The Open Automation and Control Systems Journal, 2014, *6*, 1277-1286

TA-DRD: A Three-step Automatic Duplicate Record Detection

Yongquan Dong*, Ping Ling, Yali Liu and Qiang Chu

School of Computer Science and Technology, Jiangsu Normal University, Xuzhou 221116, China

Abstract: Duplicate record detection is a key step in Deep Web data integration, but the existing approaches do not adapt to its large-scale nature. In this paper, a three-step automatic approach is proposed for duplicate record detection in Deep Web. It firstly uses cluster ensemble to select initial training instance. Then it utilizes tri-training classification to construct classification model. Finally, it uses evidence theory to combine the results of multiple classification models to construct the domain-level duplicate record detection model which can be used for large-scale duplicate record detection in the same domain. Experimental results show that the proposed approach is better than previous work and and the domain-level duplicate record detection model can get high performance.

Keywords: Duplicate Record Detection, Deep Web, Data Integration.

1. INTRODUCTION

With the rapid development of Internet, the data on the Web are growing at an unprecedented rate. These data are often “hidden” in plenty of web databases and the users can only get them by submitting the queries to the interfaces. These data are often referred as Deep Web. Nowadays, the number of web databases in Deep Web is numerous, even in the same area, the number is also surprising. Therefore, many Deep Web data integration applications often have to face a lot of web databases. In many subject areas, such as book, movies, hotel and so on, there are many duplicate records among web databases. We call the process of identifying duplicate records from the extracted data records as Duplicate Record Detection which is also referred as Record Linkage or Entity Resolution.

Duplicate record detection has received many concerns and research. In the early years, the rule-based approach [1] mainly customized the rules by hands to perform duplicate record detection according to the requirements of specific application domains and users. In order to achieve high accuracy, it inevitably needs a lot of manual involvement and only adapts to specific domains. To solve the limitation of making rules, the researchers put forward the methods based on supervised learning [2], which detect duplicate records by learning potential rules and knowledge from the samples. The limitation of this approach is that the samples are difficult to obtain and it does not apply to web-scale duplicate record detection. In order to achieve large-scale deep web data integration, automatic duplicate record detection is the developing trend, which uses unsupervised learning approach. In this paper, we propose a three-step automatic duplicate record detection in Deep Web, which is denoted as TA-DRD. It firstly uses cluster ensemble to select initial training instance. Then it utilizes tri-training classification to construct classification model. Finally, it uses evidence theory to combine the results of multiple classification models to construct the domain-level duplicate record detection model which can be used for large-scale duplicate record detection in the same domain. Experimental results demonstrate the feasibility and effectiveness of our proposed approach and the domain-level duplicate record detection model can get high performance.

2. PRELIMINARY

In this section, we define some related concepts.

Definition 1 (Entity). An entity is something that has a distinct, separate existence.

Definition 2 (Appearance). An appearance is an occurrence of an entity in the data source, where the data source may be databases or web sites. In this paper, the appearance refers to an extracted web data record. The relationship between entity and appearance is 1:n, that is to say, an entity may have many appearances and an appearance only belongs to an entity.

Definition 3 (Duplicate Record Detection, DRD). Given two databases A and B, the record pairs of A and B is $A\times B=\{(a,b)\mid a\in A, b\in B\}$, where a and b denotes any data record in A and B respectively. The entity for record a is denoted as E(a) and E(b) for record b. The process of dividing $A\times B$ into two mutually disjoint sets M and U is called duplicate record detection, where $M=\{(a,b)\mid E(a)=E(b), a \quad A, \quad b\in B\}$, $U=\{(a,b)\mid E(a)\neq E(b), a\in A, b\in B\}$ and M is called matched record pair set(MRPS), U is called non-matched record pair set(NRPS).

Definition 4 (Comparison Vector, CV). Given two databases A and B, for record a in A and b in B, let the common attributes of A and B are {a_1,..,a_n}, then CV=sim (a.a1,b.a1),…,sim(a.an,…,b.a_n) is called Comparison Vector (CV in abbreviation) of record a and b, where $sim(a.a_i,b.a_i)$ represents the similarity between the value of attribute ai in record a and the one of attribute a_i in record b. If the value of a.a_i is equal to that of b.a_i, their similarity is 1. If the value of a.a_i is completely different to that of b.a_i, their similarity is 0. If the value of a.a_i is somewhat similar to that of b.ai, their similarity is between 0 and 1.

3. THE APPROACH OF TA-DRD

3.1. The Process of TA-DRD

The process of TA-DRD consists of five phrases:

(1) Blocking. As the total number of potential record pairs comparisons equals the product of the size of the two databases, $|A|\times|B|$, with $|\bullet|$ denoting the number of records in a database. The performance bottleneck in duplicate record detection is usually the expensive comparison of attributes between pairs of records, making it unfeasible to compare all pairs when the databases are large. To reduce the large amount of potential record pair comparisons, some blocking techniques are employed to filter the pairs which are impossibly matched. Related approaches has been done in [3, 4] and will not be discussed in detail in this paper.

(2) Building Comparison Vector. For the record pairs in the same block, CV is obtained by calculating the similarity of the values of common attributes. As there may have different attribute types, we will give the corresponding similarity measure method for each attribute type. The CVs are used to classify records pairs into matches and non-matches.

(3) Selecting training samples. In order to automatic duplicate record detection, we use unsupervised method, which is usually by clustering techniques, to get training samples. But single clustering method cannot guarantee the quality of training samples, so we will use several clustering methods and get the consistent results as training data.

(4) Classifying comparison vectors. The selected training samples can be used to train initial classifiers. Then we use the initial classifiers to classify the remaining unclassified CVs and select the CVs with high confidence to update the training data. Next the classifier is trained again and the process iterates until some criteria is satisfied. In order to improve the accuracy of DRD, we employ tri-training method.

(5) Building domain-level DRD model. In the same domain, the CVs of matched record pairs have similar characteristics. So we select multiple database pairs to build their classifying model respectively, and then utilize the evidence theory to integrate the results of these models to build the domain-level DRD model, which can be directly used to detect duplicate records among other databases in the same domain without retraining the classifying model.

3.2. Building Comparison Vectors

To build CVs, the first thing is to perform schema matching between the two databases. In this paper, we use the method in [5] to annotate web data records with the global attribute labels in the domain, by which we unify the schema of data records from different web sites. The CVs are built by calculating the similarity of the values of common attributes with some similarity measures.

In this paper, we divide the attribute into two data types: text attributes and numeric attributes. For text attributes, we use SoftTFIDF [6] to calculate the similarity, where SoftTFIDF is the relaxed version of TFIDF and the word in the document is not required to be the same with the vector space. For numeric attributes, we use the equation (1) to calculate the similarity.

$$Sim(a,b)=\frac{1}{e^{\frac{2|a-b|}{a+b}}} \tag{1}$$

where a and b are the attribute value to be compared. Equation (1) can better express the difference between two numbers. When a and b is equal, the similarity is 1, otherwise, as the absolute value of the difference between these two numbers gradually, the similarity will fell sharply.

The computing method of similarity measure for two data elements with different kinds of data types is shown as follows: If their data types are different, the similarity is 0; otherwise, it is computed according to the above corresponding method.

Given the similarity measure method between data elements, we can obtain the CV for each record pair.

3.3. Automatically Selecting Training Samples with Clustering Ensemble

Many existing techniques are based on supervised learning and thus require training samples, which is available in a small amount of data records. But in deep web data integration, there are large-scale web databases with massive data records, so the training samples cannot be obtained by hand and must be selected automatically. Through the observation, we find the following two rules: (1) a record pair that has the same or very similar values in all its record attributes will likely refer to the same entity, as it is very unlikely that two entities have very similar or even the same values in all their attributes. The similarity values in the comparison vector calculated when comparing such a pair will be 1(or close to 1) in all vector elements; (2) a record pair that has different values in all record attributes will likely refer to two different entities, as it is highly unlikely that two records that refer to the same entity have different values in all its record attributes. In this condition, the similarity values in the comparison vector will be 0(or close to 0).

Based on the above observations, it is usually easy to accurately classify a record pair as a match when its corresponding comparison vector contains mainly similarity values close to or equal to 1, and as a non-match when its similarity values are mainly close to or equal to 0. It is however much more difficult to correctly classify a pair that contains some attribute values that are similar while others are not, that is to say, its comparison vector contains some similarity value close to 1 and others close to 0.

It follows that it is possible to automatically extract training examples from the set of all CVs that with high likelihood correspond to true matches or true non-matches. These training examples can then be used in the next step to train a classifier. We use the clustering techniques to automatically select training examples, which can avoid manually labeling. However, it is difficult to determine which clustering technique is suite for DRD. So, in this paper, we propose an approach based on cluster ensemble to get the training samples. The method of cluster ensemble is to synthesize the results of different cluster algorithms or the algorithms of the same cluster with different parameters, which integrates the advantages of multiple cluster algorithms to get much better results than a single cluster algorithm. In order to use cluster ensemble method, we must solve two problems: (1) how to generate effective individual clusters; (2) how to design an ensemble function to combine the results of individual clusters.

3.3.1. Individual Clusters

Given comparison vector set $CV = \{cv_1, cv_2, ..., cv_n\} \subset R^d$, where d denotes the number of common attributes between a record pair to be compared. The i-th element cv_i of the set is a d-dimension vector $[cv_{i1}, cv_{i2}, ..., cv_{id}]^T$, where T denotes the transpose, cv_{ij} denotes the similarity of the j-th attribute of the i-th record pair, $0 \le cv_{ij} \le 1, 1 \le j \le d$. The CV that contains exact similarities in all its vector elements is denoted by m(i.e. $m_j = 1.0, 1 \le j \le d$), and the CV that contains total dissimilarities only by n(i.e. $n_j = 0.0, 1 \le j \le d$).

The aim of the training sample selection problem is to choose comparison vectors from CV that with very high likelihood correspond to true matches and true non-matches, respectively, and to insert them into two sets, the match training set X_M, and the non-match training set X_N. Generally, only a fraction of all comparison vectors will be selected for training, and thus it is expected that $|X_M| + |X_N| = |CV|$. In the following, the three approaches to training sample selection are presented in more detail.

1. Threshold-based clustering method

In this approach, one threshold for matches, $t_m (0.0 \le t_m \le 1.0)$, and one for non-matches, $t_n (0.0 \le t_n \le 1.0)$, are used to select comparison vectors that have all their similarity values either within t_m of the exact match value (1.0) or within t_n of the total dissimilarity value(0.0). More formally, CV_M and CV_N are formed as follows.

$$X_M = \{cv_i \in CV \mid (m_j - cv_{ij}) \le t_m, 1 \le j \le d\} \quad (2)$$

$$X_N = \{cv_i \in CV \mid (n_j - cv_{ij}) \le t_n, 1 \le j \le d\} \quad (3)$$

Depending on the values of tm and tn, there is the possibility that a comparison vector could be included into both training sets X_M and X_N. In such a situation, this comparison vector will be removed from both X_M and X_N, as it cannot be a good quality training sample for both matches and non-matches. For example, this would happen when t_m=t_n=0.6 for a CV which has all similarity values set to 0.5, i.e. $cv_{ij} = 0.5, 1 \le j \le d$. Through several rounds of experiments, we set t_m and t_n to be 0.7 respectively.

2. K-means clustering method

K-means is a classical clustering method, which updates the center of a cluster iteratively according to the distance. When the centers do not change any more or some criteria are satisfied, the iteration ends. The biggest problem of k-means approach is the choice of the value of k and the initial cluster centers. For DRD, we set k=3 and the initial centers are set with m, n and the comparison vector p with all element 0.5(i.e. $p_j = 0.5, 1 \le j \le d$). The aim of setting these three centers is to put comparison vectors near m to X_M, vectors near n to X_N and vectors near p to possible matches.

3. Nearest-neighbor-based clustering method

In order to control the number of comparison vectors adding into X_M and X_N effectively, we use nearest neighbor clustering method. In this approach, the comparison vectors closest to m are selected into X_M, and the comparison vectors closest to n into X_N. More formally, if N_m and N_n are the number of comparison vectors to be selected into X_M and X_N, respectively, and the distance between two comparison vectors is calculated using the Manhattan distance as $dist(cv_i, cv_j) = \sum_{k=1}^{d} |cv_{ik} - cv_{jk}|$, then the training set are formed as follows.

$$X_M = \{cv_i \in CV, cv_k \notin X_M \mid dist(m, cv_i) < dist(m, cv_k)\} \quad (4)$$

$$X_N = \{cv_i \in CV, cv_k \notin X_N \mid dist(cv_i, n) < dist(cv_k, n)\} \quad (5)$$

where N_m=$|X_M|$, N_n=$|X_N|$

As this approach selects CVs regardless if some of them contain the same values in all of their vector elements. In the worst case, the CVs selected into X_M will all be equal to m and the CVs selected into X_N will all be equal to n. This situation would not be very useful for training the classifier in next step. Thus, we only select different CVs. Through several rounds of experiments, we set N_m=N_n=$|CV| \cdot 5\%$

3.3.2. Cluster Ensemble Method

In the previous section, we introduce three kinds of clustering algorithm. Each clustering algorithm may produce different result of the same data; these results capture various distinct aspects of the data. In order to improve the quality of training samples, we combine multiple clustering algorithms to get them with high confidence. It has been shown that a meaningful consensus of multiple clusters is possible by using a consensus function that maps a given ensemble to a combined clustering result. In this paper, we use the intersection method to get corresponding consensus comparison vectors as the training samples, i.e., $X_M = \bigcap_{i=1}^{3} X_{Mi}$, $X_N = \bigcap_{i=1}^{3} X_{Ni}$.

3.4. Classifying Comparison Vectors with Tri-training Approach

Tri-training was proposed by Zhou and Li [17], which was motivated from co-training. It designs three classifiers learning from unlabeled examples *via* an unlabeled example is labeled for a classifier if the other two classifiers agree on the labeling under certain conditions. This method can release the requirement of co-training with sufficient and redundant views. Additionally, tr-training learning considers the agreements of the classifiers while selecting new samples. We just need the classification results instead of the confident scores made by the classifiers. Thus we can use any classifier in tri-training learning. In this paper, we use decision tree classifier, SVM classifier and Bayes Classifier.

Next, we use tri-training approach to iteratively classify the remaining unlabeled comparison vectors. By each iteration, the unlabeled comparison vectors are labeled by the current classifiers. Next, a subset of the comparison vectors newly labeled is selected to be added to the training data. The algorithm is shown in Algorithm 1.

Algorithm 1: Duplicate Record Detection based on Tri-traing Approach

Input : all comparison vector set *CV*, initial match set : X_M, initial non-match set: X_N, three kinds of classifiers l_1, l_2, l_3.

Output : match set Z_M , non-match set Z_N and three classifier c1,c2,c3

```
1. $Z_M = Z_N = \phi$;
2. for i = 1 to 3 do
3. $T_{Mi}=X_M$; $T_{Ni}=X_N$, $T_{Ui}= CV-(X_M \cup X_N)$;
4. $c_i$=train_classifier($l_i$, $T_{Mi}$, $T_{Ni}$);
5. end for
6. for i =1 to 3 do
7. $L_{Mi} = L_{Ni} = \phi$;
8. for x in $T_{Ui}$ do
9. if $c_j(x)== c_k(x)$ ($j,k \neq i$) then
10. if $c_j(x)$ == "match" then
11. $L_{Mi} = L_{Mi} \cup x$;
12. end if
13. if $c_j(x)$=="non-match" then
14. $L_{Ni} = L_{Ni} \cup x$;
15. end if
16. end if
17. end for
18. $T_{Mi} = T_{Mi} \cup L_{Mi}$; $T_{Ni} = T_{Ni} \cup L_{Ni}$;
19. $c_i$=train_classifier($l_i$, $T_{Mi}$, $T_{Ni}$);
20. end for
21. for x in $T_{Ui}$ do
22. x_result = majority_voting(x, $c_1,c_2,c_3$);
23. if x_result == "match" then
24. $Z_M=Z_M \cup x$;
25. end if
26. if x_result == "non-match" then
27. $Z_N=Z_N \cup x$;
28. end if
29. end for
30. return $Z_M$, $Z_N$, $c_1,c_2,c_3$;
```

Algorithm 1 starts with initiating Z_M and Z_N. Lines 2 to 5 are to train three classifiers c1, c2 and c3 based on X_M and X_N respectively by l_1, l_2 and l_3. And the X_{Ui} is used to store the unlabeled datasets for classifier c_i, where $1 \leq i \leq 3$. In lines 6 to 20, we use tri-training approach to classify the unlabeled datasets for each classifier. In line 7, L_{Mi} and L_{Ni} are initialized to be empty, which are used to store the new labeled CVs. Lines 8 to 17 judge every CV in $X_{Ui.}$ When the results of classifier c_j and c_k are same, where j,k is not equal to i (Line 9), if the result is equal to "match", x is added into L_{Mi}; if the result is equal to "non-match", x is added into L_{Ni}. After all unlabeled CVs are judged, in line 18 L_{Mi} is added into T_{Mi} and L_{Ni} is added into T_{Ni} . Then in line 19, the classifier c_i is retrained by new T_{Mi} and T_{Ni}. In Lines 21 to 29, after obtaining the three new classifier c_1,c_2,c_3, for every CV x in T_{Ui}, we use majority voting approach to get the classified result of x. If the result is match, x is added into Z_M. If the result is non-match, x is added into Z_N. In line 30, the final match set $Z_{M,}$ non-match set Z_N and three classifiers c1,c2,c3 are returned.

3.5. Building domain-level Duplicate Record Detection Model

As there are plenty of Web database on the Web, we cannot perform complete duplicate record detection steps in pairs. According to the paper [7], we can see that there is a hidden duplicate record detection model in the same domain. This model is called as domain-level duplicate record detection model (DDRDM) by which we can judge the new record pairs to get its final classification result. If we only use the model generated by two Web databases as DDRDM, it is obvious that the model is not robust. Thus, in this paper we select multiple Web database pairs to build corresponding comparison vector classification models, and then use the combination approach to integrate them to construct DDRDM. As the evidence theory is the state-of-the-art approach for combining multiple sources of uncertain information [8, 9], we use it to integrate the results from multiple comparison vector classification models.

3.5.1. D-S Evidence Theory

Evidence theory is formalized by A.P. Dempster [8] and developed by G.Shafer [9], which is also called Dempster-Shafer theory of evidence(D-S Evidence Theory). D-S Evidence theory is a well-known framework for representing and reasoning with uncertain, imprecise and incomplete information. This theory can be considered as a generalized Bayesian theory. In D-S evidence theory, a set of propositional hypotheses is represented by a frame of discernment.

Definition 5 (Frames of Discernment). A frame of discernment, usually denoted as Θ, contains mutually exclusive and exhaustive propositional hypotheses, one and only one of which is true.

Definition 6 (Mass Functions). A function, m: $2^{\Theta} \rightarrow [0,1]$, is called a mass function on a frame Θ if it satisfies the following two conditions:

$$m(\varnothing)=0 \tag{i}$$

$$\sum_{A \in 2^{\Theta}} m(A) = 1 \tag{ii}$$

where Θ is an empty set and A is any subset of Θ.

Given a frame of discernment, Θ, for each source of evidence, a mass function assigns a mass to every subset of Θ, which represents the degree of belief that one of the answers in the subset is true, given the source of evidence. For example, when the patient has been observed having the symptom "coughing", the degree of belief that the patient has "flu" or "cold" is 0.64 and the degree of belief that the patient has "pneumonia" is 0.35, that is $m_1(\{C,F\}) = 0.65$ and $m_1(\{P\}) = 0.35$. Similarly, with the symptom of "sniveling", we have another mass function: $m_2(\{F\}) = 0.75, m_2(\{C\}) = 0.15$ and $m_2(\{P\}) = 0.1$.

When more than one mass function is given on the same frame of discernment, the theory also provides Dempster combination rule. Given *n* mass functions $m_1, m_2, ..., m_n$ on frame Θ, the Dempster combination rule is defined as follows.

$$m(A) = \frac{\sum_{\cap A_i = A} \prod_{1 \le i \le n} m_i(A_i)}{1 - \sum_{\cap A_i = \phi} \prod_{1 \le i \le n} m_i(A_i)} \tag{6}$$

Dempster combination rule computes a measure of agreement between multiple bodies of evidence concerning various propositions discerned from a common frame of discernment. The combination rule combines the belief values from multiple mass functions to produce a single belief value derived from different aspects. In the above example, we use (3) to combine two mass functions, m_1 and m_2, to get $m(\{F\}) = 0.79$, $m(\{C\}) = 0.16$ and $m(\{P\}) = 0.06$. Therefore given the two symptoms the patient has, it is more likely that he is having "flu".

3.5.2. DDRDM Construction

The approach of DDRDM construction is shown as follows.

(1) A frame of discernment of DDRDM is defined as $\Theta = \{O_1, O_2\}$, where O_1 denotes the match class and O_2 denotes the non-matched class.

(2) Multiple base classifiers $BC_1, BC_2, \cdots, BC_n$ are constructed, each of which corresponds to the comparison vector classification model of a web database pair, where n denotes the number of selected web databases. Every classifier has two outputs, each of which corresponds to relevant class. For example, the output j represents the class O_j.

(3) Next every output of each base classifier is assigned a mass probability. That is to say, for comparison vector CV, the mass probability of the jth output of the ith base classifier is denotesd as $m_i(O_j)$, which represents the degree of the CV classified as jth class by ith base classifier, where m_i represents the ith mass function. Note that, the value of $m_i(O_j)$ can be obtained from the result of CV classification by the base classifier BC_i.

(4) Finally, Dempster combination rule is used to combine multiple basic mass functions. In combination results of comparison vector CVs, the class of the maximum value is selected as the result of DDRDM.

4. EXPERIMENTS

4.1. Datasets

The following data sets are used to evaluate our approach.

(1) Cora dataset

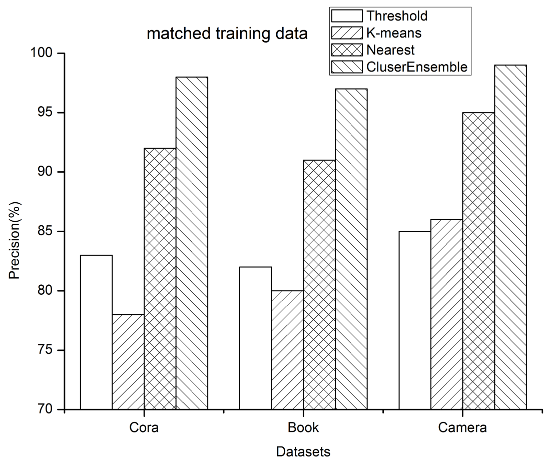
It is a dataset for references, containing 1878 records with 194 distinct one. As the maximum value of the number of the duplicate records for the same entity is 119, we divide Cora into 119 subsets, each of which is a data source. Each duplicate record is randomly assigned to any data source but not to the same one, which guarantees that every data source contains distinct reference.

(2) Book dataset

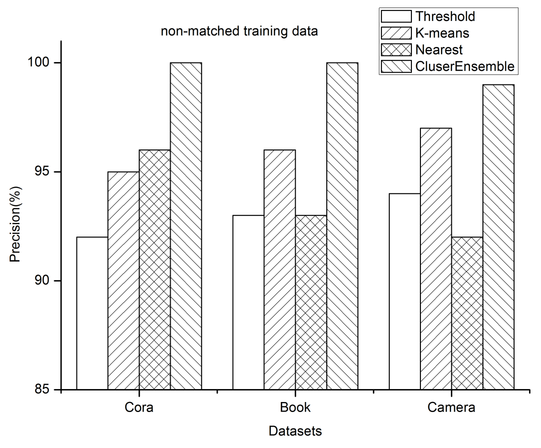
It contains 5100 book records collected from 20 online book websites. The data is tagged by hand with 768 distinct book records.

(3) Camera dataset

It contains 4600 camera records collected from 20 shopping websites. After manually tagging, there are 668 distinct camera records.



(**a**) Performance comparison of selecting matched training samples



(**b**) Performance comparison of selecting non-matched training samples

Fig. (1). Performance comparison of selecting training samples using cluster ensemble approach and single clustering approach.

4.2. Performance Metrics

We measure the performance of our approach *via* three metrics: precision, recall and F1.

Suppose A denotes the number of duplicate record pairs identified by our approach, and B the number of correct duplicate record pairs identified by our approach and C the number of correct duplicate record pairs not identified by our approach. Then the three metrics are defined as follows.

$$\text{Precision} = \frac{B}{A}, \quad \text{Recall} = \frac{B}{B+C}, \quad \text{F1} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{7}$$

Precision represents the confidence of the result, and Recall represents the coverage of correct results and F1 synthesizes precision and recall.

4.3. Discussion on Experimental Results

We have designed five experiments to evaluate the effects of our approach.

4.3.1. Performance Comparison of Selecting Training Examples Using Ensemble Approach and Single Clustering Approach

We first compare the performance of selecting training examples using ensemble approach against using single clustering approach. Threshold-based clustering approach is denoted as Threshold with t_m=t_n=0.7. In *k*-means clustering approach which is denoted as k-means, we select the number of clusters as three, and the centers of three clusters as comparison vectors with all elements as 0,1,0.5. In nearest method which is denoted as Nearest, the parameters are N_m=N_n=$|X|\cdot 5\%$, where $|X|$ represents the number of all record pairs. Fig. (**1**) is the precision comparison of the four approaches on three datasets.

Fig. (**1**) shows that the precision of selecting matched pairs and non-matched pairs using ensemble approach is higher than that of the other three approaches. And the precisions both achieve about 98%. From Fig. (**1**), we can also see that the precision of selecting non-matched pairs is all over 90%, which is higher than that of selecting matched pairs. This is because that the number of non-matched pairs is much bigger than that of matched pairs in the datasets.

4.3.2. Performance Comparison Between Our Approach and the Approaches in [10, 11]

We compare our approach with the work of Tailor and UDD on three datasets. We select 50 web database pairs as testing data on the three datasets, use three approaches to performance duplicate record detection and use the average value as the evaluation criteria. Tailor [10] first uses k-means to cluster all record pairs as three classes, then use matched pairs and non-matched pairs to train a SVM model. UDD [11] regards the record pairs from the same data source as non-matched pairs. Then it uses two cooperating classifiers to iteratively identify duplicate records. Fig. (**2**) shows the performance of three approaches on three datasets.

From Fig. (**2**), we can see that our approach is superior to Tailor and UDD. The reason for Tailor is that the precision of training samples is low. Although UDD also uses iterative learning approach, its initial assumption has limitations, that is to say, the record pairs from the same data source has some matched pairs.

4.3.3. The Effect of The Number of Web Database Pairs on DDRDM

We test the effect of the number of web database pairs on DDRDM. We randomly select 5,10,15,20 and 25 web database pairs, and use these web database pairs to train individual model. Then we use the approach in section 3.5.2 to build DDRDM. We select 50 web database pairs as testing data from three datasets. For simplicity, we use F1 as the evaluation criteria. We use DDRDM on testing data and use the average value of all F1 as the final result. Fig. (**3**) shows the average F1 change with the change of number of web database pairs on three datasets.

Fig. (**3**) shows that on three datasets with the increase of the number of web database pairs, the average F1 is a gradually improvement. But when the number of web database pairs achieve above 15, the increase is not obvious. This suggests that the construction of the DDRDM is feasible.

4.3.4. Performance Comparison of DDRDM Using Evidence Theory and Weighted Average Combination

We compare the performance of DDRDM using evidence theory against that using weighted average combination. According to the result of section 4.3.3, we randomly select 15 web database pairs from three datasets, and learn 15 comparison vector classifiers and then respectively use evidence theory and weighted average combination to get the DDRDM. Then we select 100 web database pairs from three datasets as testing data. As weighted average combination needs to determine the weights. In this paper, we use linear regression approach to learn the weight of every comparison vector classifier. We also select average F1 as evaluation criteria. Fig. (**4**) gives the average F1 of DDRDM using evidence theory and that using weighted average combination.

From Fig. (**4**), we can see that the average F1 of DDRDM using evidence theory is higher than that using weighted average combination. As evidence theory does not need the weights, we can directly use dempster combination rule to get DDRDM, which is more suitable for constructing DDRDM.

4.3.5. Performance of DDRDM

We compare the performance using DDRDM and that using the model by one web database pair. According to the result of section 4.3.3, we randomly select 15 web database pairs to learn DDRDM. Then we randomly select 100 web database pairs as testing data. We also select average F1 as evaluation criteria. Fig. (**5**) gives the performance comparison of average F1 using DDRDM and the model directly built by one database pair.

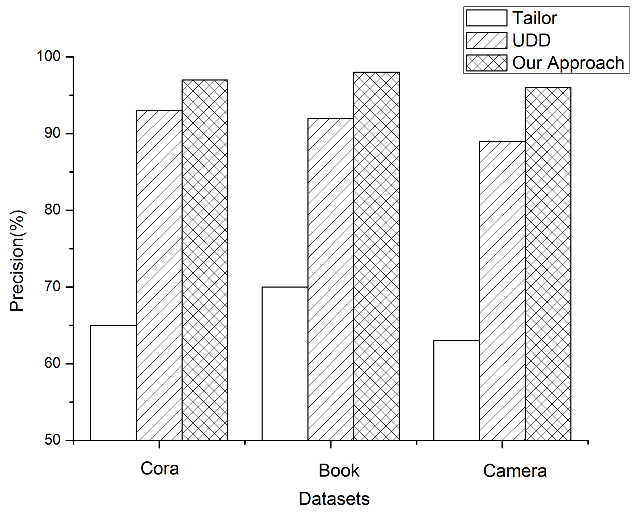
From Fig. (**5**), we can see that on three datasets, the difference of average F1 is very small using DDRDM and using the model directly built by one database pair. This suggests that DDRDM has a good performance for the domain, which can be used to duplicate detection among a large number of web databases in the domain.

5. RELATED WORKS

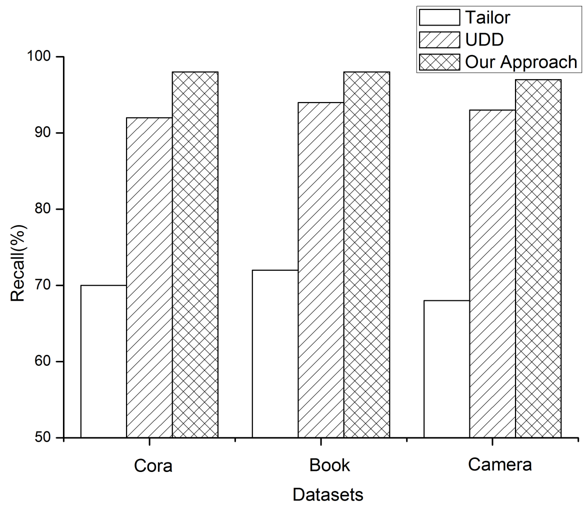
In deep web data integration, many web databases have abundant duplicate records, so duplicate record detection is an important problem. Duplicate record detection is to identify the records from different data sources referring to the same entity. There have been many approaches to solve the problem. The classical probabilistic approach [12], as developed by Fellegi and Sunter, has been improved mainly through application of the expectation-maximization(EM) algorithm for better parameter estimation in record pair classification [13], and through the use of approximate string comparisons to calculate partial agreement weights when attribute values have typographical variations [14].

In the late 1990s, many researchers started to explore the use of techniques, such as machine learning, data mining, information retrieval and database research to improve the duplicate record detection. Many of these approaches are based on supervised learning techniques and assume that training data is available. However, such training examples are often not available in real world, or have to be prepared manually.

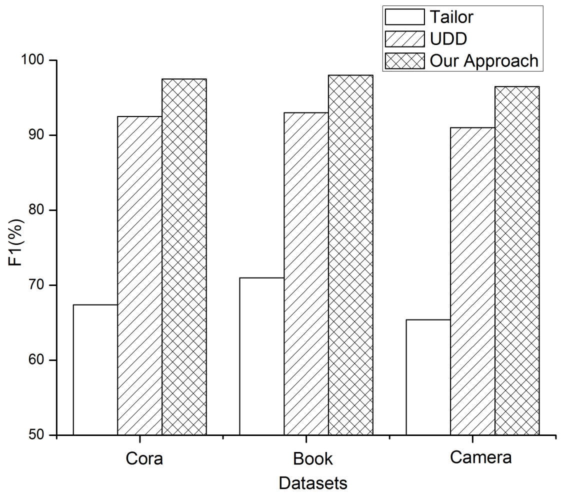
In order to perform duplicate record detection among a large number of web databases, unsupervised approach is the trend, which has been discussed in [10, 11, 15, 16]. M. Elfeky proposed TAILOR method [10], which uses unsupervised k-means method to get three clusters (matched, possible matched and non-matched) and learns a decision tree classifier model with matched and non-matched clusters. Then the model is used to classify the remaining record pairs. This approach cannot guarantee the precision of the training data for the decision tree, which affects the results for duplication record detection. Lifang Gu [15] proposed an improved clustering decision model to detect duplicate records,



(**a**) Precision



(**b**) Recall



(**c**) F1

Fig. (2). Performance comparison between Our approach and Tailor, UDD on three datasets.

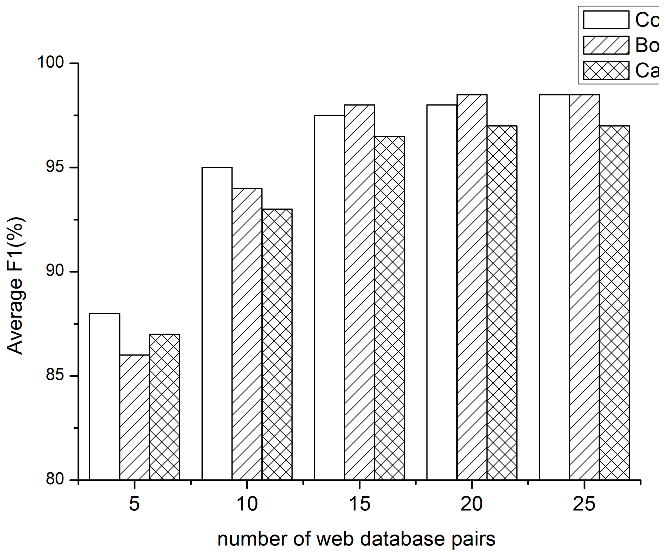


Fig. (3). The effect of the number of Web database pairs on DDRDM.

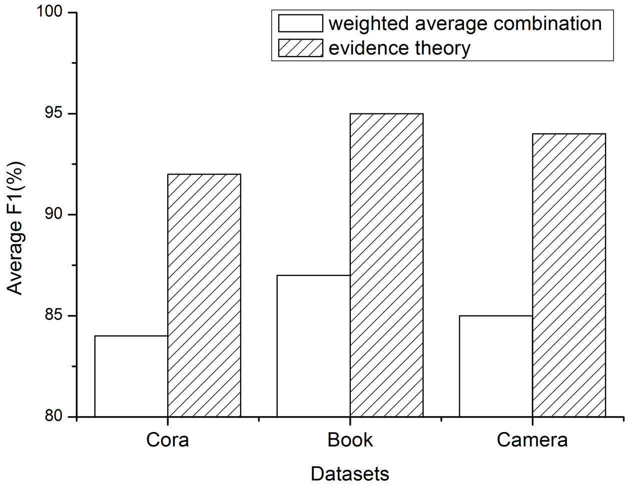


Fig. (4). Performance comparison of using evidence theory and using weighted average combination on DDRDM.

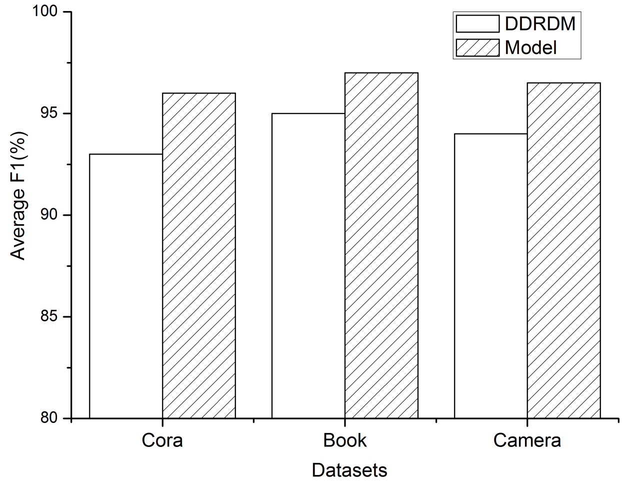


Fig. (5). Performance comparison of using DDRDM and using the model by Web database pairs**.**

which added fuzzy regions for users to adjust the clustering result of matched and unmatched record pairs. This approach achieves better result than TAILOR. However, this approach only considers the clustering and does not further combine the classifier for duplicate record detection. Peter Christen proposed a two-step approach [16] to record pair classification. In a first step, example training data is generated automatically, and then used in a second step to train a classifier. It has the following problem. In the first step, it only uses single clustering approach to generate training data whose precision is not very high. And it does not consider plenty of web databases in the same domain. Weifeng Su proposed a UDD approach [15] which focuses on duplicate record detection by ad-hoc query. This approach regards the record pairs from the single data source as non-matched. Starting from the non-matched record pairs, they use two cooperating classifiers to iteratively identify duplicate records in the query results from multiple Web databases. Its initial assumption has limitations, that is to say, the record pairs from the same data source has some matched pairs. In the same time, they also do not consider the scale of web databases.

CONCLUSION

Duplicate record detection is an important step in deep web data integration and most state-of-the-art methods do not adapt to its large-scale nature. In this paper, we presented a three-step automatic approach, TA-DRD, for detecting duplicate records. It firstly uses cluster ensemble to select initial training instance. Then it utilizes tri-training classification to construct classification model. Finally, it uses evidence theory to combine the results of multiple classification models to construct the domain-level duplicate record detection model which can be used for large-scale duplicate record detection in the same domain. Experimental results show that the proposed approach is comparable to previous work and the domain-level duplicate record detection model can get high performance.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China under Grant No. 61100167 and 61105129, the Natural Science Foundation of Jiangsu Province, China under Grant No. BK2011204, Qing Lan Project, the National Training Program of Innovation and Entrepreneurship for Undergraduates under Grant No. 201310320052, the Practice Innovation Training Program Project for the Jiangsu College Students under Grant No. 201310320052Z and the Graduate Scientific Research and Innovation Project of Jiangsu Normal University under Grant No. 2013YYB130.

REFERENCES

[1] H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C.A. Saita, "Declarative data cleaning: language, model, and algorithms", In: *Proceedings of the 27th International Conference on Very Large Data Bases*, San Francisco, CA, USA, 2001, pp. 371-380.

[2] M. Bilenko, R. Mooney, W. Cohen, and P. Ravikumar, "Adaptive name matching in information integration", *IEEE Intelligent Systems*, vol. 18, no. 5, pp. 16-23, 2003.

[3] G. Papadakis, G. Koutrika, T. Palpanas, and W. Nejdl. "Meta-blocking: taking entity resolution to the next level", *IEEE Transaction on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1946-1960, 2014.

[4] G. Papadakis, E. Ioannou, T. Palpanas, C. Nidederee, and W. Nejdl. "A blocking framework for entity resolution in highly heterogeneous informatino spaces", *IEEE Transaction on Knowledge and Data Engineering*, vol. 25, no. 12, pp. 2665-2682, 2013.

[5] Y.Q. Dong, Q.Z. Li, Y.Q. Zheng, X. Xu, and Y. Zhang, "Semantic annotation of web objects using constrained conditional random fields," *Lecture Notes in Computer Science*, vol. 6184, pp. 28-39, 2010.

[6] W.W. Cohen, P.D. Ravikumar, and S.E. Fienberg, "A comparison of string distance metrics for name-matching tasks", In: *Proceedings of IJCAI-03 Workshop on Information Integration on the Web*, Acapulco, Mexico, 2003, pp. 73-78.

[7] W. Liu, *Key Techniques on Deep Web Data Integration*, Doctoral Dissertation, Renmin University of China, 2008.

[8] A.P. Dempster, "Upper and lower probabilities induced by multi-valued mapping," *The Annals of Mathematical Statistics*, vol. 38, no. 2, pp. 325-339, 1967.

[9] G.A. Shafer, *Mathematical Theory of Evidence*, Princeton, Princeton University Press: USA, 1976.

[10] M.G. Elfeky, A.K. Elmagarmid, and V.S. Verykios. "Tailor: a record linkage tool box," In: *Proceedings of the 18th International Conference on Data Engineering*, Washington, DC, USA, vol.17, 2002.

[11] W.F. Su, J.Y. Wang, and F. Lochovsky, "Record matching over query results from multiple web databases," *IEEE Transtraction on Knowledge and Data Engineering*, vol. 22, no. 4, pp. 578-589, 2010.

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

[13] W.E. Winkler, "Using the EM algorithm for weight computation in the fellegi-sunter model of record linkage," Technical report RR2000/05, US Bureau of the Cnsus, 2000.

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

[15] L. Gu, and B. Rohan. "Decision models for record linkage," *Lecture Notes in Computer Science*, vol. 3755, pp, 146-160, 2006.

[16] P. Christen, "Automatic record linkage using seeded nearest neighbour and support vector machine classification." In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM*, 2008.

[17] Z. H. Zhou, and M. Li., "Tri-training: Exploiting unlabeled data using three classifiers." *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529-1541, 2005.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014