Active Blocking Scheme Learning for Entity Resolution

Jingyu Shao and Qing Wang [⋆]

Research School of Computer Science, Australian National University
{jingyu.shao,qing.wang}@anu.edu.au

Abstract. Blocking is an important part of entity resolution. It aims to improve time efficiency by grouping potentially matched records into the same block. In the past, both supervised and unsupervised approaches have been proposed. Nonetheless, existing approaches have some limitations: either a large amount of labels are required or blocking quality is hard to be guaranteed. To address these issues, we propose a blocking scheme learning approach based on active learning techniques. With a limited label budget, our approach can learn a blocking scheme to generate high quality blocks. Two strategies called active sampling and active branching are proposed to select samples and generate blocking schemes efficiently. We experimentally verify that our approach outperforms several baseline approaches over four real-world datasets.

Keywords: Entity Resolution, Blocking Scheme, Active Learning

1 Introduction

Entity Resolution (ER), which is also called Record Linkage [11,12], Deduplication [6] or Data Matching [5], refers to the process of identifying records which represent the same real-world entity from one or more datasets [17]. Blocking techniques are commonly applied to improve time efficiency in the ER process by grouping potentially matched records into the same block [16]. It can thus reduce the number of record pairs to be compared. For example, given a dataset D, the total number of record pairs to be compared is $\frac{|D|*(|D|-1)}{2}$ (i.e. each record should be compared with all the others in D). With blocking, the number of record pairs to be compared can be reduced to no more than $\frac{m*(m-1)}{2} * |B|$, where m is the number of records in the largest block and $|B|$ is the number of blocks, since the comparison only occurs between records in the same block.

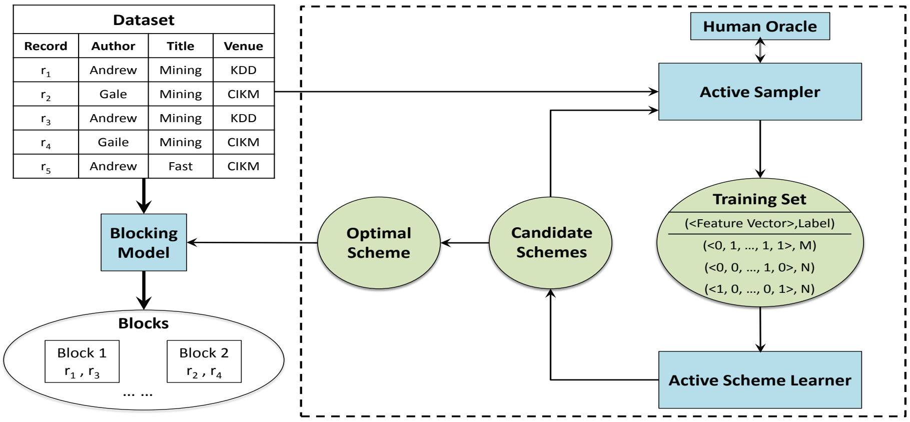
In recent years, a number of blocking approaches have been proposed to learn blocking schemes [3, 13, 15]. They generally fall into two categories: (1) supervised blocking scheme learning approaches. For example, Michelson and Knoblock presented an algorithm called *BSL* to automatically learn effective blocking schemes [15]; (2) Unsupervised blocking scheme learning approaches

[⋆] This work was partially funded by the Australian Research Council (ARC) under Discovery Project DP160101934.

[13]. For example, Kejriwal and Miranker proposed an algorithm called *Fisher* which uses record similarity to generate labels for training based on the TF-IDF measure, and a blocking scheme can then be learned from a training set [13].

However, these existing approaches on blocking scheme learning still have some limitations: (1) It is expensive to obtain ground-truth labels in real-life applications. Particularly, match and non-match labels in entity resolution are often highly imbalanced [16], which is called the *class imbalance problem*. Existing supervised learning approaches use random sampling to generate blocking schemes, which can only guarantee the blocking quality when sufficient training samples are available [15]. (2) Blocking quality is hard to be guaranteed in unsupervised approaches. These approaches obtain the labels of record pairs based on the assumption that the more similar two records are, the more likely they can be a match. However, this assumption does not always hold [17]. As a result, the labels may not be reliable and no blocking quality can be promised. A question arising is: Can we learn a blocking scheme with blocking quality guaranteed and the cost of labels reduced?

To answer this question, we propose an active blocking scheme learning approach which incorporates active learning techniques [7, 10] into the blocking scheme learning process. In our approach, we actively learn the blocking scheme based on a set of blocking predicates using a balanced active sampling strategy, which aims to solve the class imbalance problem of entity resolution. The experimental results show that our proposed approach yields high quality blocks within a specified error bound and a limited budget of labels.



Dataset

Record	Author	Title	Venue
r_1	Andrew	Mining	KDD
r_2	Gale	Mining	CIKM
r_3	Andrew	Mining	KDD
r_4	Gaile	Mining	CIKM
r_5	Andrew	Fast	CIKM

Fig. 1. Overview of the active blocking scheme learning approach

Figure 1 illustrates our proposed approach, which works as follows: Given a dataset D, an active sampler selects samples from D based on a set of candidate schemes, and asks a human oracle for labels. Then an active scheme learner generates a set of refined candidate schemes, enabling the active sampler to adaptively select more samples. Within a limited label budget and an error bound, the optimal scheme will be selected among the candidate schemes.

The contributions of this paper are as follows. (1) We propose a blocking scheme learning approach based on active learning, which can efficiently learn a blocking scheme with less samples while still achieving high quality. (2) We develop two complementary and integrated active learning strategies for the proposed approach: (a) *Active sampling strategy* which overcomes the class imbalance problem by selecting informative training samples; (b) *Active branching strategy* which determines whether a further conjunction/ disjunction form of candidate schemes should be generated. (3) We have evaluated our approach over four real-world datasets. Our experimental results show that our approach outperforms state-of-the-art approaches.

2 Related Work

Blocking for entity resolution was first mentioned by Fellegi and Sunter in 1969 [9]. Later, Michelson and Knoblock proposed a blocking scheme learning algorithm called *Blocking Scheme Learner (BSL)* [15], which is the first algorithm to learn blocking schemes, instead of manually selecting them by domain experts. In the same year, Bilenko et al. [3] proposed two blocking scheme learning algorithms called *ApproxRBSetCover* and *ApproxDNF* to learn disjunctive blocking schemes and DNF (i.e. Disjunctive Normal Form) blocking schemes, respectively. Kejriwal et al. [13] proposed an unsupervised algorithm for learning blocking schemes. In their work, a weak training set was applied, where both positive and negative labels were generated by calculating the similarity of two records using TF-IDF. A set of blocking predicates was ranked in terms of their fisher scores based on the training set. The predicate with the highest score is selected, and if the other lower ranking predicates can cover more positive pairs in the training set, they will be selected in a disjunctive form. After traversing all the predicates, a blocking scheme is learned.

Active learning techniques have been extensively studied in the past years. Ertekin et al. [8] proved that active learning provided the same or even better results in solving the class imbalance problem, compared with random sampling approaches such as oversampling the minority class and/or undersampling the majority class [4]. For entity resolution, several active learning approaches have also been studied [1, 2, 10]. For example, Arasu et al. [1] proposed an active learning algorithm based on the monotonicity assumption, i.e. the more textually similar a pair of records is, the more likely it is a matched pair. Their algorithm aimed to maximize recall under a specific precision constraint.

Different from the previous approaches, our approach uses active learning techniques to select balanced samples adaptively for tackling the class imbalance problem. This enables us to learn blocking schemes within a limited label budget. We also develop a general strategy to generate blocking schemes that can be conjunctions or disjunctions of an arbitrary number of blocking predicates, instead of limiting at most k predicates to be used in conjunctions [3, 13].

3 Problem Definition

Let D be a dataset consisting of a set of records, and each record $r_i \in D$ be associated with a set of attributes $A = \{a_1, a_2, ..., a_{|A|}\}$. We use $r_i.a_k$ to refer to the value of attribute a_k in a record r_i. Each attribute $a_k \in A$ is associated with a domain $Dom(a_k)$. A *blocking function* $h_{a_k} : Dom(a_k) \rightarrow U$ takes an attribute value $r_i.a_k$ from $Dom(a_k)$ as input and returns a value in U as output. A *blocking predicate* $\langle a_k, h_{a_k} \rangle$ is a pair of attribute a_k and blocking function h_{a_k}. Given a record pair r_i and r_j, a blocking predicate $\langle a_k, h_{a_k} \rangle$ returns *true* if $h_{a_k}(r_i.a_k) = h_{a_k}(r_j.a_k)$ holds, otherwise *false*. For example, we may have *soundex* as a blocking function for attribute *author*, and accordingly, a blocking predicate ⟨*author, soundex*⟩. For two records with values "Gale" and "Gaile", ⟨*author, soundex*⟩ returns *true* because of $soundex(Gale) = soundex(Gaile) = G4$. A *blocking scheme* is a disjunction of conjunctions of blocking predicates (i.e. in the Disjunctive Normal Form).

A *training set* $T = (X, Y)$ consists of a set of feature vectors $X = \{x_1, x_2, ..., x_{|T|}\}$ and a set of labels $Y = \{y_1, y_2, ..., y_{|T|}\}$, where each $y_i \in \{M, N\}$ is the label of x_i $(i = 1, \ldots, |T|)$. Given a record pair r_i, r_j, and a set of blocking predicates P, a *feature vector* of r_i and r_j is a tuple $\langle v_1, v_2, ..., v_{|P|} \rangle$, where each v_k $(k = 1, \ldots, |P|)$ is an equality value of either 1 or 0, describing whether the corresponding blocking predicate $\langle a_k, h_{a_k} \rangle$ returns true or false. Given a pair of records, a *human oracle* ζ is used to provide the label $y_i \in Y$. If $y_i = M$, it indicates that the given record pair refers to the same entity (i.e. *a match*), and analogously, $y_i = N$ indicates that the given record pair refers to two different entities (i.e. *a non-match*). The human oracle ζ is associated with a budget limit $budget(\zeta) \geq 0$, which indicates the total number of labels ζ can provide.

Given a blocking scheme s, a *blocking model* can generate a set of pairwise disjoint blocks $B_s = \{b_1, \ldots, b_{|B_s|}\}$, where $b_k \subseteq D$ $(k = 1, \ldots, |B_s|)$, $\bigcup_{1 \leq k \leq |B_s|} b_k = D$ and $\bigwedge_{1 \leq i \neq j \leq |B_s|} b_i \cap b_j = \emptyset$. Moreover, for any two records r_i and r_j in a block $b_k \in B_s$, s must contain a conjunction of block predicates such that $h(r_i.a_k) = h(r_j.a_k)$ holds for each block predicate $\langle a_k, h \rangle$ in this conjunction. For convenience, we use $tp(B_s)$, $fp(B_s)$ and $fn(B_s)$ to refer to *true positives*, *false positives* and *false negatives* in terms of B_s, respectively. Ideally, a good blocking scheme should yield blocks that minimize the number of record pairs to be compared, while preserving true matches at a required level. We thus define the active blocking problem as follows.

Definition 1. *Given a human oracle ζ, and an error rate $\epsilon \in [0, 1]$, the* ***active blocking problem*** *is to learn a blocking scheme s in terms of the following objective function, through actively selecting a training set T:*

$$\begin{aligned} &\textbf{\textit{minimize}} \quad |fp(B_s)| \\ &\textbf{\textit{subject to}} \quad \frac{|fn(B_s)|}{|tp(B_s)|} \leq \epsilon, \ \textit{and} \ \ |T| \leq budget(\zeta) \end{aligned} \tag{1}$$

4 Active Scheme Learning Framework

In our active scheme learning framework, we develop two complementary and integrated strategies (i.e. *active sampling* and *active branching*) to adaptively generate a set of blocking schemes and learn the optimal one based on actively selected samples. The algorithm we propose is called *Active Scheme Learning (ASL)* and described in Section 4.3.

4.1 Active Sampling

To deal with the active blocking problem, we need both match and non-match samples for training. However, one of the well-known challenges in entity resolution is the class imbalance problem [18]. That is, if samples are selected randomly, there are usually much more non-matches than matches. To tackle this problem, we have observed, as well as shown in the previous work [1, 2], that the more similar two records are, the higher probability they can be a match. This observation suggests that, a balanced representation of similar records and dissimilar records is likely to represent a training set with balanced matches and non-matches. Hence, we define the notion of balance rate.

Definition 2. *Let s be a blocking scheme and X a non-empty feature vector set, the* ***balance rate*** *of X in terms of s, denoted as $\gamma(s, X)$, is defined as:*

$$\gamma(s, X) = \frac{|\{x_i \in X | s(x_i) = true\}| - |\{x_i \in X | s(x_i) = false\}|}{|X|} \quad (2)$$

Conceptually, the balance rate describes how balance or imbalance of the samples in X by comparing the number of similar samples to that of dissimilar samples in terms of a given blocking scheme s. The range of balance rate is $[-1, 1]$. If $\gamma(s, X) = 1$, there are all similar samples in T with regard to s, whereas $\gamma(s, X) = -1$ means all the samples are dissimilar samples. In these two cases, X is highly imbalanced. If $\gamma(s, X) = 0$, there is an equal number of similar and dissimilar samples, indicating that X is balanced.

Based on the notion of balance rate, we convert the class imbalance problem into the balanced sampling problem as follows:

Definition 3. *Given a set of blocking scheme S and a label budget $n \leq budget(\zeta)$, the* ***balanced sampling problem*** *is to select a training set $T = (X, Y)$, where $|X| = n$, in order to:*

$$\textbf{\textit{minimize}} \sum_{s_i \in S} \gamma(s_i, X)^2 \quad (3)$$

For two different blocking schemes s_1 and s_2, they may have different balance rates over the same feature vector set X, i.e. $\gamma(s_1, X) \neq \gamma(s_2, X)$ is possible. The objective here is to find a training set that minimizes the balance rates in terms of the given set of blocking schemes. The optimal case is $\gamma(s_i, X) = 0$, $\forall s_i \in S$. However, this is not always possible to achieve in real world applications.

4.2 Active Branching

Given n blocking predicates, we have 2^n possible blocking schemes which can be constructed upon blocking predicates in the form of only conjunctions or disjunctions. Thus, the number of all possible blocking schemes which can be constructued through aribitary combinations of conjunction and disjunction of blocking predicates is more than 2^n. To effiently learn blocking schemes, we therefore propose a hierarchical blocking scheme learning strategy called *active branching* to avoid enumerating all possible blocking schemes and reduce the number of candidate blocking schemes to $\frac{n(n+1)}{2}$.

Given a blocking scheme s, there are two types of branches through which we can extend s with another blocking predicate: conjunction and disjunction. Let s_1 and s_2 be two blocking schemes, we have the following lemmas.

Lemma 1. *For the conjunction of s_1 and s_2, the following holds:*

$$|fp(B_{s_i})| \geq |fp(B_{s_1 \wedge s_2})|, \ where\ i = 1, 2 \tag{4}$$

Proof. For any true negative record pair $t \notin B_{s_1}$, we have $t \notin B_{s_1 \wedge s_2}$, which means $|tn(B_{s_1})| \leq |tn(B_{s_1 \wedge s_2})|$. Since the sum of true negatives and false positives is constant for a given dataset, we have $|fp(B_{s_1})| \geq |fp(B_{s_1 \wedge s_2})|$. □

Lemma 2. *For the disjunction of s_1 and s_2, the following holds:*

$$\frac{|fn(B_{s_i})|}{|tp(B_{s_i})|} \geq \frac{|fn(B_{s_1 \vee s_2})|}{|tp(B_{s_1 \vee s_2})|}, \ where\ i = 1, 2 \tag{5}$$

Proof. For any true positive record pair $t \in B_{s_1}$, we have $t \in B_{s_1} \cup B_{s_2} = B_{s_1 \vee s_2}$. This is, the number of true positives generated by s_1 cannot be larger than that generated by $s_1 \vee s_2$, i.e. $|tp(B_{s_1})| \leq |tp(B_{s_1 \vee s_2})|$. Since the sum of true positives and false negatives is constant, we have $|fn(B_{s_1})| \geq |fn(B_{s_1 \vee s_2})|$. □

Based on Lemmas 1 and 2, we develop an active branching strategy as follows. First, a locally optimal blocking scheme is learned from a set of candidate schemes. Then, by Lemma 1, the locally optimal blocking scheme is extended. If no locally optimal blocking scheme is learned, the strategy selects the one with minimal error rate and extends it in disjunction with other blocking predicates to reduce the error rate, according to Lemma 2. The extended blocking schemes are then used as a candidate scheme for active sampling to select more samples. Based on more samples, active branching strategy adaptively refines the locally optimal scheme. This process iterates until the label budget is used out.

4.3 Algorithm Description

We present the algorithm called Active Scheme Learning (ASL) used in our framework. A high-level description is shown in Algorithm 1. Let S be a set of blocking schemes, where each blocking scheme $s_i \in S$ is a blocking predicate at the beginning. The budget usage is initially set to zero, i.e. $n = 0$. A set of feature vectors is selected from the dataset as seed samples (lines 1 and 2).

After initialization, the algorithm iterates until the number of samples in the training set reaches the budget limit (line 3). At the beginning of each iteration, the active sampling strategy is applied to generate a training set (lines 4 to 10). For each blocking scheme $s_i \in S$, the samples are selected in two steps: (1) firstly, the balance rate of this blocking scheme s_i is calculated (lines 5 and 7), (2) secondly, a feature vector to reduce this balance rate is selected from the dataset (lines 6 and 8). Then the samples are labeled by the human oracle and stored in the training set T. The usage of label budget is increased, accordingly (lines 9 and 10).

A locally optimal blocking scheme s is searched among a set of blocking schemes S over the training set, according to a specified error rate ϵ (line 11). If it is found, new blocking schemes are generated by extending s to a conjunction with each of the blocking schemes in S_{prev} (lines 12 and 13). Otherwise a blocking scheme with the minimal error rate is selected and new schemes are generated using disjunctions (lines 14 to 16).

Algorithm 1: Active Scheme Learning (ASL)

```
   Input: Dataset: $D$
          Error rate $\epsilon \in [0, 1]$
          Human oracle $\zeta$
          Set of blocking predicates $P$
          Sample size $k$
   Output: A blocking scheme $s$
1  $S = S_{prev} = P$, $n = 0$, $T = \emptyset$, $X = \emptyset$
2  $X = X \cup$ RANDOM_SAMPLE($D$)
3  while $n < budget(\zeta)$ do
4      for each $s_i \in S$ do                          // Begin active sampling
5          if $\gamma(s_i, X) \leq 0$ then
6              $X = X \cup$ SIMILAR_SAMPLE($D, s_i, k$)
7          else
8              $X = X \cup$ DISSIMILAR_SAMPLE($D, s_i, k$)
9          $n = |X|$                                    // End active sampling
10     $T = T \cup \{(x_i, \zeta(x_i)) | x_i \in X\}$   // Add labeled samples into $T$
11     $s =$ FIND_OPTIMAL_SCHEME($S, T, \epsilon$); $S_{prev} = S$ // Begin active branching
12     if FOUND($s$) then
13         $S = \{s \wedge s_i | s_i \in S_{prev}\}$
14     else
15         $s =$ FIND_APPROXIMATE_SCHEME($S, T, \epsilon$)
16         $S = \{s \vee s_i | s_i \in S_{prev}\}$      // End active branching
17 Return $s$
```

5 Experiments

We have evaluated our approach to answer the following two questions: (1) How do the error rate ϵ and the label budget affect the learning results in our approach? (2) What are the accuracy and efficiency of our active scheme learning approach compared with the state-of-the-art approaches?

5.1 Experimental Setup

Our approach is implemented in Python 2.7.3, and is run on a server with 6-core 64-bit Intel Xeon 2.4 GHz CPUs, 128GBytes of memory.

Datasets: We have used four datasets in the experiments: (1) *Cora*[1] dataset contains bibliographic records of machine learning publications. (2) *DBLP-Scholar*[1] dataset contains bibliographic records from the DBLP and Google Scholar websites. (3) *DBLP-ACM* [14] dataset contains bibliographic records from the DBLP and ACM websites. (4) *North Carolina Voter Registration (NCVR)*[2] dataset contains real-world voter registration information of people from North Carolina in the USA. Two sets of records collected in October 2011 and December 2011 respectively are used in our experiments. The characteristics of these data sets are summarized in Table 1, including the number of attributes, the number of records (for each dataset) and the class imbalance ratio.

Table 1. Characteristics of datasets

Dataset	# Attributes	# Records	Class Imbalance Ratio
Cora	4	1,295	1 : 49
DBLP-Scholar	4	2,616 / 64,263	1 : 31,440
DBLP-ACM	4	2,616 / 2,294	1 : 1,117
NCVR	18	267,716 / 278,262	1 : 2,692

Baseline approaches: Since no active learning approaches were proposed on blocking scheme learning, we have compared our approach (ASL) with the following three baseline approaches: (1) *Fisher* [13]: this is the state-of-the-art unsupervised scheme learning approach proposed by Kejriwal and Miranker. Details of this approach have been outlined in Section 2. (2) *TBlo* [9]: this is a traditional blocking approach based on expert-selected attributes. In the survey [6], this approach has a better performance than the other approaches in terms of the F-measure results. (3) *RSL (Random Scheme Learning)*: it uses random sampling, instead of active sampling, to build the training set and learn blocking schemes. We run the RSL ten times, and present the average results of blocking schemes it learned.

[1] Available from: *http://secondstring.sourceforge.net*

[2] Available from: *http://alt.ncsbe.gov/data/*

Measurements: We use the following common measures [6] to evaluate blocking quality: *Reduction Ratio (RR)* is one minus the total number of record pairs in blocks divided by the total number of record pairs without blocks, which measures the reduction of compared record pairs. *Pairs Completeness (PC)* is the number of true positives divided by the total number of true matches in the dataset. *Pairs Quality (PQ)* is the number of true positives divided by the total number of record pairs in blocks. *F-measure (FM)* $FM = \frac{2*PC*PQ}{PC+PQ}$ is the harmonic mean of PC and PQ. In addition to these, we define the notion of *constraint satisfaction* as $CS = \frac{N_s}{N}$, where N_s is the times of learning an optimal blocking scheme by an algorithm and N is the times the algorithm runs.

5.2 Experimental Results

Now we present our experimental results in terms of the constraint satisfaction, blocking quality, blocking efficiency and label cost.

Constraint Satisfaction We have conducted experiments to evaluate the constraint satisfaction. In Figure 2, the results are presented under different error rates $\epsilon \in \{0.1, 0.2, 0.4, 0.6, 0.8\}$ and different label budgets ranging from 20 to 500 over four real datasets. We use the total label budget as the training label size for RSL to make a fair comparison on active sampling and random sampling. Our experimental results show that random sampling with a limited label sizes often fails to produce an optimal blocking scheme. Additionally, both error rate and label budget can affect the constraint satisfaction. As shown in Figure 2(a)-(d), when the label budget increases, the CS value goes up. In general, when ϵ becomes lower, the CS value decreases. This is because a lower error rate is usually harder to achieve, and thus no scheme that satisfies the error rate can be learned in some cases. However, if the error rate is set too high (e.g. the red line), it could generate a large number of blocking schemes satisfying the error rate, and the learned blocking scheme may vary depending on the training set.

Blocking Quality We present the experimental results of four measures (i.e. RR, PC, PQ, FM) for our approach and the baseline approaches. In Figure 3(a), all the approaches yield high RR values over four datasets. In Figure 3(b), the PC values of our approach are not the highest over the four datasets, but they are not much lower than the highest one (i.e. within 10% lower except in DBLP-Scholar). However, out approach can generate higher PQ values than all the other approaches, from 15% higher in NCVR (0.9956 vs 0.8655) to 20 times higher in DBLP-ACM (0.6714 vs 0.0320), as shown in Figure 3(c). The FM results are shown in Figure 3(d), in which our approach outperforms all the baselines over all the datasets.

Blocking Efficiency Since blocking aims to reduce the number of pairs to be compared in entity resolution, we evaluate the efficiency of blocking schemes by the number of record pairs each approach generates. As shown in Table 2, *TBlo* generates the minimal number of record pairs in Cora. This is due to the scheme that is manually selected by domain experts. *Fisher* targeted to learn disjunctive schemes, which can lead to large blocks, thus the number of record pairs is the

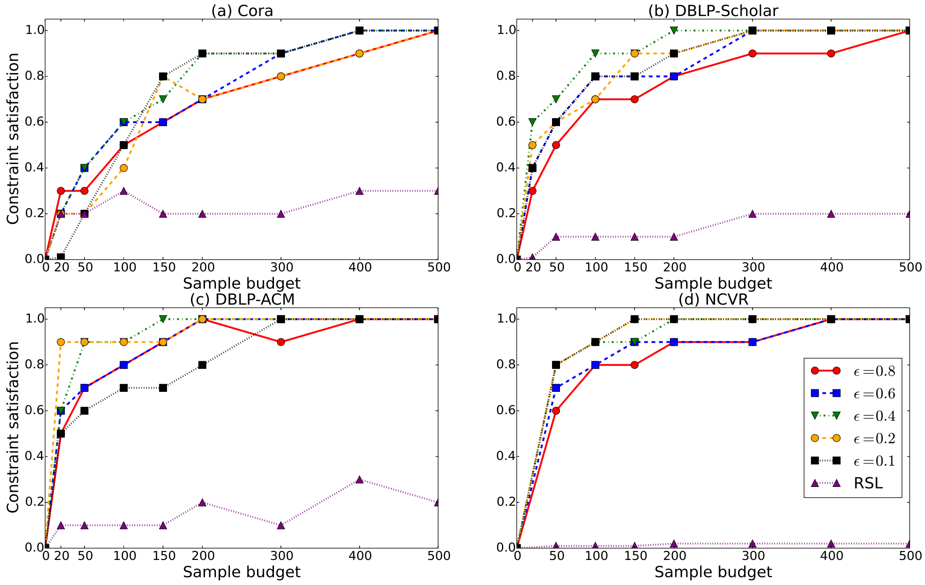


Fig. 2. Comparison on constraint satisfaction by ASL (with different error rates) and RSL under different label budgets over four real datasets

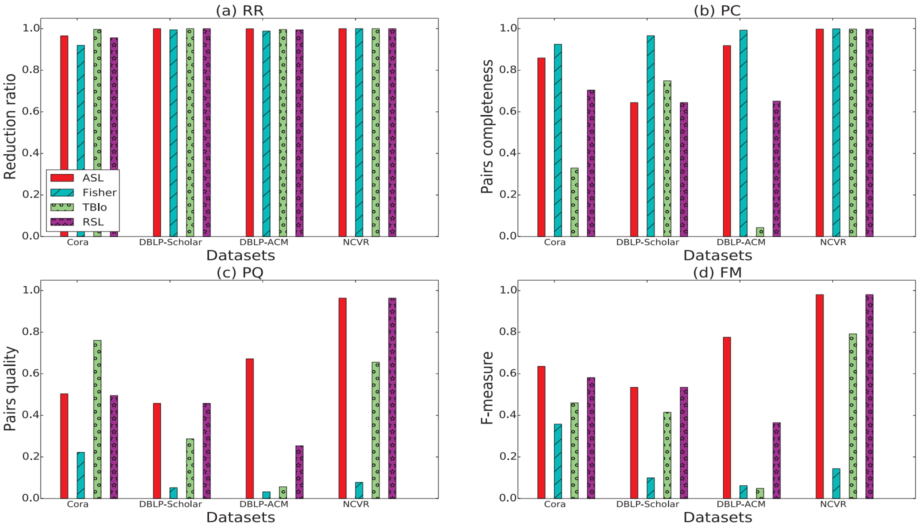


Fig. 3. Comparison on blocking quality by different blocking approaches over four real datasets using the measures: (a) RR, (b) PC, (c) PQ, and (d) FM

largest over four datasets. *ASL* considers a trade-off between PC and PQ, and the number of record pairs is often small. In *RSL*, we use the same label size as *ASL*, thus it may learn a blocking scheme that is different from the one learned by RSL, and accordingly generates different numbers of record pairs for some datasets such as Cora and DBLP-ACM. When a sufficient number of samples is used, the results of *ASL* and *RSL* would be the same.

Table 2. Comparison on the number of record pairs generated by different approaches

	TBlo	Fisher	ASL	RSL
Cora	**2,945**	67,290	29,306	17,974
DBLP-Scholar	6,163	1,039,242	**3,328**	**3,328**
DBLP-ACM	25,279	69,037	**3,043**	17,446
NCVR	932,239	7,902,910	**634,121**	**634,121**

Table 3. Comparison on label cost by ASL and RSL over four real datasets

Error rate	Cora	DBLP-Scholar	DBLP-ACM	NCVR
0.8	600	500	300	300
0.6	**400**	350	200	350
0.4	450	**250**	**150**	250
0.2	550	300	200	**150**
0.1	500	**250**	300	200
RSL	8,000	10,000+	2,500	10,000+

Label Cost In order to compare the label cost required by ASL and RSL for achieving the same block quality, we present the numbers of labels needed by our approach to generate a blocking scheme with CS=100% under different error rates, and compare them with the labels required by RSL in Table 3. In our experiments, the label budget for ASL under a given error rate starts with 50, and then increases by 50. The label budget for RSL starts with 500, and increases by 500 each time. Both ASL and RSL algorithms terminate when the learned blocking schemes remain the same in ten consecutive runs.

6 Conclusions

In this paper, we have used active learning techniques to develop a blocking scheme learning approach. Our approach overcomes the weaknesses of existing works in two aspects: (1) Previously, supervised blocking scheme learning approaches require a large number of labels for learning a blocking scheme, which is an expensive task for entity resolution; (2) Existing unsupervised blocking scheme learning approaches generate training sets based on the similarity of record pairs, instead of their true labels, thus the training quality can not be guaranteed. Our experimental results show that the proposed approach outperforms the baseline approaches under a specific error rate with a sample budget.

References

1. A. Arasu, M. Götz, and R. Kaushik. On active learning of record matching packages. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 783–794, 2010.
2. K. Bellare, S. Iyengar, A. G. Parameswaran, and V. Rastogi. Active sampling for entity matching. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 1131–1139, 2012.
3. M. Bilenko, B. Kamath, and R. J. Mooney. Adaptive blocking: Learning to scale up record linkage. In *Proceedings of the 6th International Conference on Data Mining*, pages 87–96, 2006.
4. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
5. P. Christen. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer, 2012.
6. P. Christen. A survey of indexing techniques for scalable record linkage and deduplication. *IEEE Transactions on Knowledge and Data Engineering*, 24(9):1537–1555, 2012.
7. S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 208–215, 2008.
8. S. Ertekin, J. Huang, L. Bottou, and L. Giles. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 127–136, 2007.
9. I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.
10. J. Fisher, P. Christen, and Q. Wang. Active learning based entity resolution using Markov logic. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 338–349, 2016.
11. A. Gruenheid, X. L. Dong, and D. Srivastava. Incremental record linkage. *Proceedings of the VLDB Endowment*, 7(9):697–708, 2014.
12. Y. Hu, Q. Wang, D. Vatsalan, and P. Christen. Improving temporal record linkage using regression classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 561–573, 2017.
13. M. Kejriwal and D. P. Miranker. An unsupervised algorithm for learning blocking schemes. In *Proceedings of the 13th International Conference on Data Mining*, pages 340–349, 2013.
14. H. Köpcke, A. Thor, and E. Rahm. Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment*, 3(1-2):484–493, 2010.
15. M. Michelson and C. A. Knoblock. Learning blocking schemes for record linkage. In *Proceedings of the 21st Association for the Advancement of Artificial Intelligence*, pages 440–445, 2006.
16. Q. Wang, M. Cui, and H. Liang. Semantic-aware blocking for entity resolution. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):166–180, 2016.
17. Q. Wang, J. Gao, and P. Christen. A clustering-based framework for incrementally repairing entity resolution. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 283–295, 2016.
18. Q. Wang, D. Vatsalan, and P. Christen. Efficient interactive training selection for large-scale entity resolution. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 562–573, 2015.