Information Research - Vol. 30 No. iConf (2025)

Inconsistency-driven approach for human-in-the-loop entity matching

Hiroyoshi Ito, Takahiro Koizumi, Ryuji Yoshimoto, Yukihiro Fukushima, Takashi Harada, and Atsuyuki Morishima

DOI: https://doi.org/10.47989/ir30iConf47140

Abstract

Introduction. Entity matching is a fundamental operation in a wide range of information management applications and a tremendous number of methods have been proposed to address the problem. Human-in-the-loop entity matching is a human-AI collaborative approach which is effective when the data for entity matching is incomplete or requires domain knowledge. A typical human-in-the-loop approach is to allow a machine-learning-based matcher to ask humans to match entities when it cannot match them with high confidence. However, ML-based matchers cannot avoid the unknown-unknown problem, i.e., they can resolve the entities incorrectly with high confidence.

Method. This paper addresses an inconsistency-based method to deal with this problem. The method asks humans to resolve the entities when we find inconsistency in the transitivity property behind entity matching. For example, if a matcher returns a positive result only for two combinations among three entities, the result is inconsistent.

Analysis. This paper shows an implementation of our idea in similarity-based blocking method and Bayesian inference and explains the result of an extensive set of experiments that reveals how and when the method is effective.

Results. The result showed that the inconsistency-based sampling selects very different entity pairs compared to other sampling strategies and that a simple hybrid strategy performs well in many practical situations.

Conclusion. The results indicate our approach complements any existing matcher that can cause the unknown-unknown problem in entity matching.

Introduction

Entity matching is a fundamental operation for objects in information management applications, such as bibliographic records, names (Cohen, et al., 2003), entities that appear in ontology (Xu, et al., 2008), texts and other data collections (Jaro, 1989). Therefore, a tremendous number of methods have been proposed to address the problem (Christophides, et al., 2020), (Mudgal, et al., 2018), (Jaro, 1989), and implemented services are often available (Govind, 2018). Machine-learning-based matchers (Eraheem, et al., 2014), (Li, et al., 2020), (Yao, et al., 2022) are widely used for many other problems in digital libraries (Nielsen, 2018).

Human-in-the-loop entity matching (Osawa, et al., 2021), (Gokhale, et al., 2014), (Das, et al., 2017) is a human-AI collaborative approach to the problem and known as being effective when the data set is incomplete, or the matching requires domain knowledge (Trabelsi, et al., 2022). A typical human-in-the-loop ML matching is to allow an ML matcher to ask humans to match entities when it cannot match them with high confidence.

However, ML-based matchers cannot avoid the *unknown-unknown problem*, i.e., they can match the entities incorrectly with high confidence (Chung, et al., 2019). This is an inherent weakness of the ML-based matchers, because typical human-in-the-loop approaches choose a pair when the matching result is uncertain, or multiple matchers disagree on the result (Settles, 2010), but such procedures do not guarantee that the matching decisions on the remaining pairs are correct.

This paper addresses an inconsistency-driven method that addresses this problem, which can be used with *any* matchers that output matching probabilities for a pair of entities. The method chooses the entity pairs it asks humans to match, by an inconsistency-based "sampling"; when it finds inconsistency in the equivalence relation behind entity matching, it picks up the pair of entities that cause the inconsistency and asks humans to fix the matches. For example, if a matcher returns a negative result only for a particular pair (e.g., s_1 and s_3) among three entities s_1, s_2 and s_3 (Fig 1), the result is inconsistent because it violates the transitive law of the equivalence relationship.

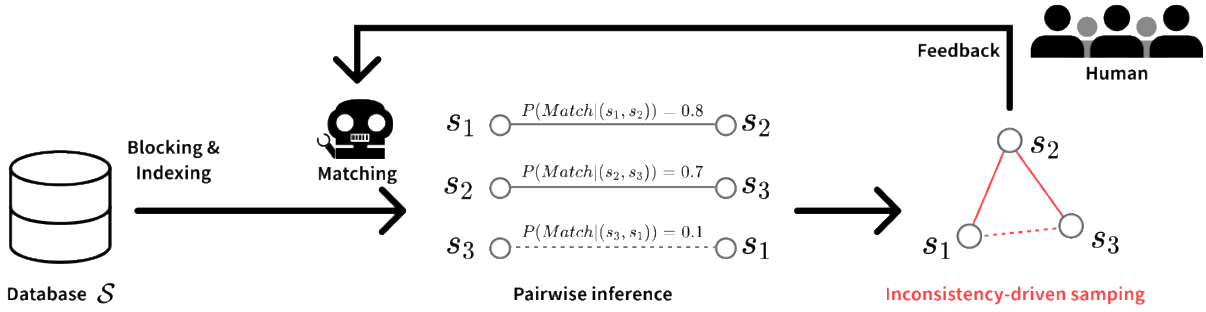

Figure 1. Inconsistency-driven human-in-the-loop entity matching: if the ML matcher outputs the results that are inconsistent with each other, humans correct the result, and the feedback is given to the matcher to improve the results

We implemented the inconsistency-driven sampling in a simple framework combining Bayesian inference and similarity-based blocking method, to highlight the effects of the sampling. In the framework, we assume that we have a certain amount of training data with known labels in advance for the blocking and Bayesian inference. Then, we conducted an extensive set of experiments with different accuracies of matchers, expecting that the result reveals how and when the method is effective. Therefore, we focus on the quality improvement relative to the performance (in terms of f1 values (We use the f1 value instead of accuracy because of the imbalance of the numbers between matched and unmatched pairs)) of the original matcher.

Our research questions are as follows: (RQ1) What is the characteristic of the inconsistency-driven approach as an entity matching strategy? (RQ2) In what situation the approach is effective?

The contributions of this paper are twofold: First, we show a principled framework for an inconsistency-driven approach for human-in-the-loop entity matching. Our approach complements any existing matcher that can cause the unknown-unknown problem. This paper shows an implementation of our approach in a simple framework to address the effects of the approach.

Second, we show the result of an extensive set of experiments with three real-world datasets with different characteristics. We then conducted a detailed analysis of the results. Consequently, we observed the effects of our approach in different situations, revealing how and when the approach is effective.

Note that our research question is not about the performance of a particular matcher. Our findings are summarized as follows.

1. The inconsistency-driven sampling chooses data items that are completely different from those chosen by uncertainty-based and random sampling.
2. The inconsistency-driven sampling has advantages even if human answers are not completely correct as long as the performance of the matcher is relatively high. This is because the inconsistency-based check works well in finding incorrect responses from humans as long as the performance of the matcher is relatively high.
3. A simple hybrid strategy broadens the sweet spot of the inconsistency-driven approach; it lowers the f1 value threshold for which inconsistency-driven sampling is effective.

Related works

In this section, we introduce the studies related to this research and describe their methods and the position of this research.

Rule-based or ML-based entity matching

Many solutions have been proposed for the entity matching problem. The classic example is a rule-based approach. The method clusters those that contain full or partial matches of attribute values or identical tokens that have been segmented into words (Jaro, 1989), (Cohen, et al., 2003), (Benjelloun, et al., 2009). Recently, a machine learning approach has been proposed. They use random forests (Gokhale, et al., 2014), (Das, et al., 2017), and metric learning (Peeters, et al., 2022), (Osawa, et al., 2021) to judge whether an entity pair is a match by calculating its similarity. These methods are simple and inexpensive but are vulnerable to orthographical variants of data.

Human-in-the-loop entity matching

Several studies have pointed out the limitations of entity matching using only computers without human intervention. (Takashi et al. 2019) identified low-precision results based on similarity using Okapi BM25 (Robertson, et al., 1995) and proposed a crowdsourcing-based method that allows for human interaction. (Das, et al., 2017) proposed the framework, Falcon. This method assumes that there is no training data, and a small portion of the target data is labelled by a human to prepare the training data. Other methods using human-in-the-loop have been proposed (Li, 2017), (Gokhale, et al., 2014), (Osawa, et al., 2021), (Eraheem, et al., 2014). Any methods are used to generate or modify data for quality of results and training of the computer, and the other methods are used in much the same way. In this study, we propose a human task from the perspective of obtaining efficient training data.

Methods using transitivity law and domain knowledge

(Zhu et al., 2020) proposed a method using the transitivity of equivalence class. The method tries to find two matched pairs that share the same entity so that it can infer another match between other two entities, because the matching is done by crowdsourcing and requires a huge monetary cost. Our work is different from it in the way to use the transitivity of equivalence because we use it to detect errors in AI decisions assuming AI is the main matcher. Other methods (Li, et al., 2020), (Trabelsi, et al., 2022) use domain knowledge to improve accuracy in entity matching. Their approach is completely different from ours and complementary to each other.

Problem definition

In this section, we define the problem. Table 1 shows definitions of symbols in this paper.

We define a tuple of entities as $x = (s_i, s_j)$, set of all possible tuple of entities as $\mathcal{X} = \{(s_i, s_j) \in \mathcal{S}^2 \mid i < j\}$, a set of labels $\mathcal{Y} = \{Match, Unmatch\}$, and a set of all true data as $\mathcal{U}^n = \{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$, where $n = {}_{|\mathcal{S}|}C_2$. $\mathcal{L}^m = \{(x_i, y_i)\}_{i=1}^m \subseteq \mathcal{U}^n$is the labelled training set with m samples(We assume that humans give answers with the platform such as crowdsourcing) (we give m and $m \ll n$). Our main goal is to design a query strategy $\mathcal{Q}: \mathcal{U}^n \to \mathcal{L}^m$ to train an entity matching model (matcher) $f \in \mathcal{F}$, $f: \mathcal{X} \to \mathcal{Y}$. The optimization problem can be expressed as follows:

$$\underset{\mathcal{L}^m \subseteq \mathcal{U}^n}{\operatorname{argmax}} \frac{1}{n} \sum_{(x,y) \in \mathcal{U}^n} \delta(f(x) = y \vee (x, y) \in \mathcal{L}^m) \tag{1}$$

where δ is the indicator function. The intuition of Eq. (1) is that, for all pairs of entities in a dataset, it is beneficial for the matcher to answer a matching for a pair x correctly, and the dataset $\mathcal{L}^m$ contains the correct label for a pair x. We assume the model returns a probability of $Match$ or $Ummatch$, and we note the probability of $Match$ that the matching model gives to a tuple (s_i, s_j) as $P(Match | s_i, s_j)$.

Symbol	Definition
$\mathcal{S} = \{s_i\}$	All entities in the dataset.
s_i	An entity.
$y \in \mathcal{Y} = \{Match, Unmatch\}$	Labels.
$x \in \mathcal{X} = \{(s_i, s_j) \in \mathcal{S} \times \mathcal{S} \mid i < j\}$	Possible tuples.
$\mathcal{U}^n \subset \mathcal{X} \times \mathcal{Y}$	A set of all possible pairs and their labels.
$\mathcal{L}^m \subset \mathcal{U}^n$	Labeled entity tuples for retraining.
$\mathcal{Q}: \mathcal{U}^n \to \mathcal{L}^m$	A sampling strategy for labeling.
$f \in \mathcal{F}$	Model (e.g., Bayesian inference).

Table 1. Symbol definitions

Inconsistency-driven sampling

Basic idea

The inconsistency-driven sampling samples the pairs of the entities asked to the human. Intuitively, with the current labelled set $\mathcal{L}^{m'}$ we compute $P(Match \mid s_i, s_j)$ for all pairs in $\mathcal{U}^n - \mathcal{L}^{m'}$, find pairs that cause inconsistency, choose m pairs, and re-train the parameters of matching model.

To find the inconsistent pairs from the inference result by the current matching model, we consider pairs in a triple of the entities. We call the matching of the three pairs inconsistent if they constitute impossible patterns under the transitivity of equivalence relation (such as "positive," "positive," and "negative," as shown in Fig. 2).

Using Bayesian inference, we can calculate the probability that each pair is a match. Given a triple $\Delta_{ijk} = (s_i, s_j, s_k)$, the probability that these three pairs cause inconsistency can be calculated as in Eq. (2).

$$Inconsistency(\Delta_{ijk}) = \sum_{(a,b,c)\in\{(i,j,k),(j,k,i),(k,i,j)\}} P(Match|s_a, s_b)P(Match|s_b, s_c)P(Unmatch|s_c, s_a) \quad (2)$$

In this study, when the *Inconsistency* is higher, we assume the matcher is getting more confused, and we prioritize the triple to ask humans to correct it.

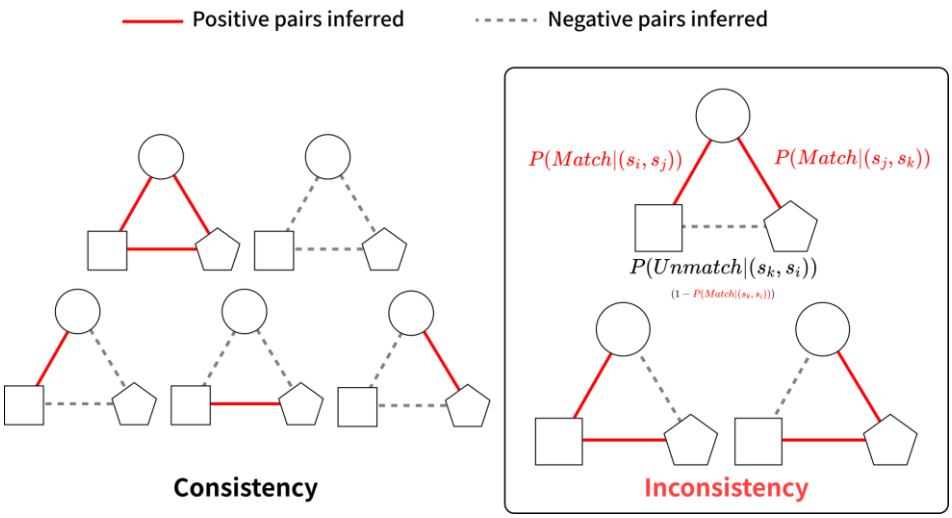

Figure 2. Inconsistency in focusing on the three data

Inconsistency-driven sampling algorithm

Algorithm 1 implements the idea of inconsistency-driven sampling. It takes a set of entity pairs with match probabilities $\hat{\mathcal{D}} = \{((s_i, s_j), p)\} \subset \mathcal{X} \times [0,1]$ and outputs m sample pairs taken from the given set. Note that the matching probability $p = P(Match|s_i, s_j)$. Note that the given set of entity pairs forms a graph, and that the triples that may cause the inconsistency in the graph are triangles. We define a set of triangles as

$$\mathcal{T} = \{\Delta_{ijk} \in \mathcal{S}^3 \mid ((s_i, s_j), p_1), ((s_j, s_k), p_2), ((s_k, s_i), p_3) \in \hat{\mathcal{D}}\} \quad (3)$$

The flow of inconsistency-driven sampling is as follows: First, we apply the blocking method for the set of entities, then sort the triangles in descending order of *Inconsistency*, and lastly, sample the $m/3$ triangles with the highest *Inconsistency* to sample m pairs of entities.

Algorithm 1 Inconsistency-driven sampling

Input: $\hat{\mathcal{D}}, m$
Output: A set of m pairs of entities
1: $\mathcal{T}' \leftarrow$ Sort $\Delta_{ijk} \in \mathcal{T}$ in descending order by $Inconsistency(\Delta_{ijk})$
2: **return** Triangles up to the $(m/3)$-th of $\mathcal{T}'$

Algorithm 1. Inconsistency-driven sampling

Experiments

We conducted an extensive set of experiments to address our research questions and evaluated the impact of sampling strategies on the f1 value of matchers on datasets from different domains.

Figure 3 overviews the overall experiment workflow. First, we apply a blocking technique to each of the three datasets for the experiment and generate training and evaluation datasets. Then, we execute human-in-the-loop entity matching iterations with each of the five sampling strategies. We explain the steps one by one.

Datasets and human settings

Datasets

We used three datasets taken from different domains (Table 2): **Persons** (Köpcke, et al., 2010), **Bibliorecords** (a private dataset supplied by public libraries in Japan), and **Music** (Köpcke, et al., 2010). Each dataset has a set of entities that has an attribute that stores cluster labels; if two have the same cluster label, they are matched entities, i.e., they represent the same entity. **Persons** and **Bibliorecords** are relatively clean datasets, while **Music** has many missing values and contains dirty attributes, such as having an album name in the (music) title attribute.

Domain (Language)	Attributes
Persons (Köpcke, et al., 2010) (English)	name, surname, suburb, postcode
Bibliorecords (Japanese)	title, author, publisher, date
Music (Köpcke, et al., 2010) (English)	artist, title, album, year, length

Table 2. Datasets from three different domains

Human settings

It is unavoidable for humans to make mistakes. In order to see the impact of errors on each sampling strategy in a systematic way, we took a simulation-based approach by implementing agents that serve as humans who give labels with a given accuracy; we examined three cases in terms of the accuracy of human labels: 100%, 95%, and 90%. The accuracy is, in other words, the percentage of noise labels in active learning, and we investigated the impact of noise labels on active learning (Wu, et al., 2022), (Younesian, et al., 2021).

Blocking and data preparation

For each of the three original datasets, we generated two data sets we use in the blocking phase and the human-in-the-loop entity matching iterations:

- the training dataset for the blocking phase and the Bayesian inference, and
- the evaluation dataset for evaluating sampling strategies to update the Bayesian inference.

The two datasets were constructed as follows and disjointed from each other.

Training dataset D^t: For each original dataset, we randomly choose a set of 15,000 positive pairs (the two cluster labels are the same) and 15,000 negative pairs (the labels are different from each other) from all pairs of entities in the original dataset. Then, every entity pair and their label form a triple contained in D^t.

Evaluation Dataset D^e: First, we randomly select clusters from each of the original datasets until each dataset contains about 2000 entities in total. Then, we generate a set of entity pairs with labels from the clusters.

Then, we constructed the dataset by applying a standard blocking technique based on metric learning to all pairs of the selected entities. The blocking technique chose only pairs of entities

that were closer than the threshold after metric learning on the distance among entities with the training dataset.

Table 3 shows the statistics of the evaluation datasets constructed this way. Note that the blocking does not work in favour of the inconsistency-driven sampling because the blocking is done based on the metric learning result, and it removed some of the potentially matched pairs that can affect the inconsistency-based sampling (i.e., the recall of the matches is less than 1).

Domain (Language)	#Entities	#Pairs	#Matches	Recall of matches of the blocking
Persons	2001	48,835	1379	0.848
Bibliorecords	2001	66,911	1594	0.840
Music	2002	196,414	1613	0.739

Table 3. Statistics of the evaluation datasets after blocking

Human-in-the-loop entity matching iterations

Algorithm 2 gives the concrete steps in the human-in-the-loop entity matching iterations in the experiment workflow. First, we conduct the Bayesian inference with $D^{\{t\}}$ for $D^{\{e\}}$. Then, in the iteration, it chooses samples for which it obtains human labels and uses the obtained result to update the inference. We chose Bayesian inference because it is a simple matcher that satisfies the requirement for the application of our inconsistency-driven sampling: it outputs a matching probability for a pair of entities. Note that our research question is not the performance of a particular matcher, and we can use this Bayesian inference matcher without loss of generality.

Algorithm 2 Human-in-the-loop entity matching iterations for the experiment

Input: Training Data $\mathcal{D}^t \subset \mathcal{U}^n$, Evaluation Data $\mathcal{D}^e \subset \mathcal{U}^n - \mathcal{D}^t$
Output: (None)
1: **for** $k = 1$ to 10 **do**
2: Conduct the Bayesian inference with $\mathcal{D}^t$ for $\mathcal{D}^e$ to output $\hat{\mathcal{D}}^e$
3: Calculate F1 value for $\hat{\mathcal{D}}^e$ (with $\mathcal{D}^e$)
4: Choose m pairs from $\hat{\mathcal{D}}^e$ using **Algorithm 1**
5: Obtain the human label for m pairs and add the data to $\mathcal{D}^t$
6: Update the matcher with m pairs
7: **end for**

Algorithm 2. Human-in-the-loop entity matching iterations for the experiment

Matcher implementation

The workflow implements a Bayesian inference-based matcher. The model infers the matching by considering the similarity measures between the given two entities. Our model requires the four similarities, which are the basic similarity measures for texts shown in Table 4.

l	Indicator
1	FastText vector (Bojanowski, et al., 2017)
2	Jaro-winkler (Winkler, et al., 1990)
3	Levenshtein (Levenshtein, et al., 1966)
4	Gestalt Pattern Matching (Virtanen, et al., 2020)

Table 4. Statistics of the evaluation datasets after blocking

The probabilities obtained from these probability density functions can be integrated as in Eq. (4) to estimate the matching probability of the pair.

$$P(\,Match \mid s_i, s_j\,) = \frac{\prod_{l=1}^{n} p(\,z_{i,j}^{(l)} \mid Match\,)P(Match)}{\sum_{Class\in\{Match,Unmatch\}} \prod_{l=1}^{n} p(\,z_{i,j}^{(l)} \mid Class\,)P(Class)} \tag{4}$$

Note that $z_{i,j} = \left(z_{i,j}^{(1)}, \cdots, z_{i,j}^{(n)}\right)^{\top}$ is a vector based on the similarity between entities s_i and s_j. The parameters for constructing the probability p are obtained by fitting a probability density function based on the mixed Gaussian distribution (Dempster, et al., 1977), (Pedregosa, et al., 2011) to the training data (Fig. 4).

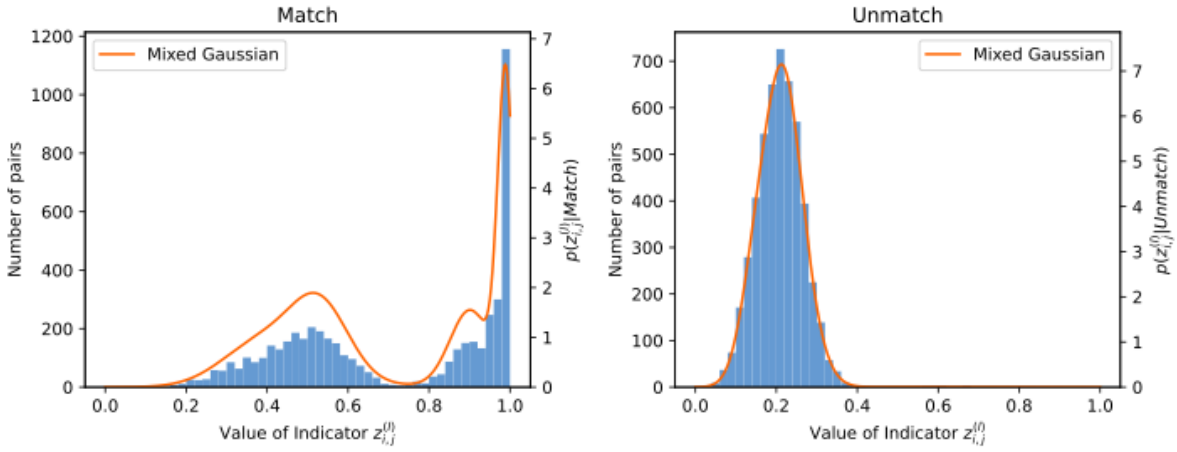

Figure 4. Fitting probability density function

Sampling strategies

We used the following five sampling strategies. Note that uncertainty sampling and query-by-committee sampling are *model-based* strategies while diversity sampling and random sampling are *model-free* strategies.

1. **Inconsistency-driven sampling**. This is the method we proposed in which pairs whose estimated labels lead to an inconsistency.

2. **Uncertainty sampling (model-based)** (Settles, 2010). In Bayesian inference, we compute the confidence value for each pair and chose those pairs that are not clearly positive or negative. Specifically, we use Eq. (5) for the confidence value and choose the pair with the least confidence value.

$$Priority = 1 - \left|P\left(Match|s_i, s_j\right) - 0.5\right| \tag{5}$$

3. **Query-by-committee sampling (model-based)** (Settles, 2010). Query-by-committee is a method of choosing pairs by aggregating the results of multiple indicators. In this experiment, we used two indicators that computed positive and negative scores for each indicator, which were calculated and sampled from antagonistic pairs. Specifically, sample in order of increasing value of Eq. (8).

$$Positive = \prod_{l=1}^{n} p\left(z_{i,j}^{(l)}|Match\right)P(Match) \tag{6}$$

$$Negative = \prod_{l=1}^{n} p\left(z_{i,j}^{(l)}|Unmatch\right)P(Unmatch) \tag{7}$$

$$Priority = 1 - (Positive + Negative) \tag{8}$$

4. **Diversity sampling (model-free)** (O'Neill, et al., 2017). This method selects distant pairs from those that have already been labelled by humans. The distance is calculated using the embedded representation of FastText (Bojanowski, et al., 2017). The first iteration is $\mathcal{L}^m = \emptyset$, so random sampling is executed only in the iteration.

5. **Random sampling (model-free)**. This method randomly selects the candidate pairs. We used the random function from Python's random module.

Other settings

Prior distribution for Bayesian inference. The prior distribution was set to $P(Match) = 0.1$ and $P(Unmatch) = 0.9$.

Batch sampling. We adopted a batch sampling scheme to reduce the number of inference updates; in each iteration, we choose m samples (instead of choosing one sample) and obtain their human labels before each inference update. We set $m = 300$. We need to be careful when dealing with inconsistency-driven sampling in the batch because we may obtain more than one human label for the same pair if it appears in different sets of inconsistent triangles. We solved the duplication by majority vote.

Languages and libraries. These algorithms were implemented by Python3, using the modules Tensorflow (Abadi et al., 2015) for metric learner construction, Cupy (Okuta et al., 2017) and Faiss (Johnson et al., 2019) for blocking and indexing, and Scipy (Virtanen, et al., 2020) and Scikit-Learn (Pedregosa, et al., 2011) for Bayesian inference probability density function manipulation.

Results

Difference of sampling distributions

Fig. 5 shows how different the five sampling strategies are from each other in terms of chosen samples. The horizontal axis represents the match probability determined by the inference, where the closer the match probability is to 1, the more likely the pair is to be matched, while the closer it is to 0, the more unlikely. The vertical axis represents the number of chosen pairs in the sampling strategy (in the log scale). The higher orange intensity means that they are chosen in earlier iterations.

The result clearly shows that the strategies are remarkably different in terms of chosen samples. Inconsistency-driven sampling tends to choose pairs that are highly likely or unlikely to be positive, while uncertainty sampling constantly chooses pairs in the middle. Query-by-committee, diversity, and random sampling choose pairs from a wide range.

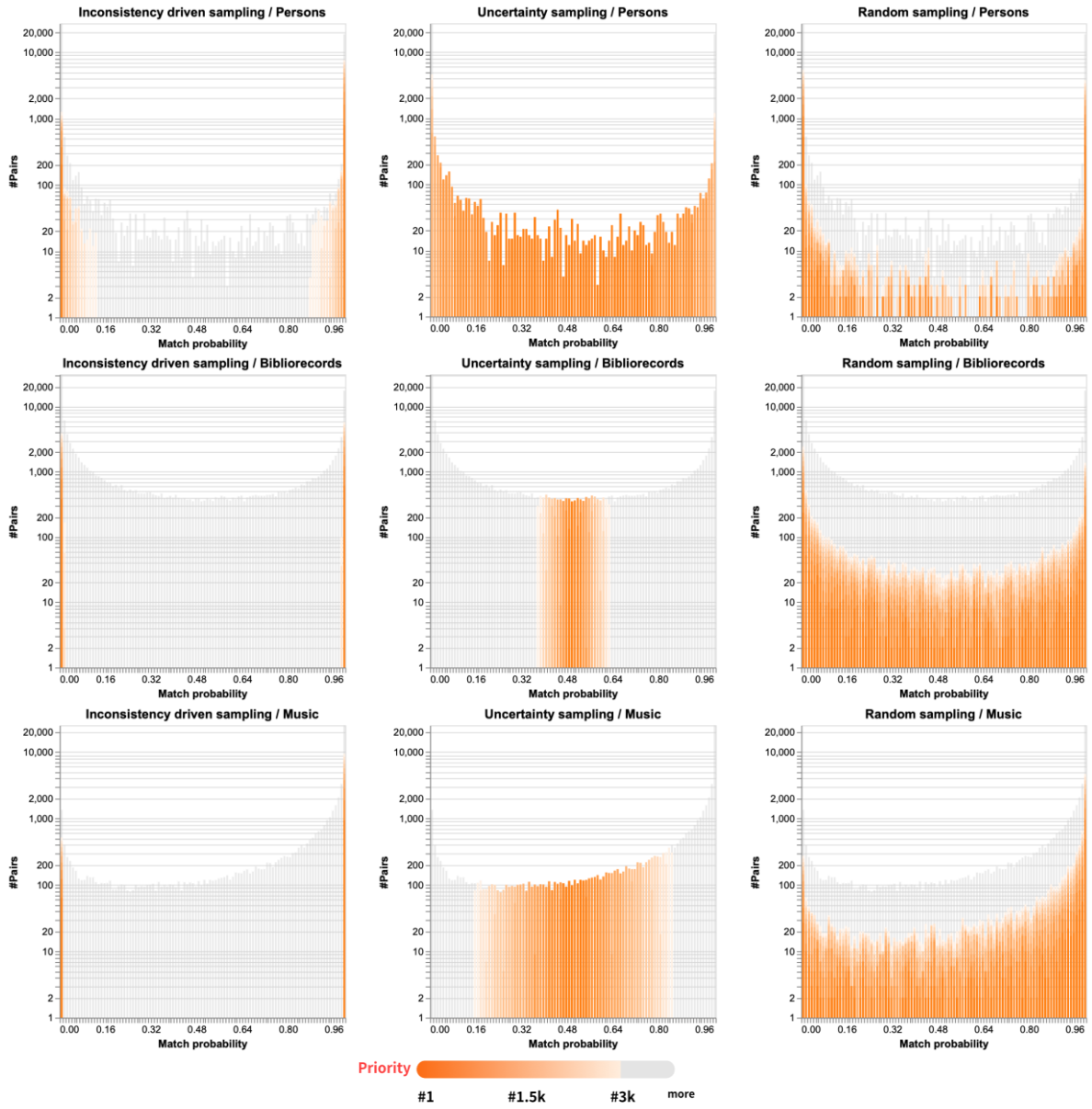

Figure 5. Analysis: Distribution of match probability for each sampling strategy

Effects on matcher performance

Fig. 6 shows the effects of each sampling strategy on the matcher performance in the iterations. The X-axis is the number of iterations. The Y-axis is the F1 value for the output of Bayesian Inference and human labels after each iteration for the evaluation dataset. Each line represents a sampling strategy.

The aim of seeing the figure is to identify the sweet spots of each sampling strategy. Note that the performance of matchers (to be used with sampling strategies) for **Persons**, **Bibliorecords**, and **Music** are very different (very high, moderate, and very low, respectively). The result suggests the following. First, inconsistency-driven sampling is effective when the f1 value of the matcher is high (i.e., higher than 0.8), while uncertainty sampling performs the best, especially for lower-quality matchers. Second, when the f1 value is high, the performance inconsistency-driven is stable, while other sampling strategies are directly influenced when the accuracy of human labels becomes lower. The results are reasonable for the following reasons: if the f1 value of matchers is not high,

it is difficult to identify inconsistency correctly. Third, the inconsistency-driven approach can identify inconsistencies even for newly obtained human labels, while other strategies take human inputs as oracles, even if they are incorrect.

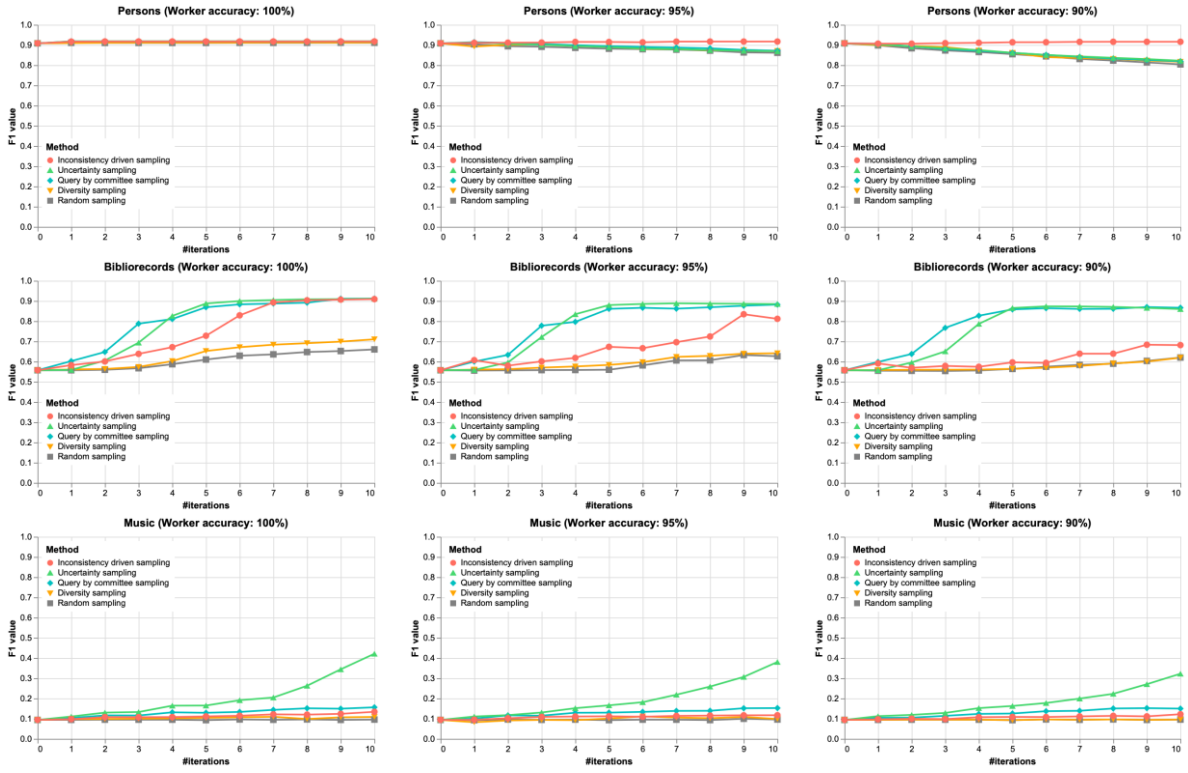

Figure 6. Experiment: Comparison of f1 value for different sampling strategies

Hybrid strategy

The results so far showed that (1) the inconsistency-based sampling works better when the inference f1 value is high while uncertainty sampling works better otherwise and that (2) the chosen pairs of the strategies are completely different to each other. They suggest that it is worth considering a hybrid strategy. Since it is difficult to set the threshold to switch strategies, we considered a simple hybrid strategy that switches uncertainty and inconsistency-driven strategies in each iteration.

Fig. 7 shows the results of applying the hybrid strategy to the three datasets with 90%-accurate human labels. The result shows that the hybrid strategy performs well when the inference f1 value is moderate or higher, which covers many practical situations. On the other hand, uncertainty sampling still works better with the inference with an extremely low f1 value.

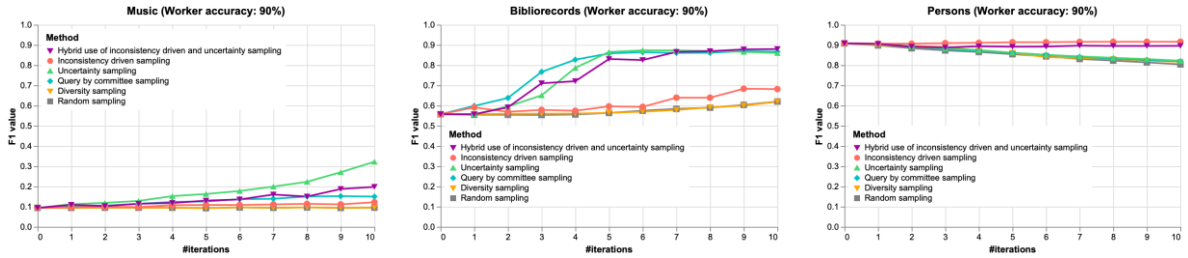

Figure 7. Experiment: comparison of f1 value for different sampling strategies

Conclusion

This paper addressed an inconsistency-based sampling strategy to deal with the *unknown-unknown problem* in active learning for entity matching. The method asks humans to resolve the entities when we find inconsistency in the transitivity property. This paper implemented a human-in-the-loop entity matching framework with this sampling strategy with similarity-based blocking method and Bayesian inference. It also explained the result of an extensive set of experiments that reveals how and when the method is effective. The result showed that the inconsistency-based sampling selects very different entity pairs compared to other sampling strategies and that a simple hybrid strategy performs well in many practical situations. Future work includes the interaction of the sampling strategy and matcher implementations. For example, the sampling result may suggest switching to other types of matchers.

Acknowledgements

This work was supported in part by JST CREST(JPMJCR22M2), Grants-in-Aid for Scientific Research (22H00508, 21H03552).

About the authors

Hiroyoshi Ito is an Assistant Professor at Institute of Library, Information and Media Science, University of Tsukuba. He can be contacted at: ito@slis.tsukuba.ac.jp

Takahiro Koizumi is a master's student at Graduate School of Comprehensive Human Sciences, University of Tsukuba. He can be contacted at: takahiro.koizumi.2022b@gmail.com

Ryuji Yoshimoto is an Engineer at CARLIL Inc. He can be contacted at: ryuuji@calil.jp

Yukihiro Fukushima is an Associate Professor at Faculty of Letters, Keio University. He can be contacted at: fukusima-y@keio.jp

Takashi Harada is a Professor at Center for License and Qualification, Doshisha University. He can be contacted at: ushi@slis.doshisha.ac.jp

Atsuyuki Morishima is a Professor at Institute of Library, Information and Media Science, University of Tsukuba. He can be contacted at: mori@slis.tsukuba.ac.jp

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), http://tensorflow.org/, software available from tensorflow.org

Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S.E., Widom, J. (2009). Swoosh: a generic approach to entity resolution. The VLDB Journal 18, 255–276.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2017). Enriching word vectors with subword information

Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., Stefanidis, K. (2020). An overview of end-to-end entity resolution for big data. ACM Comput. Surv. 53(6). https://doi.org/10.1145/3418896

Chung, Y., Haas, P.J., Upfal, E., Kraska, T. (2019). Unknown examples & machine learning model generalization.

Cohen, W.W., Ravikumar, P., Fienberg, S.E., et al. (2003). A comparison of string dis- tance metrics for name-matching tasks. In: IIWeb. vol. 3, pp. 73–78.

Das, S., G.C., P.S., Doan, A., Naughton, J.F., Krishnan, G., Deep, R., Arcaute, E., Raghavendra, V., Park, Y. (2017). Falcon: Scaling up hands-off crowdsourced entity matching to build cloud services. In: Proceedings of the 2017 ACM International Conference on Management of Data. pp. 1431–1446. https://doi.org/10.1145/ 3035918.3035960

Dempster, A.P., Laird, N.M., Rubin, D.B. (1977). Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society: series B (methodological) 39(1), 1–22.

Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., Tang, N. (2018). Distributed representations of tuples for entity resolution. Proc. VLDB Endow. 11(11), 1454–1467. https://doi.org/10.14778/3236187.3236198

Gokhale, C., Das, S., Doan, A., Naughton, J.F., Rampalli, N., Shavlik, J., Zhu, X. (2014). Corleone: Hands-off crowdsourcing for entity matching. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. pp. 601–612. https://doi.org/10.1145/2588555.2588576

Govind, Y., Paulson, E., Nagarajan, P., C., P.S.G., Doan, A., Park, Y., Fung, G.M., Conathan, D., Carter, M., Sun, M. (2018). Cloudmatcher: A hands-off cloud/crowd service for entity matching. Proc. VLDB Endow. 11(12), 2042–2045. https://doi.org/10.14778/3229863.3236255

Jaro, M.A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. Journal of the American Statistical Association 84(406), 414. https://doi.org/10.2307/2289924

Johnson, J., Douze, M., Jégou, H. (2019). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data 7(3), 535–547.

Köpcke, H., Thor, A., Rahm, E. (2010). Evaluation of entity resolution ap- proaches on real-world match problems. Proc. VLDB Endow. 3(1-2), 484–493. https://doi.org/10.14778/1920841.1920904

Levenshtein, V.I., et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710.

Li, G. (2017). Human-in-the-loop data integration. VLDB Endowment 10(12), 2006–201.

Li, Y., Li, J., Suhara, Y., Doan, A., Tan, W.C. (2020). Deep entity matching with pre-trained language models. Proceedings of the VLDB Endowment 14(1), 50–60. https://doi.org/10.14778/3421424.3421431

Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., Raghavendra, V. (2018). Deep learning for entity matching: A design space exploration. In: Proceedings of the 2018 International Conference on Management of Data. pp. 19–34. https://doi.org/10.1145/3183713.3196926

Nielsen, R.D. (2018). Introduction to machine learning for digital library applications. In: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries. p. 421–422. https://doi.org/10.1145/3197026.3201780

Okuta, R., Unno, Y., Nishino, D., Hido, S., Loomis, C. (2017). Cupy:Anumpy-compatible library for nvidia gpu calculations. In: Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural In- formation Processing Systems (NIPS), http://learningsys.org/nips17/assets/papers/paper_16.pdf

O'Neill, J., Delany, S., MacNamee, B. (2017). Model-Free and Model-Based Active Learning for Regression, vol. 513, pp. 375–386. https://doi.org/10.1007/ 978-3-319-46562-3_24

Osawa, N., Ito, H., Fukushima, Y., Harada, T., Morishima, A. (2021). Bubble: A quality-aware human-in-the-loop entity matching framework. In: The 5th IEEE Workshop on Human-in-the-loop Methods and Future of Work in Big-Data (IEEE HMData2021). pp. 3557–3565. https://doi.org/ 10.1109/BigData52589.2021.9672002

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Peeters, R., Bizer, C. (2022). Supervised contrastive learning for product matching. In: Companion Proceedings of the Web Conference 2022. pp. 248–251 https://doi.org/10.1145/3487553.3524254

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M. (1995). Okapi at trec-3, https://www.microsoft.com/en-us/research/ publication/okapi-at-trec-3/

Settles, B. (2010). Active learning literature survey. In: Active Learning Literature Survey. University of Wisconsin-Madison, https://minds.wisconsin.edu/bitstream/handle/1793/60660/TR1648.pdf

Takashi, H., Yukihiro, F., Sho, S., Misato, T., Ryuji, Y., Atsuyuki, M. (2019). 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.

Trabelsi, M., Heflin, J., Cao, J. (2022). Dame: Domain adaptation for matching entities.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Courna- peau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., Vander- Plas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P. (2020). SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Winkler, W. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. Proceedings of the Section on Survey Research Methods pp. 354–359.

Wu, M., Li, C., Yao, Z. (2022). Deep active learning for computer vision tasks: Methodologies, applications, and challenges. Applied Sciences 12(16) https://doi.org/10.3390/app12168103

Xu, X., Zhang, F., Niu, Z. (2008). An ontology-based query system for digital libraries. IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application. vol. 1, pp. 222–226. https://doi.org/10.1109/PACIIA.2008. 360

Yao, D., Gu, Y., Cong, G., Jin, H., Lv, X. (2022). Entity resolution with hierarchical graph attention networks. In: Proceedings of the 2022 International Conference on Man- agement of Data. pp. 429–442. https://doi.org/10.1145/3514221.3517872

Younesian, T., Zhao, Z., Ghiassi, A., Birke, R., Chen, L.Y. (2021). Qactor: Active learning on noisy labels. In: Balasubramanian, V.N., Tsang, I. (eds.) Proceedings of The 13th Asian Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 157, pp. 548–563., https://proceedings. mlr.press/v157/younesian21a.html

Zhu, Y., Liu, H., Wu, Z., Du, Y. (2020). Relation-aware neighborhood matching model for entity alignment. https://arxiv.org/abs/2012.08128