] *R. Singh, V. Meduri, A. Elmagarmid, S. Madden, P. Papotti, J.-A. Quiané-Ruiz, A. Solar-Lezama, and N. Tang, "Generating concise entity matching rules," in* Proceedings of the ACM International Conference on Management of data (SIGMOD)*, pp. 1635–1638, 2017.*

[7] *S. Sarawagi and A. Bhamidipaty, "Interactive deduplication using active learning," in* Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 269–278, ACM, 2002.*

[8] *P. Christen, "Automatic record linkage using seeded nearest neighbour and support vector machine classification," in* Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 151–159, ACM, 2008.*

[9] *B. Mozafari, P. Sarkar, M. Franklin, M. Jordan, and S. Madden, "Scaling up crowdsourcing to very large datasets: a case for active learning,"* Proceedings of the VLDB Endowment*, vol. 8, no. 2, pp. 125–136, 2014.*

[10] *G. Li, "Human-in-the-loop data integration,"* Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 2006–2017, 2017.*

[11] *Y. Zhuang, G. Li, Z. Zhong, and J. Feng, "Hike: A hybrid human-machine method for entity alignment in large-scale knowledge bases," in* Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pp. 1917–1926, 2017.*

[12] *A. Arasu, M. Götz, and R. Kaushik, "On active learning of record matching packages," in* Proceedings of the ACM International Conference on Management of data (SIGMOD)*, pp. 783–794, 2010.*

[13] *K. Bellare, S. Iyengar, A. G. Parameswaran, and V. Rastogi, "Active sampling for entity matching," in* Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 1131–1139, 2012.*

[14] *Z. Chen, Q. Chen, and Z. Li, "A human-and-machine cooperative framework for entity resolution with quality guarantees,"* Proceedings of the IEEE 33rd International Conference on Data Engineering (ICDE), Demo paper*, pp. 1405–1406, 2017.*

[15] *Z. Chen, Q. Chen, F. Fan, Y. Wang, Z. Wang, Y. Nafa, Z. Li, H. Liu, and W. Pan, "Enabling quality control for entity resolution: A human and machine cooperation framework,"* IEEE 34th International Conference on Data Engineering (ICDE)*, 2018.*

[16] C. Chai, G. Li, J. Li, D. Deng, and J. Feng, "Cost-effective crowdsourced entity resolution: A partial-order approach," Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pp. 969–984, 2016.*

[17] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: A survey," IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2296–2319, 2016.*

[18] J. Wang, T. Kraska, M. J. Franklin, and J. Feng, "Crowder: Crowdsourcing entity resolution," Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1483–1494, 2012.*

[19] N. Vesdapunt, K. Bellare, and N. Dalvi, "Crowdsourcing algorithms for entity resolution," Proceedings of the VLDB Endowment*, vol. 7, no. 12, pp. 1071–1082, 2014.*

[20] P. Singla and P. Domingos, "Entity resolution with markov logic," IEEE 6th International Conference on Data Mining (ICDM)*, pp. 572–582, 2006.*

[21] S. E. Whang, D. Marmaros, and H. Garcia-Molina, "Pay-as-you-go entity resolution," IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 5, pp. 1111–1124, 2013.*

[22] Y. Altowim, D. V. Kalashnikov, and S. Mehrotra, "Progressive approach to relational entity resolution," Proceedings of the VLDB Endowment*, vol. 7, no. 11, pp. 999–1010, 2014.*

[23] S. Lacoste-Julien, K. Palla, A. Davies, G. Kasneci, T. Graepel, and Z. Ghahramani, "Sigma: Simple greedy matching for aligning large knowledge bases," pp. 572–580, 2013.

[24] A. Gruenheid, D. Kossmann, R. Sukriti, and F. Widmer, "Crowdsourcing entity resolution: When is a=b?," Eth Department of Computer Science Systems Group*, 2012.*

[25] L. Getoor and A. Machanavajjhala, "Entity resolution: theory, practice & open challenges," Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 2018–2019, 2012.*

[26] X. Chu, J. Morcos, I. F. Ilyas, M. Ouzzani, P. Papotti, N. Tang, and Y. Ye, "Katara: A data cleaning system powered by knowledge bases and crowdsourcing," in Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, SIGMOD '15, (New York, NY, USA), pp. 1247–1261, ACM, 2015.*

[27] D. Firmani, B. Saha, and D. Srivastava, "Online entity resolution using an oracle," Proceedings of the VLDB Endowment*, vol. 9, no. 5, pp. 384–395, 2016.*

[28] S. E. Whang, P. Lofgren, and H. Garcia-Molina, "Question selection for crowd entity resolution," Proceedings of the VLDB Endowment*, vol. 6, no. 6, pp. 349–360, 2013.*

[29] C. Gokhale, S. Das, A. Doan, J. F. Naughton, N. Rampalli, J. Shavlik, and X. Zhu, "Corleone: Hands-off crowdsourcing for entity matching," Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pp. 601–612, 2014.*

[30] S. Wang, X. Xiao, and C.-H. Lee, "Crowd-based deduplication: An adaptive approach," Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pp. 1263–1277, 2015.*

[31] V. Verroios, H. Garcia-Molina, and Y. Papakonstantinou, "Waldo: An adaptive human interface for crowd entity resolution," pp. 1133–1148, 2017.

[32] L. Chen, "Cost and quality trade-offs in crowdsourcing," in Encyclopedia of Database Systems*, L. Liu and M. T. Özsu, Eds. Springer-Verlag New York, 2017, pp. 1–3.*

[33] J. Fan, G. Li, B. C. Ooi, K. L. Tan, and J. Feng, "icrowd: An adaptive crowdsourcing framework," in Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pp. 1015–1030, 2015.*

[34] A. Elmagarmid, I. F. Ilyas, M. Ouzzani, N. Tang, and S. Yin, "Nadeef/er: generic and interactive entity resolution," in ACM SIGMOD International Conference on Management of Data*, pp. 1071–1074, 2014.*

[35] C. E. Rasmussen and C. K. Williams, Gaussian processes for machine learning. *MIT press Cambridge, 2006.*

[36] H. M. Markowitz, "Foundations of portfolio theory," Journal of Finance*, vol. 46, no. 2, pp. 469–477, 1991.*

[37] R. T. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," Journal of Banking & Finance*, vol. 26, no. 7, pp. 1443–1471, 2002.*

[38] C. Acerbi and D. Tasche, "Expected shortfall: A natural coherent alternative to value at risk," Economic Notes*, vol. 31, no. 2, pp. 379–388, 2002.*

[39] M. Hababou, A. Y. Cheng, and R. Falk, "Variable selection in the credit card industry." [Online]. Available: https://lexjansen.com/nesug/nesug06/an/da23.pdf.

[40] Y. Zhang, G. Chu, P. Li, X. Hu, and X. Wu, "Three-layer concept drifting detection in text data streams," Neurocomputing*, vol. 260, pp. 393–403, 2017.*