BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **16**, No 2
Sofia • 2016 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2016-0028

Duplicate Literature Detection for Cross-Library Search

Wei Liu, Jianxun Zeng

Institute of Scientific and Technical Information of China, 10038 China
liuw@istic.ac.cn zeng@ istic.ac.cn

Abstract: *The proliferation of online digital libraries offers users a great opportunity to search their desired literatures on Web. Cross-library search applications can help users search more literature information from multiple digital libraries. Duplicate literatures detection is always a necessary step when merging the search results from multiple digital libraries due to heterogeneity and autonomy of digital libraries. To this end, this paper proposes a holistic solution which includes achieving automatic training set, holistic attribute mapping, and weight of attribute training. The experiments on real digital libraries show that the proposed solution is highly effective.*

Keywords: *Information integration, digital library, duplicate detection, schema mapping, data cleaning.*

1. Introduction

The rapid development of the Web makes more and more digital libraries to be accessed online. According to the survey of Complete Planet [1], there are thousands of digital libraries on the Web. This gives users not only a great opportunity but also a challenge to search their desired information from a large number of digital libraries. Cross-library search applications aim to integrate services and information of digital libraries and provide users a unified view. So users need only to submit one query to get the Literature Records (LR for short in some cases) from multiple digital libraries at same time.

There are many research issues [2-5] in the area of cross-source search, which have been widely studied. However one of the necessary steps, how to merge the search results from different digital libraries, has not drawn great attention yet. In fact, there are usually a considerable amount of duplicate literatures among digital libraries. Due to the heterogeneity and autonomy of digital libraries, one problem that can arise is that one literature in the real world can exist in inconsistent presentations. In this paper, we focus on the problem of duplicate literature detection in the context of cross-library search. To the best of our knowledge, most existing solutions on duplicate detection are manual or semi-automatic, and they can only address limited

digital libraries. In contrast, the duplicate literatures detection process needs to be as automatic as possible and scalable to large quantities of digital libraries.

Until now, there are already great deals of research works [6-11] on duplicate detection. They try to map duplicates between two sources, which result in C_n^2 implementations of duplicate detectors towards n total sources. Such expense poses a great threat to the efficiency. In addition, traditional methods assume that semantic mapping has been well built between sources. While in the context of cross library search, this assumption has to be dropped. A new challenge is then put forward: how to address the duplicate detection problem "holistically" among a lot of digital libraries? In this paper, we propose a domain-level solution to address this challenging problem. That is, our solution can detect the duplicates among multiple digital libraries. The intuition behind our solution is that each attribute in it plays a definite role on the duplicate detection problem. Let's say, the importance (or weight) of an attribute is domain-dependent instead of source-dependent. For example, for any two literature records, "title" is always more important than "author", in order to determine whether they are one literature. A survey [22] indicates that the attributes in one domain is convergent, and we argue that this phenomenon is also fit for digital library domain. For instance, the frequent attributes are title, author, affiliation, abstract, keywords, classification code, and so on. This gives us the feasibility to investigate the weights of most attributes in this domain. In order to achieve the accurate and objective weights of these attributes, there are three key techniques in our solution: **training set extracting, attribute mapping**, **and attribute weight learning**. And they will be introduced respectively in the following sections.

In summary, the contributions of this paper are: as the problem, we probe the duplicate detection problem in the context of large scale cross-library search, where lots of digital libraries bring forward an inherent challenge of finding all duplicated literatures at the same time; as our insight on the observation, we discover the attributes in the digital library domain play definite roles on the problem of duplicate detection; as the solution, we propose a holistic automatic solution on duplicate literature detection under the context of large scale cross-library search in a given domain. Our experiments show the promise of this solution.

The rest of the paper is organized as follows: In Section 2 we talk about related works. In Section 3 we present the overview of our solution. Section 4 proposes an approach to extract the training set automatically. Section 5 builds attribute mappings among digital libraries. Section 6 proposes a novel approach of weights assignment with inequalities-based metrics. An experimental evaluation of our approach is shown in Section 7. Section 8 discusses several further opportunities and then concludes the paper.

2. Related works

The goal of duplicate detection is to identify records in the same or different sources that refer to the same real world entity, even if the records are not identical. It is well known that duplicate detection have been studied for more than five decades. The

first works [13] have been proposed by Fellegi-Sunter in the late 1950s and 1960s. A recent survey [30] introduces the state-of-the-art researches.

2.1. Probabilistic approaches

[14] is the first to recognize duplicate detection as a Bayesian inference problem. The comparison vector x is the input to a decision rule that assigns x to M or to U, where M is a set of right samples, U is a set of false samples, and x is a random vector to represent each record pair. In [15] is used a binary model for the values of x_i (i.e., if the field i "matches" $x_i = 1$, else $x_i = 0$) and suggested using an Expectation Maximization (EM) algorithm [16] to compute the probabilities $p(x_i = 1|M)$. When the conditional independence is not a reasonable assumption, then Winkler [17] suggested using a generalized maximization (EM) algorithm to estimate $p(x|M)$, $p(x|U)$.

2.2. Supervised learning approaches

The supervised learning systems rely on the existence of training data in the form of record pairs, pre-labeled as matching or not. In [18] is used the well-known CART algorithm [19], which generates classification and regression trees, a linear discriminant algorithm [20], which generates linear combination of the parameters for separating the data according to their classes, and a "vector quantization" approach, which is a generalization of nearest neighbor algorithms. In [21] is used SVM light to learn how to merge the matching results for the individual fields of the records. They showed that the SVM approach usually outperforms simpler approaches, such as treating the whole record as one large field. In [23] is proposed a supervised approach in which the system learns from training data how to cluster together records that refer to the same real-world entry. In [24] is proposed using the training data for learning the clustering method and tried to find the min-cut and the appropriate number of clusters for the given data set.

2.3. Active-learning-based approaches

One of the problems with the supervised learning techniques is the requirement for a large number of training examples. In order to solve this problem, some duplicate detection systems used active learning techniques to automatically locate such ambiguous pairs. ALIAS [25] is learning based duplicate detection system which uses the idea of a "reject region" to significantly reduce the size of the training set. In [26] is used a similar strategy and employed decision trees to teach rules for matching records with multiple fields. Their method suggested that by creating multiple classifiers trained to use slightly different data or parameters, it is possible to detect ambiguous cases and then ask the user for feedback.

2.4. Distance-based approaches

Probability models require an accurate estimate of the probability parameters and counts from training data. In [27] is described a similarity metric that uses not only the textual similarity, but the "co-occurrence" similarity of two entries in a source. In [28] is proposed a distance metric that is based on ranked list merging. The basic idea

is that if we use only one field, the matching algorithm can easily find the best matches and rank them according to their similarity, putting the best matches first. In [29] is proposed a new framework for distance-based duplicate detection, observing that the distance thresholds for detecting real duplicate entries is different from each record.

Though there are lots of techniques for duplicate detection, all of them focus on the accuracy in the context of small-scale heterogeneous data integration, which are not feasible to deal with a large number of sources, while our approach is just the solution to this challenge.

3. Holistic solution

In this paper we propose a holistic solution to address the problem of duplicate detection for the application of cross-library search. Fig. 1 shows the overview of the solution. The input records are the literatures extracted from the search result web pages generated by different digital libraries. The output is a set of literature record pairs, where each pair denotes a same entity. A reasonable assumption in this paper is that all literature records from digital libraries have been extracted at the attribute level. This assumption is reasonable because in case of XML data format returned from APIs, web services or even RSS feeds, each attribute has been assigned with a specific syntax. In case of response Web pages and Web data, item extraction have been proposed in [14] and confirmed to achieve satisfying accuracy.

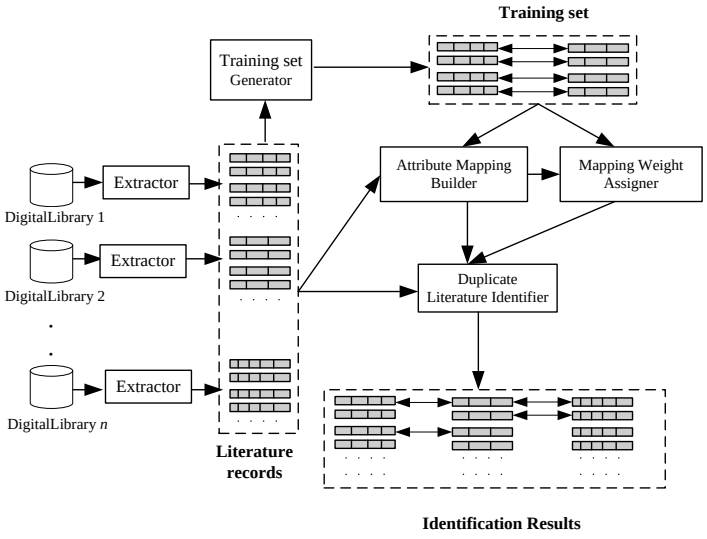

Fig. 1. Solution architecture

There are four primary components in our solution, and we introduce their functions briefly as follows.

Training set extracting. Training set generator aims to extract a set of training samples from the literature records of digital libraries automatically. Each training sample is a matched literature record pair which is judged by the training set

generator. In previous works, the used training sets are collected manually. The manual approach is a time-consuming task if the quantity of digital libraries is very large. Unfortunately, there is a small quantity of errors hiding in the training set.

Attribute mappings. The main task of this component is to build the semantic mappings between the attributes using the training set achieved by the training set generator. In this step we propose a holistic approach to accomplish the attribute mappings among all digital libraries effectively and efficiently. In order to ensure the accuracy of attribute mapping building, robust means are used to avoid the potential errors caused by the errors in the training set. And at the same time, these errors will be tried to prune. At last, the attribute mappings among all digital libraries are built.

Attribute weight learning. This step is to assign appropriate weight (importance) to each attribute. A novel supervised learning method is put forward in our solution, which overcomes the weakness of traditional machine learning methods that C_n^2 implementations are required towards n data sources. With our inequalities-based approach, a group of weights are computed out for each digital library pairs prepared in advance at first; each group of weights corresponds to the attributes shared by the digital library pair. After normalization, such kind of weights between two digital libraries is extended to the whole domain to finally achieve the weights of attributes at domain level. In addition, two thresholds, T_1 and T_2 ($T_1 > T_2$), are obtained to determine whether two literature records are matched, which are also applied at the domain level.

Entity identifier. This component aims to identify the matched literature records from different digital libraries. For any two literature records, the similarities on all shared attributes are computed respectively, and then the weighted sum is used to denote the similarity of the two literature records. If the similarity is larger than threshold T_1, the two literature records will be determined to be matched; if the similarity is smaller than the threshold T_2, this pair will be regarded as unmatched; if the similarity is between T_1 and T_2, this pair will be determined as a possibly-matched pair which needs to be further manually checked. The reason is that, in some scenarios, two literature records cannot be determined even by people without provision of additional information. The main idea of entity identifier is simple and direct, so we do not discuss it more.

The rest of this paper will focus on the technique details of training set generator, and attribute mappings builder. We will discuss them in the following three sections respectively.

4. Training set extracting

In previous related works, the training sets are always prepared manually in advance. The manual way is feasible when the number of data sources is small. But this is impossible in the context of large scale cross-library search. In this section, an automatic method is proposed to extract training samples among the literature records returned by digital libraries.

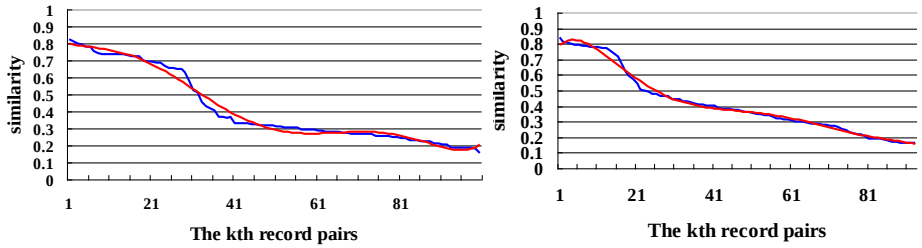

Fig. 2. The relationship of the *i*th literature record pair and its similarity

Instinctively, if two literature records from different digital libraries are determined to be matched in the real world, they often (not absolutely) share more same texts than the unmatched ones. So a naïve approach is to regard one literature records a short text document, and determine whether two literature records are matched by text similarity comparison technique, such as tf-idf function. But obviously the accuracy is not satisfying and not stable. We have done the experiment with this naïve approach to the literature records (more than 1000) in two domains (book, paper), and the accuracies are only about 83% and 47% respectively. We check the experimental results manually and divide all matched data record pairs into right literature record pairs and wrong literature record pairs. The right literature record pairs refer to the really same entities, while the wrong data record pairs are not. If we rank all matched literature record pairs by their tf-idf similarity from high to low, we get a literature record pair sequence. Most right ones are congregated in the head of this sequence, while most wrong ones are in the tail of it. This phenomenon motivates us to get the right ones from the head of this sequence.

Fig. 2 shows the curves of the literature record pair sequences for two types of literatures. Through farther analyzing, we find that: if the total number of matched literature record pairs is large enough, there are two distinct arcs which can divide the whole curve into three segments (head, body, and tail). We denote these two arcs a_1 and a_2 respectively. After the manual check we find that: most right ones are congregated in the head segment, while most wrong ones are congregated in the tail segment. The body segment is mainly the mixture of right ones and wrong ones. So it is feasible to regard the literature record pairs in the head segment as the top-*k* ones. And the problem is now transformed into how to find the head segment in the curve.

In order to detect a_1 accurately, we adopt a mathematic mean which consists of two main steps: curve fitting and curvature computation. In the first step, given a sequence of similarity values, point them in a two-dimensional reference frame, where *y* axes is the similarity value and *x* axes is the *i*-th similarity value. The least squares fitting method is used to fit these similarity value. And then a smoothing curve is got, just as Fig. 2 shows. The least squares fitting method is a very popular mathematic method of fitting data, and so it is not discussed here. In the second step, compute the curvature for each similarity value in the curve, and find the similarity value with the maximum curvature in the curve and the direction is downward. Then this similarity value in the curve is what we want to locate.

Now we describe the main idea on automatically achieving training samples from literature records. This process consists of three steps:

Step 1. Given two sets of literature records from different digital libraries in response to a same query, compute the tf-idf similarity for any two literature records from the two sets respectively.

Step 2. Rank the literature record pairs according to their similarities from high to low, and then process them in turn as follows: first initialize a queue and put the first literature record pair into it; then for the current literature record pair, if both of them haven't appeared in the queue, it is put into the queue.

Step 3. Locate the *k*-th similarity value with the method has been introduced above. Then the top-*k* literature record pairs are output as the training set.

Thus, a training set is obtained automatically. But it is noisy, which means there is a small part of wrong matched data record pairs. In previous works, the training sets are all provided manually, so the accuracy of them is perfect. If their experiments are based on the noisy training set, the accuracy will be far away from what they reported in their experiments. So in Section 5 we will propose a smart attribute-mapping method based on noisy training set, which can build attribute correspondences among multiple schemas of digital libraries accurately.

5. Attribute mapping

Attribute mappings building is actually the issue of schema matching. Though the training set only provides some instances, instance-level data can give important insight into the contents and meaning of schema elements. So we propose a simple instance-based way to implement attribute mappings building.

Above, we have proposed an automatic approach to obtaining the training set. Using the training set, we implement attribute correspondence building in three steps: first, identify the type of attributes; second, build the attribute mappings (mapping scheme) for each instance in the training set, and merge all mapping schemes into the mapping scheme between two digital libraries with a simple voting technique; third, accomplish the attribute mappings building among multiple digital libraries.

5.1. Data types of attributes

As our best knowledge, text-based similarity metrics are primary means to compute the similarity of two attribute values. Actually, it is unreasonable to regard all attributes as the text type. For instance, the literature records often contain page number attribute. Obviously, the page number should be treated as a numeric value instead of a text value.

So instead of regarding all attributes as text type entirely, three data types are defined: Text type, Numeric type, and Date&time type. For Text type, we use the tf-idf function. Date&time type can be regarded as a special Numeric type, and we use the following formula as the similarity metric for both Numeric type and Date&time type:

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

This formula can represent the similarity of two numeric attributes *a* and *b*. In practical: when *a* is equal to *b*, the similarity of them is 1, or the similarity will drop greatly as the difference of them becomes larger. For example, if two products are very close on the attribute "price", they are more likely to be the same one. Based on these data types and their corresponding similarity metrics, the similarity of any two attribute values can be computed as following: the similarity is 0 if they are different data types; otherwise the similarity is computed by the corresponding similarity metric.

5.2. Instance-based attribute mappings

This step aims at the attribute mapping problem: finding the semantic correspondences among the attributes of a large set of digital libraries. Essentially, this is the problem of schema matching. Some annotation works may improve the accuracy of duplicate detection somewhat. But this is a time-consuming and error-prune process, and there is still not an available tool currently. Considering for such situation, we propose a smart and simple instance-based approach to accomplish attribute mapping using the training set automatically.

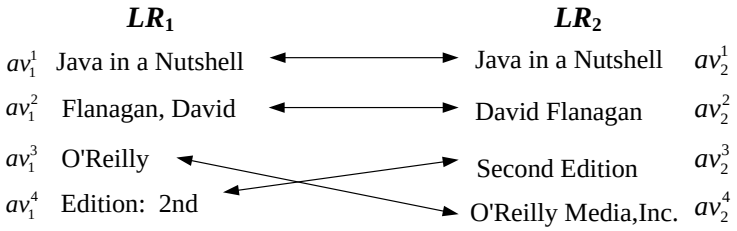

Fig. 3. Attribute mapping between two book records

Each Literature Record (LR) can be regarded as a set of *n* attribute values, and given a LR_1, denoted by $\{av_1^1, av_1^2, \ldots, av_1^n\}$ (av is the abbreviation of **a**ttribute **v**alue). We have classified attributes into three data types, Text type, Number type, and Date&time type, and further present the similarity metrics for them. So given any two attribute values, av_1^i and av_2^j, their similarity is computed with the corresponding metric. Based on this rule, we can easily build the attribute correspondences with a given instance (two matched book records) in the training set. Fig. 3 illustrates the process.

Further, *n* mapping schemes are generated for *n* matched literature record pairs in the training set. In order to assure that attributes can be mapped correctly, we should eliminate the minor wrong samples in the training set. This problem has not been encountered in previous works. We use the voting technique to try to avoid the problem produced by the noisy training set. Our basic idea is to combine these *n*

mapping schemes into the final mapping scheme obeying the rule of subordination of the minority to the majority for each attribute mapping. This technique has been claimed successful in [12] and is focused on addressing the robustness problem of interface integration caused by noisy data quality of interface schema extraction.

Obviously, it is unwise to repeat the process of building attribute mappings C_n^2 times for n Digital Libraries (DL). In this paper we use "bridge" strategy to avoid the C_n^2 cost. The main idea of attribute mapping building among multiple digital libraries consists of two phases and is described as following:

Phase 1. All digital libraries (suppose n) are ranked according to the quantity of their attributes from high to low, suppose the rank result is $\{DL_1, DL_2, \ldots, DL_n\}$, and then accomplish attribute mapping building between DL_1 and DL_2.

Phase 2. Suppose attribute mappings have been built among m digital libraries. Consider for DL_{m+1}, the attribute mappings are built between DL_1 and DL_{m+1} first. Using "bridge" strategy, the attributes (not all) of DL_{m+1} and DL_2 can be mapped by the "bridge" DL_1. For the left unmapped attributes of DL_{m+1} and DL_2, they are mapped by the similarity metrics of attribute value. Then the attribute mappings are built between DL_3 and DL_{m+1} using by the "bridge" DL_1 and DL_2. The process is repeated until the attribute mappings are built between DL_{m+1} and DL_m. Similarly, the left digital libraries are processed.

6. Attribute weight learning

The attribute mappings have been built among all digital libraries. In this part we discuss how to assign appropriate weights for these attribute mappings with an iterative training way. The process is shown in Fig. 4.

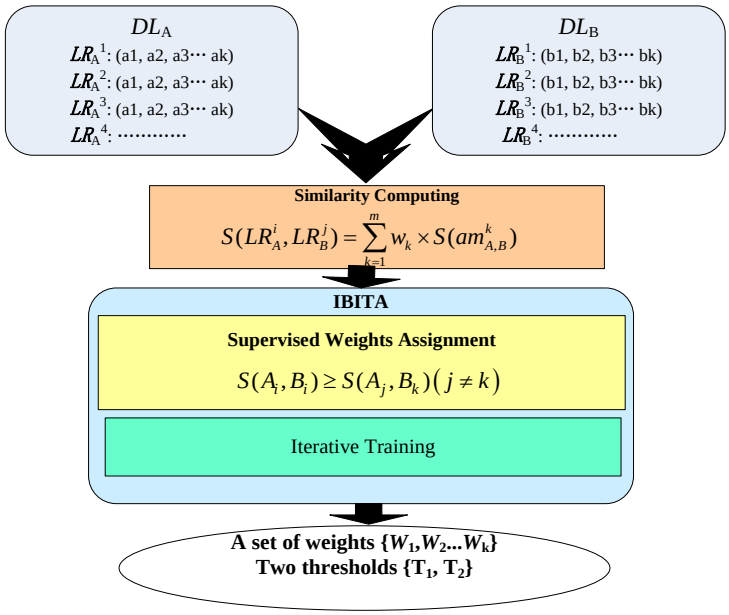

Fig. 4. General inequalities based iterative training approach architecture

The similarity of two literature records is expressed as a weighted sum of the similarities of the attribute mappings. Initially, a solution space of the weights is produced from an inequalities group. Then an iterative training approach is weights. Upon that, two thresholds will simultaneously be obtained to help accurately capture the matching relationships of two literature records.

The process consists of three stages. First, assign weights through supervised training. Second, analyze different similarity distributions caused by different weights in order to find the optimum one from the solution space. Third, extend the current weights to digital library dependent weights.

6.1. Supervised learning weight

How to assign the optimum weights for the attribute mappings of two digital libraries is the core technique in our paper. An iterative training mechanism is used to implement this technique; we call it IBITA (Inequalities Based Iterative Training Approach)

IBITA starts with two literature record sets from DL_A and DL_B. We suppose, without loss of generality, that DL_A and DL_B have m attribute mappings. For each literature record pair $\langle LR_A^i, LR_B^j \rangle$ from DL_A and DL_B, we define the similarity measurement as follows.

Definition 1 (literature record similarity). The similarity of LR_A^i and LR_B^j (in the form of two literature records with m attribute mappings) is equal to the weighted sum of the similarities based on the attribute Mappings Set $MS_{A,B}$. Correspondingly, weight w_k ($1 \leqslant k \leqslant m$) is assigned to the corresponding **a**ttribute **m**apping $am_{A,B}^k$ to show its contribution to the similarity measurement of LR_A^i and LR_B^j.

$$S(\mathrm{LR}_A^i, \mathrm{LR}_B^j) = \sum_{k=1}^{m} w_k \times S(\mathrm{am}_{A,B}^k). \tag{1}$$

The similarity for $\mathrm{am}_{A,B}^k$ is compared individually with the corresponding metrics. Using Weight Vector (WV) $\{w_1, w_2, \ldots, w_m\}$, we can measure the similarity of any given literature record pair $\langle \mathrm{LR}_A^i, \mathrm{LR}_B^j \rangle$ as a real number larger than 0. The most ideal weights vector is hoped to make all the matched literature record pairs and non-matched literature record pairs take on a distinct bipolar distribution, when projecting their similarities on the axis as shown in Fig. 5. The bipolar distribution requires all those matched literature record pairs $\langle \mathrm{LR}_A^i, \mathrm{LR}_B^j \rangle$ (represented by circles), to be located at the starboard of the axis, while all those non-matched literature record pairs $\langle \mathrm{LR}_A^i, \mathrm{LR}_B^j \rangle$, $i \neq j$ (represented by rectangles), to be located at the larboard of the axis. We aim an optimal weight vector ($WV_{optimal}$) which makes the bipolar distribution on the axis most distinct, that is, to bring the largest distance of matched and non-matched literature record pairs marked on Fig. 5. Meanwhile, two thresholds are also needed to classify each literature record pair $\langle \mathrm{LR}_A^i, \mathrm{LR}_B^j \rangle$ as "matched", "non-matched" or "possibly matched". The solution will be based on training set in the form of literature record pairs.

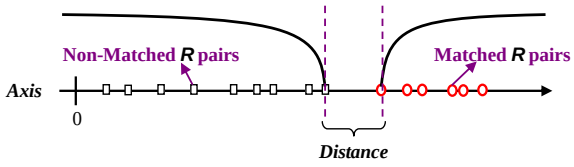

Fig. 5. The ideal bipolar distribution

For two literature record sets from DL_A and DL_B, we have automatically achieved n pairs of matched literature record pairs to form the training data set, where each pair describes the same entity. Suppose LR_A^1, LR_A^2, …, LR_A^n denote the n *Literature records* coming from DL_A that are used for training and LR_B^1, LR_B^2, …, LR_B^n denote the n corresponding *Literature records* coming from DL_B. Each LR pair $\langle LR_A^i, LR_B^j \rangle$ is a matched pair representing the same entity. Meanwhile each LR pair $\langle LR_A^i, LR_B^j \rangle$, $j \neq i$, is a non-matched pair representing two different entities.

6.2. Inequalities-based metrics

By observing the axis in Fig. 5 which shows the bipolar distribution, we find that the weight vector $\{w_1, w_2, \ldots, w_m\}$ needs to be adjusted to satisfy the following condition in the first place: The similarity of a matched pair is greater than the similarity of a non-matched pair. Formally said, the similarity of n uniquely matched pair $\langle LR_A^i, LR_B^j \rangle$ should be greater than any of the $n(n-1)$ non-matched pairs $\langle LR_A^i, LR_B^j \rangle$, $j \neq k$. Therefore, a group of $n(n-1)$ inequalities can be correspondingly obtained as follows:

(2) $$\{S(LR_A^i, LR_B^i) \geq S(LR_A^j, LR_B^k),\ 1 \leq i,\ j,\ k \leq n,\ j \neq k\}.$$

Then for all the n training samples from DL_A, a total of $n(n-1)$ inequalities will be obtained. And our aim is to find $WV_{optimal}$ from the solution space of Inequalities (2). Intuitively, we have to solve these $n(n-1)$ inequalities, a right (not optimal) WV can be the output. In practice, the exponential growth of the number of inequalities is too costly. So we use the following subset of these inequalities instead of all:

(3) $$\{S(LR_A^i, LR_B^i) > S(LR_A^i, LR_B^j),\ 1 \leq i,\ j \leq n,\ i \neq j\}.$$

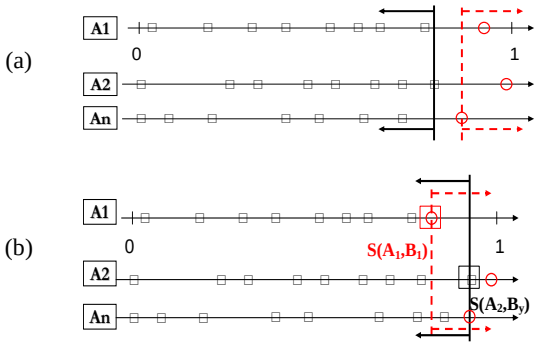

Fig. 6. Ideal situation and cross-region situation

For any WV in the solution space of Inequalities (3), there will be two possibilities: that is, the WV satisfies Inequalities (2), or not. In another word, not all the WVs of Inequalities (3) can make the n matched literature record pairs and $n(n-1)$ non-matched literature record pairs a bipolar distribution as we wanted (see Fig. 6a). Some WVs may lead to the cross-region situation shown in Fig. 6b, where it is still guaranteed on each axis, the matched pair (denoted by small circle) is closer to the starboard than all the non-matched pairs (denoted by small rectangle). The cross-region situation means not all the similarities of n matched pairs are larger than the similarities of all $n(n-1)$ non-matched pairs. This cross-region situation is thus caused where the n literature record pairs in training data set cannot be divided into two categories: matched or non-matched. We can see from Fig. 6b that, the similarity of the non-matched pair $\langle LR_A^2, LR_B^y \rangle$, $y \neq 2$, is larger than the similarity of the matched pair $\langle LR_A^1, LR_B^1 \rangle$. The confusion caused by this situation can be described as follows: if the similarity of the new literature record pair falls into the cross-region formed by T_1 and T_2, the system will not be able to judge whether this two literature records represent the same entity due to the ambiguity they have. So, we need to try and achieve a WV in the solution space of Inequalities (2) based on Inequalities (3).

6.3. Iterative training

Given a WV$\{w_1, w_2, \ldots, w_m\}$ in the solution space of Inequalities (3), the similarity of any $\langle LR_A^i, LR_B^j \rangle$, $1 \leqslant i, j \leqslant n$, in the n training samples can be derived as the weighted sum of the similarities of all attribute mappings. Then in total, for all the n literature records from DL_A and n literature records from DL_B in the training set, there are $n \times n$ similarities being computed, each of which corresponds to one possible combination of LR_A^i, $1 \leqslant i \leqslant n$, and LR_B^j, $1 \leqslant j \leqslant n$. We project these n^2 similarities to n axes and try to iteratively analyze different similarity distributions on the axes caused by different WVs in order to find $WV_{optimal}$.

For each LR_A^i, $1 \leqslant i \leqslant n$, we build an axis, and n similarities are projected on the axis as shown in Fig. 6. The similarities of LR_A^i with all n literature records from DL_B are located on the i-th axis. The circle denotes matched LR pair $\langle LR_A^i, LR_B^j \rangle$ which are closest to the starboard of the axes, while the small rectangles denote non-matched LR pairs $\langle LR_A^i, LR_B^j \rangle$, $i \neq j$.

Given a WV, the minimum similarity of all n matched pairs is regarded as a threshold T_1 (dashed line in Fig. 6) and the maximum similarity of all $n(n-1)$ non-matched pairs is regarded as a threshold T_2 (real line in Fig. 6). Formally, we denote them as the following form:

$$T_1 = \min\{S(LR_A^i, LR_B^i)_{WV}\},$$

$$T_2 = \max\{S(LR_A^i, LR_B^j)_{WV}\}, \; i \neq j,$$

where $S(LR_A^i, LR_B^j)_{WV}$ is the similarity of LR_A^i and LR_B^j being computed with WV.

If $T_2<T_1$, we can assure the similarity of any literature record pair $\langle \mathrm{LR}_A^i, \mathrm{LR}_B^j \rangle$ is larger than the similarity of any LR pair $\langle \mathrm{LR}_A^i, \mathrm{LR}_B^j \rangle$, $i \neq j$. So, the ideal situation can be judged by $T_2<T_1$, and cross-region situation can be judged by $T_1<T_2$.

There are two main steps in the implement of this component which tries to achieve $\mathrm{WV}_{\mathrm{optimal}}$ staring at any WV in the solution space of Inequalities (3). The first step is achieving a WV satisfying Inequalities (2) from the WV of Inequalities (3), and the second step is achieving $\mathrm{WV}_{\mathrm{optimal}}$ from a WV of Inequalities (2).

Step 1. Computing WV in Inequalities 2

This process starts at Inequalities 3. At the beginning, a WV is got by solving Inequalities (3), and further T_1 and T_2 are got. If $T_2<T_1$, this WV satisfies Inequalities (2), and the next process is activated. Otherwise, the WV caused the cross-region situation, and this is what this process aims at. Next, for $T_1<T_2$, it is represented in this form:

$$\min\{S(\mathrm{LR}_A^i, \mathrm{LR}_B^j)_{\mathrm{WV}}\} < \max\{S(\mathrm{LR}_A^i, \mathrm{LR}_B^j)_{\mathrm{WV}}\},\ i \neq j. \tag{4}$$

In the next Step 2 Inequalities (5) is formed by appending Inequality (4) to Inequalities (3), and WV′ is got by solving Inequalities (5). The left of this step is repeating the above process until the WV can satisfy Inequalities (2).

The main idea of this step is to iteratively append the inequalities which do not satisfy Inequalities (2) to Inequalities (3) until a WV which satisfies Inequalities (2) is got. In another word, the solution space continues shrinking during the process and a WV in the solution space of Inequalities (3) has more probability to be in the solution space of Inequalities (2). Actually, there is more than one inequality which does not satisfy Inequalities (2), but only one inequality (Inequality (4)) is appended at every iteration, due to the consideration of efficiency improvement. In practical, the iteration is less than two times on an average.

Step 2. Computing $\mathrm{WV}_{\mathrm{optimal}}$

This process starts at a WV of Inequalities (2). The current WV can make the similarity of any matched literature record pair larger than that of any non-matched LR pair of the training set. In order to reach high accuracy, we need to achieve $\mathrm{WV}_{\mathrm{optimal}}$ which can make the matched literature record pairs and non-matched literature record pairs the most distinct bipolar distribution. In another word, $\mathrm{WV}_{\mathrm{optimal}}$ can make the distance of T_1 and T_2 (i.e., $T_1 - T_2$) reach the maximum.

In order to make the description concisely and without confusion, we use Inequality (4) to denote all the inequalities appended to Inequalities (3). Suppose Inequalities (5) is Inequalities (3) and the inequalities appended to Inequalities (3) in the first step. So Inequalities (5) is denoted as

$$\begin{cases} \{S(\mathrm{LR}_A^i, \mathrm{LR}_B^i) - S(\mathrm{LR}_A^i, \mathrm{LR}_B^i) \geq 0,\ 1 \leq i,\ j \leq n,\ j \neq i\}, \\ \{\max\{S(\mathrm{LR}_A^i, \mathrm{LR}_B^j)_{\mathrm{WV}} - \min\{S(\mathrm{LR}_A^i, \mathrm{LR}_B^i)_{\mathrm{WV}} > 0, \quad i \neq j\}, \\ \cdots \end{cases} \tag{5}$$

Initially, the "0"s in the right side of inequalities is replaced by $T_1 - T_2$, and the new inequalities (Inequalities (6)) are denoted as the following:

$$(6)\quad \begin{cases} \{S(\mathrm{LR}_A^i, \mathrm{LR}_B^i) - S(\mathrm{LR}_A^i, \mathrm{LR}_B^i) \geq T_1 - T_2, \ \ 1 \leq i, j \leq n, \ \ j \neq i\}, \\ \{\max\{S(\mathrm{LR}_A^i, \mathrm{LR}_B^j)_{\mathrm{WV}} - \min\{S(\mathrm{LR}_A^i, \mathrm{LR}_B^i)_{\mathrm{WV}} > T_1 - T_2, \quad i \neq j\}. \end{cases}$$

WV′ is got by solving Inequalities (6), and further T'_1 and T'_2 are got. Then $T'_1 - T'_2$ replace $T_1 - T_2$ in Inequalities (6), and the above process is repeated until $(T'_1 - T'_2) - (T_1 - T_2) < \sigma$; σ is set in advance, and the smaller σ is the current WV closer to $\mathrm{WV}_{\mathrm{optimal}}$.

Till now, when applying the similarity measurement to any two web data sources DL_A and DL_B for duplicate detection, IBITA can ultimately bring an optimum group of quantified weights and two stabilized thresholds. Then it is easy to compute the similarity for any literature record from DL_A and any literature record from DL_B based on the derived weights. By comparing the similarity value with the two thresholds, we can easily determine whether they are matched. If the similarity of the literature record pair falls into the possibly matched region, it needs to be manually checked.

6.4. Weights merging

For any two digital libraries, we can determine the weights of the attributes they share. As we have discussed above, each attribute in a given domain plays a definite role on the problem of duplicate detection (independent to any special digital library). So, we try to determine the weights of attributes under the context of domain by training representative digital libraries.

The key is how to determine the weights of attributes according to a group of WVs received from special literature record pairs. Given *n* WVs achieved from special digital library pairs, we use the means of them to represent the weights of attributes of domain. Actually, the weights refer to the importance of attributes, and we only care about the proportions among the weights in one WV. For example, ⟨1, 2⟩ and ⟨2, 4⟩ are the same. So it is unreasonable to compute the means of WVs directly. According to our observation, though *digital libraries* often cover different attributes of domain, all of them always share at least one attribute. For instance, the attribute "title" appears at all *digital libraries* in book domain. We select one attribute (suppose *a*) which is shared by all *digital libraries* and set its weight to be 1. Then consider each group weights (WV), the weights except the weight of *a* are adjusted correspondingly by proportion. At last, all adjusted WVs are merged into one WV by computing their mean. The whole process is illustrated on Fig. 7.

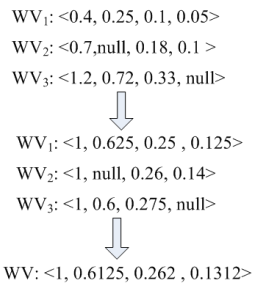

Fig. 7. Illustration of weights merging

7. Experiments

7.1. Test bed

The purpose of this work is to evaluate our holistic duplicate detection algorithms under the context of large scale cross-library search. We select 10 most popular university digital libraries which can search paper literatures and book literatures. The reason these digital libraries are selected is not only because they are popular, but also because each of them contains large number of literature records, which may cause large proportion of duplicates entities. Five pairs of digital libraries are selected from total C_n^2 pairs of digital libraries by ranking on the quantity of shared attributes. Then the weights were hoped to generate from the selected five pairs of digital libraries. For each pair of digital libraries, we submit six queries. Both the training set and testing set are coming from the returned query results. The weights for different attributes are obtained from the training data set, which is automatically achieved from half of the returned query results. The testing set comes from the other half of returned query results, which is used to evaluate the accuracy of generated weights. For each query, we would select 100 returned literature records.

Table 1. Attribute weights of paper literature

Pair	Title	Author	Publisher	P. Date	ISSN	Page
P1	0.97	0.53	0.29	_	1.0	0.52
P2	0.86	0.43	0.35	0.56	_	_
P3	0.91	0.49	0.22	0.70	_	_
P4	0.92	0.55	0.36	0.81	_	0.58
P5	0.95	0.47	0.33	0.67	_	_
AVG	0.92	0.48	0.31	0.68	1.0	0.55

Table 2. Attribute weights of book literature

Pair	Title	Author	Publisher	P. Date	Page	ISBN	Edition
P_1	0.62	0.54	0.46	0.58	0.52	_	_
P_2	0.76	_	_	_	_	1.0	_
P_3	0.72	_	_	_	_	1.0	_
P_4	0.79	0.72	0.61	_	0.58	_	0.81
P_5	0.81	0.61	_	0.46	0.66	_	0.71
AVG	0.74	0.62	0.53	0.52	0.59	1.0	0.76

The characteristics of the data set can be concluded as follows: (1) for each query, the returned results from 2 paired digital libraries shared a large proportion of overlapping entities; (2) the scale of our data set is quite large that the amount of literature records has achieved 1000 for each pair of digital libraries; (3) the submitted queries are as independent as possible, which guarantees that no overlap will exists between different query results of different queries. All those features of our data set ensure the objectivity of our experimental results.

7.2. Evaluation criteria

A popular measure for evaluating the effectiveness of duplicate detection approach is adopted. Four criteria are defined as follows.

$$\text{Precision}M = \frac{|\text{Predicted MP} \bigcap \text{Actual MP}|}{|\text{Predicted MP}|},$$

$$\text{Precision}N = \frac{|\text{Predicted NP} \bigcap \text{Actual NP}|}{|\text{Predicted NP}|},$$

$$\text{Uncertainty} = \frac{|\text{Uncertain P}|}{|\text{Predicted MP} + \text{Predicted NP} + \text{Uncertain P}|},$$

$$\text{Precision}T = \frac{|\text{Predicted MP} \bigcap \text{Actual MP}| + |\text{Predicted NP} \bigcap \text{Actual NP}|}{|\text{Predicted MP} + \text{Predicted NP} + \text{Uncertain P}|},$$

where Actual MP is the set of real matched LR pairs in the testing set and Predicated MP is the set of matched literature record pairs discovered by our method. Similarly, Actual NP is the set of real non-matched record pairs in the testing set and Predicated NP is the set of non-matched record pairs discovered by us. In addition, Uncertain P denotes the set of data records that cannot be assigned to one of the two classes (matched/non-matched) for sure. Those uncertain pairs need to be further manually checked.

7.3. Results and analysis

For both *Paper* literatures and *Book* literatures, we calculate for each of the five selected digital libraries pairs a group of optimal weights. Generally speaking, the selected top five pairs of digital libraries have covered all of the domain attributes basically. Then the weights group for the domain can be formed by summarizing these entire five member weights group. Table 1 and Table 2 shows the generated weights group for *Paper* literature and *Book* literature respectively.

Table 3. Accuracy analysis for paper literatures

Pair	*M*	Precision*M*	*N*	Precision*N*	*U*	Uncertainty	*T*	Precision*T*
P1	128	0.858	111	0.997	7	0.026	246	0.914
PD1	121	0.811	106	0.952	9	0.033	236	0.882
P2	101	0.954	113	0.895	3	0.012	217	0.923
PD2	105	1	109	0.863	7	0.029	221	0.940
P3	147	0.876	111	0.813	0	0	258	0.848
PD3	140	0.834	106	0.776	6	0.019	252	0.828
P4	135	0.819	100	0.848	5	0.017	240	0.836
PD4	140	0.849	113	0.958	4	0.013	257	0.893
P5	108	0.943	147	0.919	2	0.007	257	0.931
PD5	112	0.978	152	0.950	4	0.014	268	0.969

Table 4. Accuracy analysis for book literatures

Pair	*M*	Precision*M*	*N*	Precision*N*	*U*	Uncertainty	*T*	Precision*T*
P1	95	0.989	91	1	4	0.020	190	0.994
PD1	85	0.885	87	0.956	7	0.036	179	0.937
P2	138	0.965	58	0.983	2	0.009	198	0.970
PD2	131	0.916	59	1	5	0.024	195	0.955
P3	120	0.96	73	0.986	5	0.024	198	0.970
PD3	116	0.928	67	0.905	9	0.044	192	0.941
P4	88	0.977	111	0.991	2	0.009	201	0.985
PD4	81	0.9	106	0.946	3	0.014	190	0.931
P5	97	0.979	73	0.948	4	0.022	174	0.966
PD5	91	0.919	75	0.974	4	0.022	170	0.944

In particular, we find in Table 1 that digital libraries from pair 2(P2) and pair 3(P3) share ISSN attribute and ISBN itself is enough to judge the entity matching. Therefore, weight 1.00 is assigned to ISBN, and other attributes need not to be considered.

For each selected pair, we apply both the weights generated from its own training set and the summarized domain weights to its testing set (e.g., PD1 means applying domain weights to pair 1). Table 3 and Table 4 show the accuracy of our method for *Paper* literatures and *Book* literatures respectively. Here, we use *M* to denote how many PredicatedMP are ActualMP, *N* to denote how many PredicatedNP are ActualNP, *U* to denote how many pairs are UncertainP and T to denote the sum of *M*, *N* and *U*. As we can see, our experimental results have 3 features: (1) the high precisions on four criteria show that our algorithm is highly effective; (2) the amount of uncertain pairs is relatively small and most *Uncertainty* are lower than 3%, which is a great reduce of the manual intervention; (3) the precision achieved from different pair of digital libraries from different domain won't vary from each other intensively. Overall, the experimental results show that our web duplicate detection approach is highly effective.

In addition, we found in our experiments that the achieved precision is not positive linear correlated with the scale of training data set. We find that the best performance could be reached when there are about 20 pairs of data records selected as training set for two digital libraries. Excessive training samples would bring some noisy inequalities, while inadequate training samples are not enough to infer the different importance of different attributes. Our holistic solution of duplicate detection can achieve good performance with required small scale of training data set.

8. Conclusions and future works

In this paper, we aim at the problem of duplicate detection for cross library search. We first give an observation to the attributes in this domain and hypothesize that their roles are definite and domain-dependent. Then, we propose a holistic approach to address this problem, which includes training set extracting, attribute mapping, and attribute weight assigning. In the experiments, we choose two representative domains

(book and computer) to evaluate our solution. The experimental results prove that the solution accuracy is satisfying in practical.

In the future we will make the improvements of this solution. First, it is not suitable to digital libraries with a small duplicate overlap yet. Second, the efficiency may not be satisfying though the whole approach is fully automatic and overmatches the previous works. Third, many technique details still have not been solved in theory. And further, the experiment on other types of literatures will be done.

***Acknowledgements*:** This work was partially sponsored by National Science Foundation of China "On Construction of Scholarly Relations Network Based on Massive Digital Resources" under Grant No 71273251.

References

1. **http://www.brightplanet.com/completeplanet/**
2. Su, W., H. Wu, Y. Li et al. Understanding Query Interfaces by Statistical Parsing. – ACM Transactions on the Web (TWEB), Vol. **7**, 2013, No 2, p. 8.
3. Dragut, E. C., W. Meng, C. T. Yu. Deep Web Query Interface Understanding and Integration. – Synthesis Lectures on Data Management, Vol. **7**, 2012, No 1, pp. 1-168.
4. Lu, Y, H. He, H. Zhao et al. Annotating Search Results from Web Databases. – Knowledge and Data Engineering, IEEE Transactions on, Vol. **25**, 2013, No 3, pp. 514-527.
5. Palekar, V. R., M. S. Ali, R. Meghe. Deep Web Data Extraction Using Web Programming-Language Independent Approach. – Journal of Data Mining and Knowledge Discovery, Vol. **3**, 2012, No 2, p. 69.
6. Wang, Z., G. Xu, H. Li et al. A Probabilistic Approach to String Transformation. – Knowledge and Data Engineering, IEEE Transactions on, Vol. **26**, 2014, No 5, pp. 1063-1075.
7. Sood, S., D. Loguinov. Probabilistic Near-Duplicate Detection Using Simhash. – In Proc of 20th ACM International Conference on Information and Knowledge Management, ACM, 2011, pp. 1117-1126.
8. Zhao, W. L., C. W. Ngo, H. K. Tan et al. Near-Duplicate Keyframe Identification with Interest Point Matching And Pattern Learning. – Multimedia, IEEE Transactions on, Vol. **9**, 2007, No 5, pp. 1037-1048.
9. Hajishirzi, H., W. Yih, A. Kolcz. Adaptive Near-Duplicate Detection via Similarity Learning. – In: Proc. of 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2010, pp. 419-426.
10. Zhao, P., J. Xin, X. Xian et al. Active Learning for Duplicate Record Identification in Deep Web. Foundations of Intelligent Systems. Berlin, Heidelberg, Springer, 2014, pp. 125-134.
11. Xiao, C., W. Wang, X. Lin et al. Efficient Similarity Joins for Near-Duplicate Detection. – ACM Transactions on Database Systems (TODS), Vol. **36**, 2011, No 3, p. 15.
12. He, B., K. C.-C. Chang. Making Holistic Schema Matching Robust: An Ensemble Approach. – KDD, 2005, pp. 429-438
13. Fellegi, I. P., A. B. Sunter. A Theory for Record Linkage. – Journal of the American Statistical Association, Vol. **64**, December 1969, No 328, pp. 1183-1210.
14. Newcombe, H. B., J. M. Kennedy, S. J. Axford, A. P. James. Automatic Linkage of Vital Records. – Science, Vol. **130**, October 1959, No 3381, pp. 954-959.
15. Jaro, M. A. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. – Journal of the American Statistical Association, Vol. **84**, June 1989, No 406, pp. 414-420.
16. Dempster, A., N. Laird, D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. – Journal of the Royal Statistical Society, Vol. **B**, 1977, No 39, pp. 1-38.

17. W i n k l e r, W. E. Improved Decision Rules in the Felligi-Sunter Model of Record Linkage. Technical Report Statistical Research Report Series RR93/12, U.S. Bureau of the Census, Washington, D.C., 1993.
18. C o c h i n w a l a, M., V. K u r i e n et al. Improving Generalization with Active Learning. – Information Sciences, Vol. **137**, September 2001, No 1-4, pp. 1-15.
19. B r e i m a n, L., J. F r i e d m a n et al. Classification and Regression Trees. CRC Press, July 1984.
20. H a s t i e, T., R. T i b s h i r a n i, J. F r i e d m a n. The Elements of Statistical Learning. – Springer Verlag, August 2001.
21. B i l e n k o, M., R. M o o n e y et al. Adaptive Name Matching in Information Integration. – IEEE Intelligent Systems, Vol. **18**, 2003, No 5, pp. 16-23.
22. C h a n g, K. C., B. H e, C. L i, M. P a t e l, Z. Z h a n g. Structured Databases on the Web: Observations and Implications. – SIGMOD Record, Vol. **33**, 2004, No 3, pp. 61-70.
23. C o h e n, W., J. R i c h m a n. Learning to Match and Cluster Large High-Dimensional Data Sets for Data Integration. – In Proc. of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.
24. M c C a l l u m, A., B. W e l l n e r. Conditional Models of Identity Uncertainty with Application to Noun Coreference. – In: Proc. of Advances in Neural Information Processing Systems (NIPS'2004), 2004.
25. X i a o, C., W. W a n g, X. L i n et al. Efficient Similarity Joins for Near-Duplicate Detection. – ACM Transactions on Database Systems (TODS), Vol. **36**, 2011, No 3, p. 15.
26. T e j a d a, S., C. K n o b l o c k, S. M i n t o n. Learning Domain-Independent String Transformation Weights for High Accuracy Object Identification. – In: Proc. of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.
27. R o h i t, A., S. C h a u d h u r i, V. G a n t i. Eliminating Fuzzy Duplicates in Data Warehouses. – In: Proc. of 28th International Conference on Very Large Databases, 2002.
28. G u h a, S., N. K o u d a s et al. Merging the Results of Approximate Match Operations. – In: Proc. of 30th International Conference on Very Large Databases, 2004, pp. 636-647.
29. C h a u d h u r i, S., V. G a n t i, R. M o t w a n i. Robust Identification of Fuzzy Duplicates. – In: Proc. of 21st IEEE International Conference on Data Engineering (ICDE'2005), 2005, pp. 865-876.
30. C h r i s t e n, P. A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication. – IEEE Transactions on Knowledge and Data Engineering, Vol. **24**, 2012, No 9, pp. 1537-1555.