BUBBLE : A Quality-Aware Human-in-the-loop Entity Matching Framework

Naofumi Osawa* Hiroyoshi Ito* Yukihiro Fukushima† Takashi Harada‡ Atsuyuki Morishima*
*University of Tsukuba, Japan †Keio University, Japan ‡Doshisha University, Japan
Email: naofumi.osawa.2020b@mlab.info, ito@slis.tsukuba.ac.jp, fukusima-y@keio.jp,
ushi@slis.doshisha.ac.jp, mori@slis.tsukuba.ac.jp

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9672002

***Abstract*—Entity matching is an issue of interest in information integration and data cleaning. Since the representations of the same entity vary, it is often impossible to fully automate the entity matching and require human inputs. However, to guarantee high-quality entity matching, how to integrate human resources into the entity matching while minimizing the cost of human resources? In this paper, we propose BUBBLE, a novel human-in-the-loop entity matching framework hybridizing Bayesian inference and crowdsourcing. To *guarantee entity matching quality*, Bayesian inference is conducted to determine whether the matching requires crowdsourcing. We show that we can define Bayesian error rate for this problem. For *optimization*, we use metric learning to select the candidate matching pairs by nearest-neighbor search in the learned embedding space, and we construct a k-nearest neighbor graph to avoid the redundant matching. We applied BUBBLE to a bibliographic data matching problem on the National Diet Library. The experimental results show that BUBBLE can assign tasks to humans with higher quality results compared to those of the same number of task assignments to humans. The result also shows that our optimization scheme is effective without sacrificing the quality.**

***Index Terms*—Entity Resolution, Human-in-the-loop, Task Assignment**

I. Introduction

Detection of duplicate data and data cleaning techniques have attracted much attention in the field of big data.Entity matching refers to identifying a set of records, in a database, that refer to the same entity [1], [2] and has been studied extensively in the past. Entity matching is essential for data cleaning and integration of multiple databases, but it is often subject to shaky or missing input during entity creation. Moreover, a completely rule-based approach cannot realize perfect matching for all entities [3], [4]. Therefore, human-in-the-loop is a promising solution where workers can be experts or crowd workers, and crowdsourcing-based approaches have been proposed for entity matching [5]–[7]. However, when the number of records in the database is large, it is critical to identify what tasks should be done by crowd workers, because it is unrealistic to assign to crowd workers all combinations of records and determine whether they refer to the same entity.

Informal statement of the problem. Our research question is whether we can develop a principled framework for human-in-the-loop entity matching or not, which has theoretical background to guarantee the entity matching quality while minimizing the number of assignment to human. In this paper, we answer this question positively, by introducing BUBBLE, a novel, principled human-in-the-loop entity matching framework hybridizing the Bayesian inference and crowdsourcing.

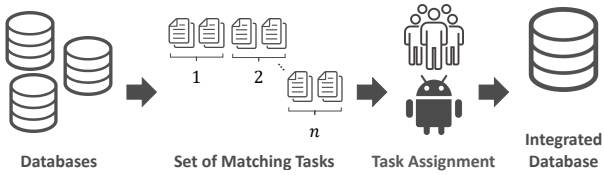


Fig. 1: *Problem Setting* : We generate a set of pair-wise entity matching tasks and execute matching tasks using Bayesian inference and human resources. If Bayesian inference answered uncertainly, we assign the task to human.

BUBBLE consists of two phases. The first phase involves k-nearest neighbor graph construction with metric learning, and the second phase involves iterative k-NN graph contraction with Bayesian inference and crowdsourcing. Figure 2 illustrates the steps of BUBBLE in more detail. First, we embed the data from the database into the vector space via metric learning, considering the similarity of data based on their relationships. Further, we construct a k-nearest neighbor graph connecting the top k nodes with the closest distances, using each embedded data as a node. We perform edge scoring for each edge based on the number of shared nodes and identify the pair with the highest score. The edge pairs are identified by Bayesian inference, and the pairs of records that exceed a certain error rate are matched manually using crowdsourcing.

We apply the proposed method to the problem of bibliographic identification on the general catalog of the National Diet Library to verify the effectiveness of the method. We adopted the Siamese network structure as the metric learning method, and metric learning was performed by minimizing the loss function based on contrastive loss. For entity matching by Bayesian inference, we created multiple probability distributions based on each feature of the records and performed multivariate Bayesian inference.

Experimental results show that the accuracy of matching based on Bayesian inference was over 80%, and blocking using metric learning could generate candidates for candidates with a recall rate of approximately 90% by constructing a k-

NN graph with $k = 5$.

We also showed that by increasing the error rate threshold of the Bayesian inference model in our framework and actively incorporating human into the task can improve the accuracy compared to using only the Bayesian Inference model. Furthermore, we showed that scoring based on the number of shared nodes reduces the number of comparisons by randomly selecting and integrating edges.

Contributions. In this paper, we propose a novel framework, named BUBBLE, which tackles an entity matching problem through a human and machine hybrid approach. Experimental results show that BUBBLE realizes almost 1.0 of F-1 value under assigning only 30% of tasks to crowdsourcing for the bibliographic data matching problem. Our paper makes the following contributions:

- **Accurate and quality-aware.** The framework guarantees matching quality by minimizing the error rate of Bayesian inference.
- **Non-redundant.** KGB, a blocking method using distance learning and k-nearest neighbor graph, reduces unnecessary matching and the number of comparisons.
- **Cost effective task assignment.** The method allows scalable adjustment of the matching cost by a threshold value according to the problem setting.

This paper is organized as follows. In Section 2, we discuss related research, and in Section 3, we propose our novel human-in-the-loop entity matching framework. In Section 4, we present an entity matching experiment using the National Diet Library's General Catalog. Finally, we discuss the experiments in Section 5 and conclude the paper in Section 6.

II. Related Work

In this section, we discuss related works on (1) Active learning, (2) Blocking methods for entity matching, and (3) Human-in-the-loop entity matching.

A. Active learning

Active learning exists as a hybrid approach that combines human intelligence into machine learning [8]. This is one of the fields of machine learning, where instead of labeling all data, develop a strategy of "which data should be labeled" to increase the learning efficiency. Their goal is to improve the recognition rate of machine learning models, and they generally do not consider quality guarantees or human resource constraints. Our goal is to achieve perfect matching with minimal human intervention, considering cost effectiveness. Therefore, our research objective is different from the setting of active learning.

B. Blocking methods for entity matching

Blocking can reduce the number of redundant matches in large databases, by splitting similar data in advance. This is a crucial technique in the context of entity matching [9]–[12]. This paper proposes an approach that utilizes the embedding and k-NN graph construction as a blocking technique, reducing redundant task pairs by iterative graph contraction.

C. Human-in-the-loop entity matching

There are several approaches that iteratively generates tasks to human to realize entity matching [6], [13], [14]. Harada et al. [15] proposed a bibliographic identification method using crowdsourcing and achieved high quality results. Doan et al. have developed two entity matching systems, Magellan [16] and Corleone/Falcon [17]–[19]. These are crowdsourcing-based entity matching tools that iteratively matches entities by crowd worker while estimating the difficulty of the matching. Also another human-in-the-loop system was developed [5]. That applies humans to the two steps of rule-based pruning of matching candidates and cloud-based refinement of those candidates.

However, this approach is not cost-effective when the records in the database are very large since these approaches only rely on human resources. Our main research objective is to reduce the number of human tasks by performing hybridization of machine-based and human-based matching while minimizing the error rate.

III. Proposed framework: BUBBLE

In this section, we present the proposed framework BUBBLE. Algorithm 1 shows the process of the framework.

In BUBBLE, we first embed each data $(\boldsymbol{a}, id)$ from the database $\mathcal{S}$ into the metric space. The embedding into the metric space is done using the metric learning method described in Section 4.2. Further, we construct a k-nearest neighbor graph that connects the top k nodes with the closest distance in which each embedded data as a node. A score is then calculated for each node, and the nodes with the highest scores are identified as candidate pairs to obtain a set of identical data. If the nodes match, the graph is contracted by merging the nodes; otherwise, the edges between the nodes are cut off. The above operations are repeated until the nodes become independent. By connecting nodes that are close to each other in the embedding space, nodes that have similar meanings are connected. This plays the role of blocking in data integration. In addition, we used an edge scoring function based on the number of overlapping neighboring nodes to reduce the number of comparisons, and decisions were made based on the highest value. Table I shows the notation of the symbols used in this paper.

Node pairs are identified using Bayesian inference, which is constructed by the binary classification model described in Section 4.3. If the Bayesian Inference has an error rate of less than τ, the identification is based on the result of the Bayesian Inference. If the error rate is greater than τ, the identification is done by the worker as a crowdsourcing task. Each component of the algorithm is explained in the following sections.

A. Metric Learning

For metric learning, a function $M_\Theta : \mathcal{S} \to \mathbb{R}^n$ is learned to map the data in the database into the embedding space, where Θ is a parameter set of the mapping function of metric learning. The basic idea of metric learning is that data belonging to the same class should be closer together in the

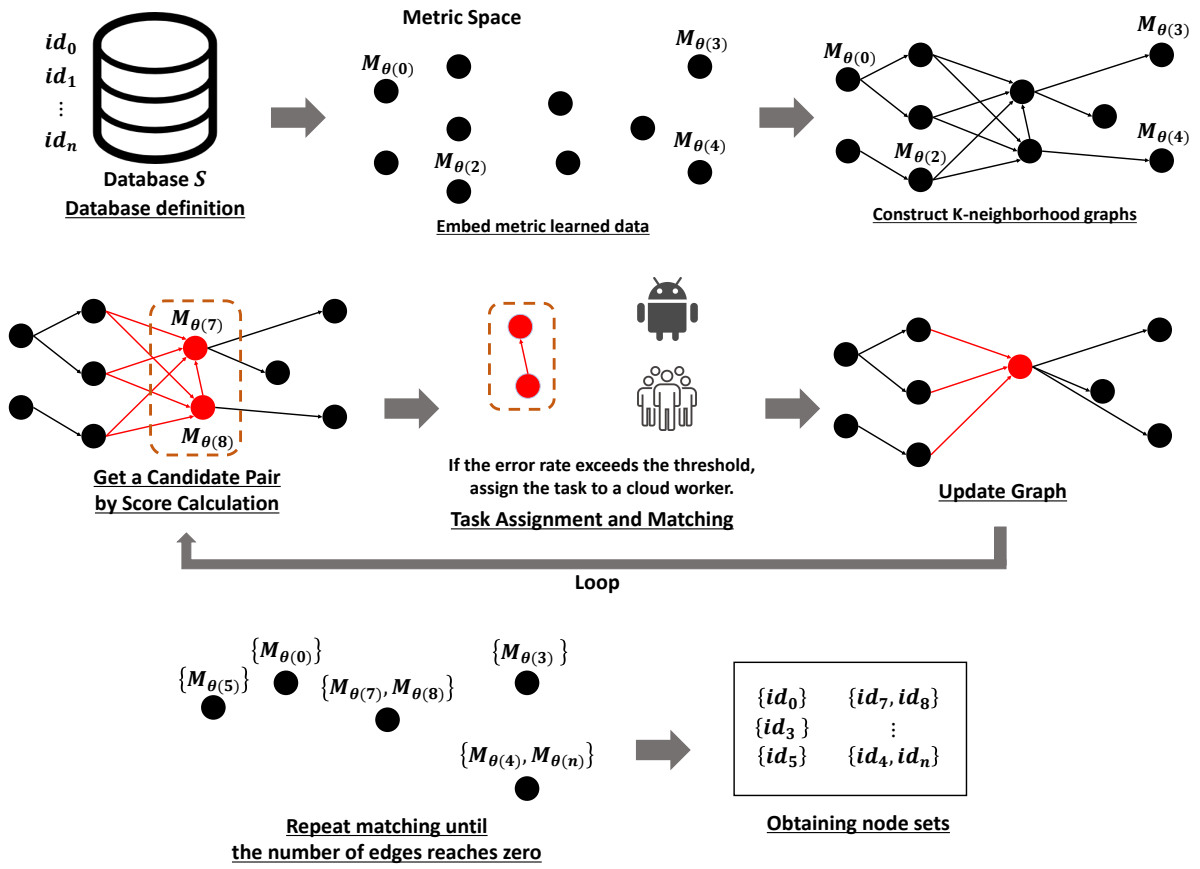


Fig. 2: Steps in BUBBLE

TABLE I: symbol definitions

symbol	definition
$\mathcal{S} = \{S_i\}$	Database
$S_i = (\boldsymbol{a}, id)$	A data ($\boldsymbol{a} : features$, $id : number$)
$\mathcal{D} \subseteq \mathcal{S} \times \mathcal{S}$	A set of pairs of data matches
$\mathbf{x}_{i,j} \in \mathbb{R}^n$	Similarity measures between S_i and S_j
$M_\Theta \colon \mathcal{S} \to \mathbb{R}^n$	Metric learning
$Nearest(S_i, k)$	Top k data that are close to i
$\mathcal{G}_k = (\mathcal{V}, \mathcal{E})$	k-NN graph of the data
$\mathcal{V} \subseteq \mathcal{P}(\mathcal{S})$	Node sets in k-NN graphs
$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$	Edge sets in k-NN graphs
$BI \colon \mathcal{V} \times \mathcal{V} \to [0,1]^2$	Bayesian Inference
$Human \colon \mathcal{V} \times \mathcal{V} \to 0, 1$	Crowdsourcing Tasks
τ	Threshold about error rate

embedding space, whereas data belonging to different classes should be farther apart. Given a pair of matched data $\mathcal{D}$, we learn the metric learning parameter Θ by solving the following optimization problem.

$$\Theta = \arg\min_{\Theta} \sum_{(S_i,S_j)\in\mathcal{D}} Dist_1(M_\Theta(S_i), M_\Theta(S_j)) - \sum_{(S_i,S_j)\notin\mathcal{D}} Dist_2(M_\Theta(S_i), M_\Theta(S_j)) \quad (1)$$

Here, $Dist_1$ and $Dist_2$ represent the distances in the m-dimensional embedding space.

B. Construction of a k-NN graph based on metric learning

We construct a k-nearest neighbor graph for the data points in the embedding space obtained by metric learning. We define a k-NN graph as follows.

$$\mathcal{G}_k = (\mathcal{V}, \mathcal{E}) \quad (2)$$

Here, $\mathcal{V} \subseteq \mathcal{P}(\mathcal{S})$ represents the node set, and each node $v \subseteq \mathcal{S}$ is a set of data. In the initial state of the k-NN graph, each node v is a set consisting of only one data point. The edge set $\mathcal{E}$ is initialized as follows.

$$\mathcal{E} = \{(\{S_i\}, \{S_j\}) \mid S_j \in Nearest(S_i, k),\ S_i \in \mathcal{S}\} \quad (3)$$

Here, the edges are directed edges and $Nearest(S_i, k) \subseteq \mathcal{S}$ represents the top k data sets in the embedding space that are close to the data S_i. In the proposed framework, node pairs that are determined to be identical are contracted in the k-NN graph, and the edges of node pairs that are determined to be non-identical are deleted from the graph. Constructing cliques with k-neighborhood graphs allows for leak-free matching. In this paper, we denote the contraction of a graph $\mathcal{G}$ by an edge (v_i, v_j) as $\mathcal{G} \setminus (v_i, v_j)$.

Algorithm 1 Framework of BUBBLE

Input: Database $\mathcal{S}$, Pair $\mathcal{D}$
Output: A set of nodes in the k-NN Graph $\mathcal{V}$

Learn Metric M_Θ
Construct k-NN Graph $\mathcal{G}_k = (\mathcal{V}, \mathcal{E})$
while $\mathcal{E} \neq \varnothing$ **do**
  $(v_i, v_j) \leftarrow \arg\max_{(v_i,v_j)\in\mathcal{E}} \boldsymbol{Score}(v_i, v_j)$
  $(P(Same \mid \mathbf{x}_{i,j}), P(Not\text{-}Same \mid \mathbf{x}_{i,j})) \leftarrow BI(v_i, v_j)$
  if $1 - P(Same \mid \mathbf{x}_{i,j}) \leq \tau$ **then**
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(v_i, v_j)\}$
    $\mathcal{G} \leftarrow \mathcal{G} \setminus \{(v_i, v_j)\}$
  else if $1 - P(Not\text{-}Same \mid \mathbf{x}_{i,j}) \leq \tau$ **then**
    $\mathcal{E} \leftarrow \mathcal{E} \setminus \{(v_i, v_j)\}$
  else
    Human score $\leftarrow Human(v_i, v_j)$
    if Human score is 1 **then**
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(v_i, v_j)\}$
      $\mathcal{G} \leftarrow \mathcal{G} \setminus \{(v_i, v_j)\}$
    else
      $\mathcal{E} \leftarrow \mathcal{E} \setminus \{(v_i, v_j)\}$
    end if
  end if
end while

C. KGB : k-nearest neighbor graph blocking

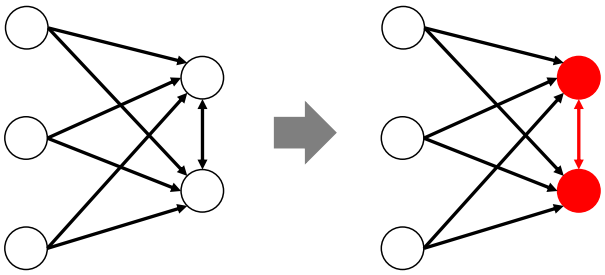


Fig. 3: *Basic idea of KGB* : When there are many sharing nodes between an edge, (1) more edges are eliminated by contracting the edge, and (2) the probability that two nodes are the same entity is high.

In this section, we propose k-nearest neighbor graph blocking: $\boldsymbol{KGB}$, which is a blocking method for data pairs to reduce the number of comparisons. To minimize the number of pairs to be compared, we calculate the priority among the nodes to be identified. This makes it possible to realize blocking that considers the features of the data. The basic idea is that in a k-NN graph, the more neighbors two nodes share, the more likely they are to refer to the same entity. Moreover, the larger the number of edges that shrink when the two nodes are merged, the smaller the number of data pairs that need to be compared.

The score for an edge (v_i, v_j) is calculated as follows.

$$\boldsymbol{Score}(v_i, v_j) = |\mathcal{N}(v_i) \cap \mathcal{N}(v_j)|, \tag{4}$$

where $\mathcal{N}(v_i) \subseteq \mathcal{S}$ is the set of nodes with edges for node v_i. To minimize the number of pairs to be compared, we calculate the priority between the nodes to be identified.

D. Matching based on Bayesian identification rules and task assignment

We use Bayesian inference to match data. Bayesian inference refers to the computation of the conditional and marginal distributions of interest from a given simultaneous distribution. In this study, the multivariate Bayesian discriminant rule is used to make matching decisions. This rule is calculated by ensembling the multiple distance metrics of data, which are the string similarities calculated from each metadata. We use the following Bayesian discrimination rule to make matching decisions.

$$P\left(Same \mid \mathbf{x}_{i,j}\right) = \frac{\prod_{l=1}^{n} p\left(x_{i,j}^{(l)} \mid Same\right) P(Same)}{\sum_{Class\in\{Same,\ Not\text{-}Same\}} \prod_{l=1}^{n} p\left(x_{i,j}^{(l)} \mid Class\right) P(Class)}, \tag{5}$$

$$P\left(Not\text{-}Same \mid \mathbf{x}_{i,j}\right) = \frac{\prod_{l=1}^{n} p\left(x_{i,j}^{(l)} \mid Not\text{-}Same\right) P(Not\text{-}Same)}{\sum_{Class\in\{Same,\ Not\text{-}Same\}} \prod_{l=1}^{n} p\left(x_{i,j}^{(l)} \mid Class\right) P(Class)}, \tag{6}$$

where $\mathbf{x}_{i,j} = \left(x_{i,j}^{(1)}, ..., x_{i,j}^{(n)}\right)^\top \in \mathbb{R}^n$ is the vector of the similarity measures between the nodes v_i and v_j.

The identification rule based on Bayes' theorem minimizes the error rate. When considering the classification problem of two classes, i.e., $Same, Not\text{-}Same$, which is the problem setting of this study if the class of agreement $P(Same \mid \mathbf{x}_{i,j})$ is greater than that of disagreement $P(Not\text{-}Same \mid \mathbf{x}_{i,j})$, as shown earlier, the observed data $\mathbf{x}_{i,j}$ is classified into the class $Same$, and if the opposite is true, it is classified into the class $Not\text{-}Same$.

The probability of making a wrong decision based on the identification rule $\varepsilon(\mathbf{x}_{i,j})$ is the smaller of the posterior probabilities, i.e.

$$\varepsilon(\mathbf{x}_{i,j}) = \min[P(Same \mid \mathbf{x}_{i,j}),\ P(Not\text{-}Same \mid \mathbf{x}_{i,j})]. \tag{7}$$

This is called the conditional Bayesian error rate. This is expressed as the expected value of the conditional Bayesian error rate and is calculated as follows:

$$\varepsilon^* = \mathbb{E}[\varepsilon(\mathbf{x}_{i,j})] = \int_{R_{Same}+R_{Not\text{-}Same}} \varepsilon(\mathbf{x}_{i,j}) p(\mathbf{x}_{i,j})\ d\mathbf{x}_{i,j} \tag{8}$$

$$= \int_{R_{Not\text{-}Same}} p(\mathbf{x}_{i,j} \mid Same) P(Same)\ d\mathbf{x}_{i,j} + \int_{R_{Same}} p(\mathbf{x}_{i,j} \mid Not\text{-}Same) P(Not\text{-}Same)\ d\mathbf{x}_{i,j}, \tag{9}$$

where $R_{Same} = \{\mathbf{x}_{i,j} \in \mathbb{R}^n \mid p(\mathbf{x}_{i,j} \mid Same)P(Same) > p(\mathbf{x}_{i,j} \mid Not\text{-}Same)P(Not\text{-}Same)\}$ and $R_{Not\text{-}Same} =$

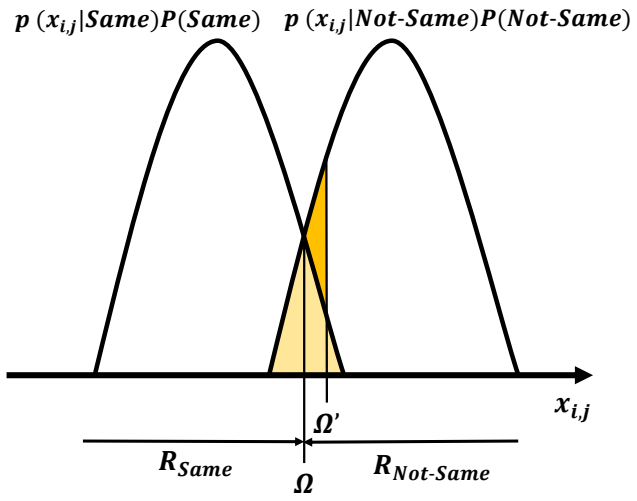


Fig. 4: *Conceptual illustration of discriminant bounds and Bayesian error rates.* Discriminant boundary Ω gives a minimal error rate of decision ε^*, which is shown as a yellow area.

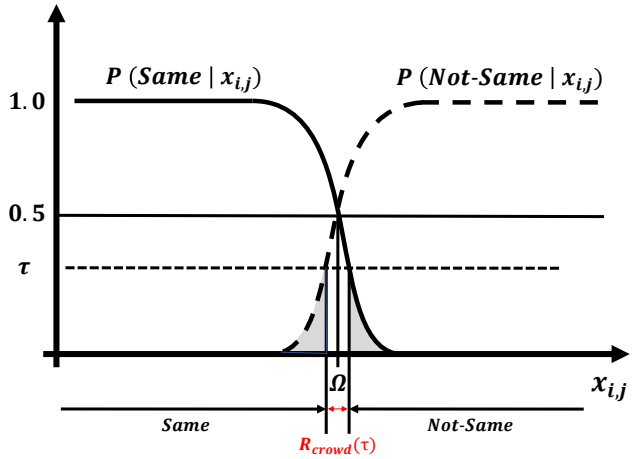


Fig. 5: *Task Assignment Area*: If the error rate exceeds τ, assign the task to human.

$\{\mathbf{x}_{i,j} \in \mathbb{R}^n \mid p(\mathbf{x}_{i,j} \mid Same)P(Same) < p(\mathbf{x}_{i,j} \mid Not\text{-}Same)P(Not\text{-}Same)\}$. This integral is shown in Figure 4. The Bayesian discriminant boundary is at $\Omega = \{\mathbf{x}_{i,j} \in \mathbb{R}^n \mid p(\mathbf{x}_{i,j} \mid Same)P(Same)\} = p(\mathbf{x}_{i,j} \mid Not\text{-}Same)P(Not\text{-}Same)\}\}$, and this integral corresponds to the area of the light yellow area in Figure 4. As the identification boundary shifts to Ω', the Bayesian error rate increases by the amount indicated by dark yellow in the figure. This property ensures that the Bayesian identification rule minimizes the error rate.

Based on Bayesian discriminant rules, it is possible to reject a decision if the error rate is larger. The basic idea of the reject rule is to reject a decision when the error rate for the class to be identified is greater than or equal to the rejection threshold τ. Lowering the rejection threshold τ increases the number of rejected tasks and decreases the probability of incorrectly recognizing a class. In this study, we assign this rejected data as a task to crowdsourcing, which has high decision-making capability. The region that the task are assigned to crowdsourcing is defined as follows:

$$R_{crowd}(\tau) = \{\mathbf{x}_{i,j} \mid \epsilon(\mathbf{x}_{i,j}) \geq \tau\}. \tag{10}$$

The illustration of $R_{crowd}(\tau)$ is shown in Figure 5. The number of tasks in crowdsourcing can be controlled by changing the threshold value τ according to the problem setting. Thus, Bayesian inference can always guarantee the best matching quality because the error rate is minimized.

IV. Experiments

In this section, we describe the experiments conducted to validate the usefulness of the proposed method using real-world data. We performed three types of evaluation experiments: (1) initial recall of paired candidates by applying KGB, (2) estimation of the distribution using the maximum likelihood estimation, and (3) entity matching experiments.

A. Dataset

We consider the problem of bibliographic data matching in the Japanese bibliographic database $\mathcal{S}$ created by the General Catalogue of the National Diet Library. The bibliographic database $\mathcal{S}$ is a mixture of bibliographic databases from multiple libraries. This is synonymous with multiple databases with a common schema.

Each bibliographic datum S_i has the following features: (1) title, (2) volume number, (3) author, (4) publisher, (5) ISBN, (6) page number, and (7) size.

ISBNs that identify the bibliographic information of books[1] are assigned unique numbers .

Therefore, bibliographic entries with the same ISBN are normally considered to be the same. However, because acquiring an ISBN is expensive, ISBNs for books that are no longer in circulation are sometimes used for new books. In addition, ISBNs have only been used since 1981, and books written before the 80s do not have ISBNs. Thus, we use ISBNs for the KGB described in Section 4.2 and the Bayesian identification rules constructed in section 4.3 but not for the actual matching problem.

B. Initial recall of paired candidates by applying KGB

We evaluated whether the graph obtained via KGB described in Section 3.4 possesses the pairs of nodes to be matched and whether the number of comparisons can be reduced. The comparison method is Naive method, that extracts nodes randomly after constructing the KGB, and matches the pairs of edges that the nodes have.

Metric learning for KGB. We describe the metric learning performed to adopt the KGB. Metric learning is a method to learn measures such as the similarity and distance between

[1]https://isbn.jpo.or.jp/

TABLE II: Examples of bibliographic data pairs

Books	Title	Volume Number	Author	Publisher	Page	Size
S_1	若おかみは小学生!		亜沙美,令丈ヒロ子	講談社	215p	18cm
S_2	若おかみは小学生!：花の湯温泉ストーリー 1	[PART1]	亜沙美,令丈ヒロ子	講談社	215p	18cm

data. If a feature space that takes semantic distance into account can be learned, unknown data can be handled robustly.

We adopt the *Siamese network* [20] as the metric learning model M_Θ. The loss function for learning the parameter Θ is called *Contrastive Loss* [21]. The *Contrastive Loss* is expressed as the following equation:

$$\boldsymbol{Loss} = \frac{1}{2}\left(\sum_{(S_i,S_j)\in\mathcal{D}} Dist_{euc}(S_i,S_j)^2 - \sum_{(S_i,S_j)\notin\mathcal{D}} \max\left(m - Dist_{euc}(S_i,S_j)\right),0)^2\right), \quad (11)$$

where $m \in \mathbb{R}$ and

$$Dist_{euc}(S_i,S_j) = \|M_\Theta(S_i) - M_\Theta(S_j)\|_2. \quad (12)$$

We used Keras[2] to train the parameters.

The model is comprised of an input layer, all coupling layers 1, dropout layer, all coupling layers 2, dropout layer, all coupling layers 3, dropout layer, and output layer. For the all-coupled layer and the output layer, we used the ReLu function as the activation function and the *Contrastive Loss* as the loss function.

Comparison of the frequency and recall. First, we show the change in the number of comparisons of the proposed method. Since the nodes with high $\boldsymbol{Score}$ share many edges, we can reduce the number of comparisons by preferentially identifying them. We also confirmed that the increase in the number of comparisons is smaller than that of the naive method when the number of nearest neighbor searches k is increased.

Results. Figure 6 (a) and (b) show that the rate of edge reduction decreased from the latter half of the matching process. This indicates that the scores become uniform, and the graph does not shrink further irrespective of which pairs are matched.

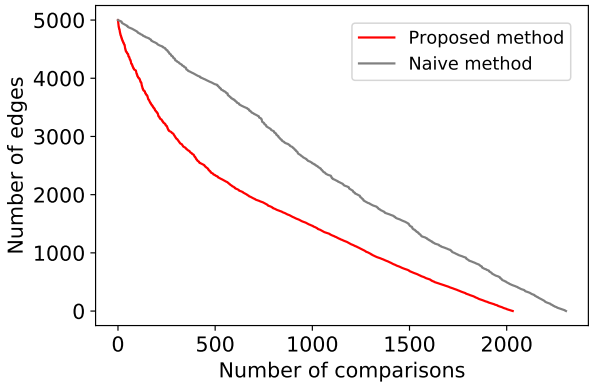
Further, we discuss that the reproduction rate of pairs should be matched. We make identification decisions on the edges of the k-NN graph. Therefore, pairs of records to be matched should occupy all edges of the k-NN graph. Figure 7 shows the reproduction rate of the matching pairs when constructing the k-NN graph. The reproduction rate indicates the number of edges set in the k-NN graph included in $\mathcal{D}$, which is the set of matched pairs of data.

$$\mathcal{E}_{init} = \{(S_i,S_j) \mid S_j \in Nearest(S_i,k), S_i \in \mathcal{S}\} \quad (13)$$

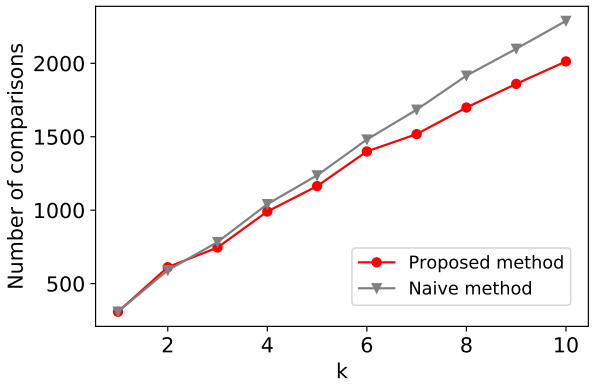
$$Init\ Recall = \frac{|\mathcal{D}\cap\mathcal{E}_{init}|}{|\mathcal{D}|} \quad (14)$$

Experiments with several datasets confirmed that the reproduction rate reaches approximately 90% for $k = 5$. This indicates that appropriate data embedding can be obtained by metric learning, and the number of comparisons can be reduced by setting the value of k appropriately. We showed KGB to reduce the number of comparisons of matching tasks. By metric learning and k-NN graph construction, the number of comparisons is significantly reduced while maintaining the reproducibility of the matching candidates, compared to the non-blocking method that compares all combinations. Furthermore, by preferentially matching node pairs with many shared edges, we can achieve entity matching with an even smaller number of comparisons.

[2]https://keras.io/ja/



(a) Change in number of edges



(b) Change in comparison frequency

Fig. 6: *The number of comparisons plotted against the number of edges and the number of nearest neighbors k* : (a) the number of comparisons was reduced compared to the naive method, where nodes were selected randomly by contracting the graph from the node with the highest $\boldsymbol{Score}$. (b) When the number of nearest neighbors k was increased, there was a significant difference in the cumulative number of comparisons.

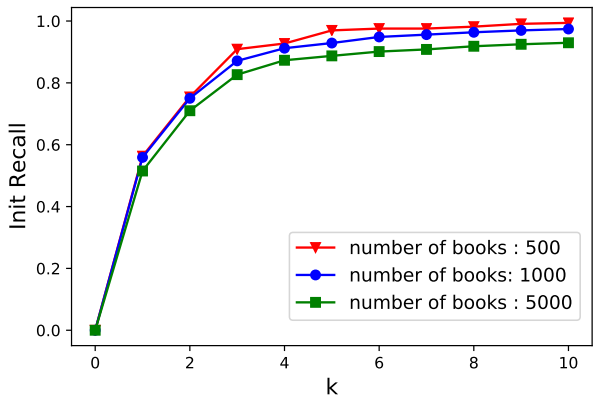


Fig. 7: *Initial Recall of identified pairs when constructing k-NN graphs*: The larger k is, the higher the *Initial Recall* of the pairs to be identified.

C. Generating probability distributions by maximum likelihood estimation

In this section, we describe the generation of probability distributions using maximum likelihood estimation, which corresponds to the classed conditional probability $\prod_{l=1}^{n} p(x_{i,j}^{(l)} \mid Same), \prod_{l=1}^{n} p(x_{i,j}^{(l)} \mid Not\text{-}Same)$ of Bayes' theorem described in Section 3.1. To estimate the class using Bayes' discriminant rule, we use (1) variance representation, (2) Jaro-Winkler [22] distance, (3) Levenshtein [23] distance, and (4) SequenceMatcher distance [3]. Each of the distance metrics represents each dimension $x_{i,j}$. We performed multivariate Bayesian estimation by creating probability distributions for them. The conditional probabilities with classes in the entity matching problem has $Same$ and $Not\text{-}Same$. We used the ISBNs of the bibliographic data to compute the similarity between pairs whose ISBNs match and mismatched pairs whose ISBNs do not match and satisfies following condition:

$$1.0 > 1.0 - Distance(S_i, S_j) \geq P, \tag{15}$$

Here, $Distance(S_i, S_j)$ is a difference measure and we used the Python standard library *difflib*. In this experiment we set $P = 0.6$.

The bibliographic data $x_{i,j}^{(l)}$ $(l \in \{1, 2, 3, 4\}$ are sampled independently from the true distribution. In this study, we fit the probability density function to the sampled data distribution $p(x_{i,j}^{(l)} \mid \Theta_l)$, where Θ_l are a set of parameters. The parameters of the probability density functions Θ are learned by maximum likelihood estimation from the obtained pairs of data $\mathcal{D}$. The optimization problem is

$$\Theta_l^* = \arg\max_{\Theta_l} \prod_{(i,j)\in\mathcal{D}} p\left(x_{i,j}^{(l)} \mid \Theta_l\right). \tag{16}$$

In this study, we used the Python library SciPy[4] to perform maximum likelihood estimation. The estimation was done by fitting a gamma distribution to each distance. The probability density function of gamma is

$$p(x; a, b) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}. \tag{17}$$

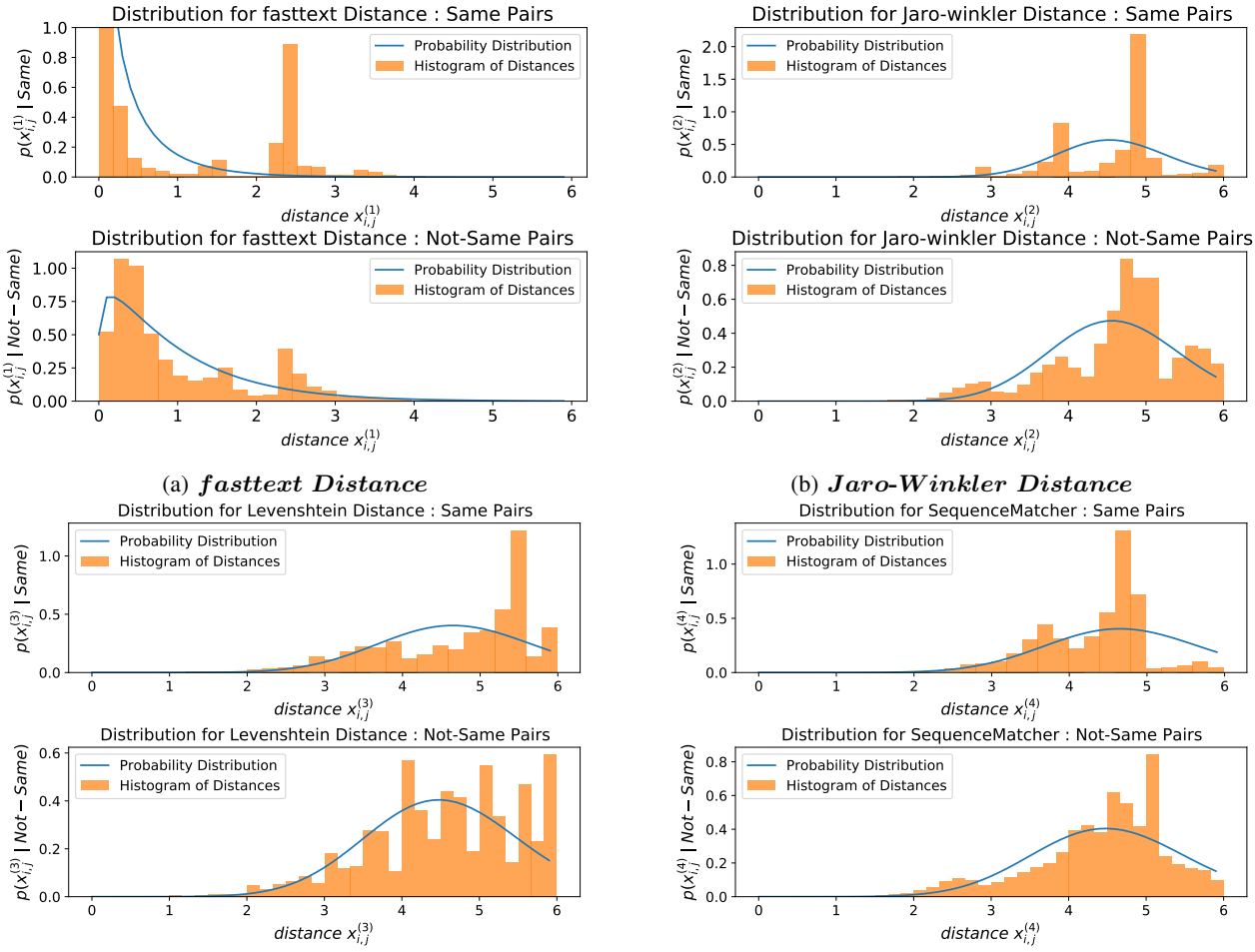
In this study, We constructed a multivariate Bayesian estimation model that combines these factors and applied it to entity matching. Here, the class conditional probabilities are assumed to be conditionally independent. The prior distribution $P(Same), P(Not\text{-}Same)$ is considered to be an uninformed prior. Figures 8 shows the histogram of each distance and their fitted probability distribution in the dataset. As we can see, each distribution for the same and not-same pairs is different, enabling us to make the Bayesian inference-based classifier.

D. Entity matching experiment

We conducted an experiment of matching bibliographic data to see whether BUBBLE can choose appropriate human tasks for the quality. In the experiment, we randomly selected 500 bibliographic data and constructed a k-NN graph with the number of nearest neighbors k set to 5. The following shows the number of tasks and the F-1 measure of the proposed method when the threshold of the rejected area τ is changed. It is assumed here that crowdsourcing decisions always return correct answers. The comparison method is Random method, that randomly assigns matching tasks to humans.

Result. Figure 9 shows that Bayesian inference can be used to assign tasks to humans that are likely to be misjudged. We prioritized the matching tasks, and we crowdsourced approximately 30% of the total matching tasks; consequently, we achieved an overall matching accuracy of almost 1.0. By reducing the size of τ, we were able to encourage human intervention and improve the matching accuracy of the framework. Figure 10 shows the comparison of the matching accuracy between the random task assignment and the BUBBLE error rate based assignment method. In our proposed method, we show that by assigning matching tasks to human based on the Bayesian error rate, we can obtain high matching accuracy with a smaller number of task assignments. In addition, since the F-1 score increases by a large percentage, we can conclude that the matching ranking by KGB is appropriate.

[3]https://docs.python. org/en/3/library/difflib.html
[4]https://www.scipy.org/index.html



(a) ***fasttext Distance*** (b) ***Jaro-Winkler Distance***

(c) ***Levenshtein Distance*** (d) ***SequenceMatcher Distance***

Fig. 8: *Probability distribution for each of distance metrics:* Each figure shows the histogram of data that is $Same$ and $Not\text{-}Same$ and the probability density functions $p(x_{i,j}^{(l)} \mid Same)$ and $p(x_{i,j}^{(l)} \mid Not\text{-}Same)$, respectively.

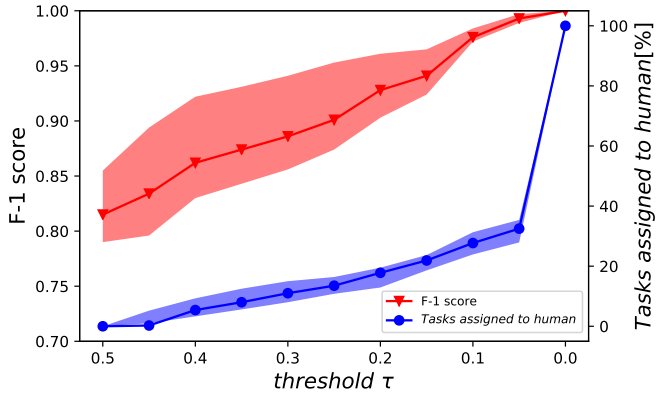


Fig. 9: *Change in F-1 value and task fraction for varying BI threshold* τ: In BUBBLE, the smaller τ is, the greater the number of crowdsourcing tasks, and the higher the F-1 value.

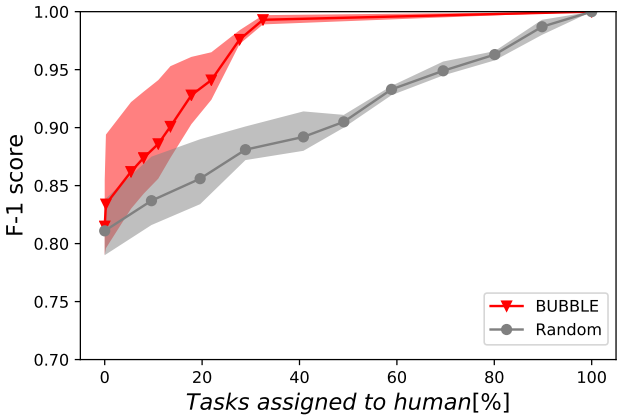


Fig. 10: *Percentage of tasks and change in F-1 value of* BUBBLE: Based on the error rate, the proposed method achieves high matching accuracy by requesting about 30% of all matching tasks to human. In the comparison method, tasks are requested randomly, resulting in non-redundant task assignment, which is difficult to improve the matching accuracy.

V. Conclusion

We proposed a human and machine hybrid entity matching framework BUBBLE, based on Bayesian inference and crowdsourcing and applied it to a real-world entity matching problem. We explained that BUBBLE uses graph theory as a background for acquiring matching candidates and that Bayesian discriminative rules guarantee the accuracy of matching due to the minimum error rate. The results of the experiments can be summarized as follows. (1) Blocking with KGB reduces the number of comparisons compared to naive methods. (2) Multivariate Bayesian estimation using the features of metadata as probability distributions realizes almost 1.0 of F-1 value under assigning only 30% of tasks to crowdsourcing. (3) Reject-based task assignment shows that assigning non-redundant tasks to humans improves the matching ability of the framework.

We are considering incorporating the following in BUBBLE in future studies. (1) Consideration of loss. (2) In the loop for Bayesian inference models. (3) Task assignment considering spam workers. To consider the loss, there is a difference in the risk obtained by the domain between agreement and disagreement. Therefore, it is necessary to provide an appropriate loss in matching. Regarding the Bayesian inference models, we will feed back the results of crowdsourcing decisions to the conditional probabilities. We plan to reduce the number of tasks assigned to humans by further improving the accuracy of inference through human judgment. For task assignment to spammer, in this experiment, we assumed that the crowdsourced workers always behave correctly, but in an actual crowdsourcing environment, there may be spammer or human who make wrong decisions. In the future, we plan to construct a framework that is robust against such uncertain behavior of human.

Acknowledgment

This work was partially supported by JST CREST Grant Number JPMJCR16E3, Japan and JSPS KAKENHI Grant Number 20K23337,19H04428 Japan.

References

[1] V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, and K. Stefanidis, "An overview of end-to-end entity resolution for big data," *ACM Comput. Surv.*, vol. 53, Dec. 2020.

[2] H. Köpcke and E. Rahm, "Frameworks for entity matching: A comparison," *Data & Knowledge Engineering*, vol. 69, no. 2, pp. 197–210, 2010.

[3] E. Rahm and H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Eng. Bull.*, vol. 23, pp. 3–13, 01 2000.

[4] M. A. Hernández and S. J. Stolfo, "The merge/purge problem for large databases," in *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, SIGMOD '95, (New York, NY, USA), p. 127–138, Association for Computing Machinery, 1995.

[5] G. Li, "Human-in-the-loop data integration," *Proc. VLDB Endow.*, vol. 10, p. 2006–2017, Aug. 2017.

[6] J. Wang, T. Kraska, M. J. Franklin, and J. Feng, "Crowder: Crowdsourcing entity resolution," 2012.

[7] S. E. Whang, P. Lofgren, and H. Garcia-Molina, "Question selection for crowd entity resolution," *Proc. VLDB Endow.*, vol. 6, p. 349–360, Apr. 2013.

[8] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[9] J. Nin, V. Muntés-Mulero, N. Martínez-Bazan, and J.-L. Larriba-Pey, "On the use of semantic blocking techniques for data cleansing and integration," *Database Engineering and Applications Symposium, International*, vol. 0, 09 2007.

[10] S. E. Whang, D. Menestrina, G. Koutrika, M. Theobald, and H. Garcia-Molina, "Entity resolution with iterative blocking," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, SIGMOD '09, (New York, NY, USA), p. 219–232, Association for Computing Machinery, 2009.

[11] A. Das Sarma, A. Jain, A. Machanavajjhala, and P. Bohannon, "An automatic blocking mechanism for large-scale de-duplication tasks," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, (New York, NY, USA), p. 1055–1064, Association for Computing Machinery, 2012.

[12] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng, "Leveraging transitive relations for crowdsourced joins," 2014.

[13] G. Li, Y. Zheng, J. Fan, J. Wang, and R. Cheng, "Crowdsourced data management: Overview and challenges," in *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, (New York, NY, USA), p. 1711–1716, Association for Computing Machinery, 2017.

[14] G. Demartini, D. Difallah, and P. Cudre-Mauroux, "Large-scale linked data integration using probabilistic reasoning and crowdsourcing," *The VLDB Journal*, vol. 22, 10 2013.

[15] T. Harada, Y. Fukushima, S. Sato, M. Tsuruta, R. Yoshimoto, and A. Morishima, "Advancement of bibliographic identification using a crowdsourcing system," *Proceedings of the 9th Asia-Pacific Conference on Library & Information Education and Practice(A-LIEP 2019)*, pp. 71–82, 11 2019.

[16] P. Konda, S. Das, P. Suganthan G. C., A. Doan, A. Ardalan, J. R. Ballard, H. Li, F. Panahi, H. Zhang, J. Naughton, S. Prasad, G. Krishnan, R. Deep, and V. Raghavendra, "Magellan: Toward building entity matching management systems," *Proc. VLDB Endow.*, vol. 9, p. 1197–1208, Aug. 2016.

[17] C. Gokhale, S. Das, A. Doan, J. F. Naughton, N. Rampalli, J. Shavlik, and X. Zhu, "Corleone: Hands-off crowdsourcing for entity matching," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, (New York, NY, USA), p. 601–612, Association for Computing Machinery, 2014.

[18] S. Das, P. S. G.C., A. Doan, J. F. Naughton, G. Krishnan, R. Deep, E. Arcaute, V. Raghavendra, and Y. Park, "Falcon: Scaling up hands-off crowdsourced entity matching to build cloud services," in *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, (New York, NY, USA), p. 1431–1446, Association for Computing Machinery, 2017.

[19] A. Doan, A. Ardalan, J. Ballard, S. Das, Y. Govind, P. Konda, H. Li, S. Mudgal, E. Paulson, G. C. P. Suganthan, and H. Zhang, "Human-in-the-loop challenges for entity matching: A midterm report," in *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, HILDA'17, (New York, NY, USA), Association for Computing Machinery, 2017.

[20] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, (San Francisco, CA, USA), p. 737–744, Morgan Kaufmann Publishers Inc., 1993.

[21] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1735–1742, 2006.

[22] W. E. Winkler, "String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage," in *Proceedings of the Section on Survey Research*, pp. 354–359, 1990.

[23] V. I. LEVENSHTEIN, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Physics Doklady*, vol. 10, pp. 707–710, 1966.