

PAPER

A Similarity-Based Concepts Mapping Method between Ontologies

Jie LIU^{†a)}, Member, Linlin QIN[†], Jing GAO^{††}, and Aidong ZHANG^{††}, Nonmembers

SUMMARY Ontology mapping is important in many areas, such as information integration, semantic web and knowledge management. Thus the effectiveness of ontology mapping needs to be further studied. This paper puts forward a mapping method between different ontology concepts in the same field. Firstly, the algorithms of calculating four individual similarities (the similarities of concept name, property, instance and structure) between two concepts are proposed. The algorithm features of four individual similarities are as follows: a new WordNet-based method is used to compute semantic similarity between concept names; property similarity algorithm is used to form property similarity matrix between concepts, then the matrix will be processed into a numerical similarity; a new vector space model algorithm is proposed to compute the individual similarity of instance; structure parameters are added to structure similarity calculation, structure parameters include the number of properties, instances, sub-concepts, and the hierarchy depth of two concepts. Then similarity of each of ontology concept pairs is represented by a vector. Finally, Support Vector Machine (SVM) is used to accomplish mapping discovery by training and learning the similarity vectors. In this algorithm, *Harmony* and *reliability* are used as the weights of the four individual similarities, which increases the accuracy and reliability of the algorithm. Experiments achieve good results and the results show that the proposed method outperforms many other methods of similarity-based algorithms.

key words: ontology mapping, similarity aggregation, SVM

1. Introduction

In the research of information integration, semantic web and knowledge management, ontology has been used widely. With the growth of ontological applications, researchers have constructed various heterogeneous ontologies. Although these different ontologies are in the same domain, they don't have identical linguistics or structure because of the complexity and flexibility of the natural language. So the diverse ontologies cannot be directly integrated into an unified ontology. Ontology mapping is a major way to solve this problem.

In general, the main work of ontology mapping is concept mapping. The similarity calculation is the main method of concept mapping. Concept mapping is achieved by merging individual similarities between the same features of two ontology concepts. The features of a concept include concept name, properties, instances, structure, etc., so the individual similarities include concept name similarity, property

similarity, instance similarity, structure similarity and so on. After merging the sub-similarities, we can form the similarity between concepts. In this paper, ontology mapping discovery means finding the mapped concept pairs.

In existing methods of similarity-based ontology mapping, the weights of individual similarities are decided by experts or by simple linear functions, and both methods are too subjective. Or the weights are determined by the importance index of individual similarities such as *harmony* or *reliability*, but this would excessively bias the results of the individual similarities. Besides, the methods of machine learning are adopted to get the weights or to achieve directly mapping discovery. There are still many drawbacks for the methods mentioned above. Therefore, it is necessary to study a more effective algorithm of ontology mapping.

In this paper, our study is mainly to discover the mappings between concepts belonging to the different ontologies, contributions include two points as follows:

1. Using new methods of calculating individual similarities.

A new WordNet-based method is used to compute semantic similarity between concept names and the edit distance is used to calculate grammar similarity between concept names. As to the similarity algorithm between properties, firstly calculate similarity value of each pair of properties, form property similarity matrix between concepts, then process the matrix into a numerical similarity, whose value is just the property similarity between concepts. This paper introduces a new vector space model algorithm to compute the similarity between instances, and identify the words which are literally different but very similar semantically, consuming no more time. In order to more clearly reflect the structure of concept, structure parameters are added in similarity calculation, which include the number of property, instance, sub-concept and the hierarchy depth of two concepts respectively.

2. Proposing the method of individual-similarities aggregation and using SVM to classify the similarity vector which reflects the similarity of concept pairs. Here, the elements of a similarity vector consist of the weighted individual similarities, and the weight of an individual similarity is the linear function of *harmony* and *reliability* (more detail in Sect. 3.2). The *harmony* and *reliability* enable the mapping effect clearly.

To evaluate the method proposed in this paper, we used the benchmark tests for OAEI ontology matching campaign as data sets, and got precision, recall and f-measure of the

Manuscript received June 9, 2014.

Manuscript revised October 24, 2014.

Manuscript publicized January 26, 2015.

[†]The authors are with College of Information and Engineering, Capital Normal University, Beijing 100048, P.R. China.

^{††}The authors are with University at Buffalo, State University of New York, USA.

a) E-mail: liujxxxxy@126.com

DOI: 10.1587/transinf.2014EDP7188

different ontology mapping algorithms. Experimental results demonstrate the effectiveness of our approach.

2. Related Work

The studies about ontology mapping mainly focus on merging individual similarities corresponding to different features of concept to achieve ontology mapping. Widely-used methods include grammar-based, instance-based, structure based, semantic-based, the definition of the concept based, machine learning based, and statistics based, and so on [1], [2]. The advantages and disadvantages of the existing similarity computation methods about precise ontology mapping are as follows:

We analyze firstly the methods of calculating individual similarities. In the relevant studies about concept name similarity, syntax similarities between the concepts were calculated by Jaccard coefficient [3]–[6] and edit distance [7], [8], but they ignored that different words in syntax may be synonymous. P. Giuseppe et al. [9] used WordNet to compute the semantic similarities between the English concepts. Huang Li et al. [2] considered the descriptive sentence similarity of the concept name when calculating the similarity between the concept names. In the relevant studies about property similarity, in [10], [11] some properties were taken as common properties between two concepts, the ratio of the common properties to all properties is property similarity, the method didn't consider that non-identical properties may be very similar in the semantics. In the relevant studies about instance similarity, in [12], [13] the ratio of the identical instances to all instances was used as instance similarity, but it did not be considered that the non-identical instances may have the very similar senses. In the relevant studies about structural similarities, the methods in [3], [14]–[16] took the weighted sum of similarities of super-concept, sub-concept and sibling-concept as structure similarity, but the weights were given by experts.

In the relevant studies about using similarity aggregation to achieve ontology mapping discovery, Max, Min, Weighted, Average and SIGMOD [17], have been proposed to aggregate different individual similarities. The Max/Min strategy returns the maximal/minimal similarity of individual matchers. The weighted strategy determines a weighted sum of similarity of individual matchers. The Average strategy is one special case of the weighted strategy and returns the average similarity over all individual matchers. The SIGMOID strategy combines multiple results using a sigmoid function, which is essentially a smoothed threshold function. Currently the systems that adopt the weighted strategy or the SIGMOID strategy to aggregate similarities need to manually set aggregation weights based on experience for different similarities or tentatively set center position and steepness factor in the sigmoid function. However, manually predefined parameters cannot be generalized to adapt to different mapping situations. In R. Trillo et al. [18] the sum of weighted individual similarities was seen as aggregated similarity, but the weights are given by experts. In

Q.V. Tran et al. [19] the K-means algorithm was used to seek out the concept pairs whose similarities higher than threshold, and then the ratio of the number of the mapped concept pairs to the number of all picked out concept pairs was used as weights. In [20], [21] the harmony or reliability of individual similarities was used as weights, and these two methods were more objective, but they were too dependent on the results of individual similarities to lead to not reflecting the original importance of different features of the concepts. In addition, there are many studies about using machine learning to achieve ontology mapping. In [22], [23] neural network was adopted to train the weights of individual similarities, which requires a lot of samples, and the network structure was set by the researchers subjectively, and this method may lead to not have a global optimal solution. In R. Ichise [24] SVM was applied to classify similarity vectors to achieve ontology mapping without using the weights.

At present, there has been a lot of variety of mapping systems, such as ASCO [25], OLA [26], H-MATCH [27]. These systems mostly use a variety of strategy to find the similarity between ontology elements. And a small quantity of systems use the integration of various strategies to find the appropriate mapping. The integrated approaches use the similarity of multi-strategy of weighted average, such as hybrid, composite merging methods. Although they have achieved some results, there is still insufficient. First of all, the integrated methods employed by these systems simply merge similarity value of multi-strategy. The strategy itself doesn't affect the final results of mapping at the semantic level, leading to the inferior quality of mapping. Secondly, these methods require a large amount of manual intervention, such as the weighted average method. The excessive intervention of users and experts in the field, often leads to partial mapping relationship is omitted. Moreover, these methods would be unable to effectively deal with the large-scale ontology mapping task. There are also some famous ontology mapping systems, such as GLUE [12], QOM [28], Similarity Flooding [29], PROMPT [30], Falcon-AO [31], RiMOM [13], LILY [32], Cupid [33], and ASMOV [34]. From the aggregation view, though Falcon-AO measures both linguistic comparability and structural comparability of ontologies to estimate the reliability of matched entity pairs, it only uses them to form three heuristic rules to integrate results generated by GMO [35]. ASMOV is an automated ontology mapping tool that iteratively calculates the similarity between concepts in ontologies by analyzing some features such as textual description. But ASMOV is a heuristic rule based weighted aggregation and only two constraints are validated for their final results. Comparing with R. Trillo, RiMOM calculates two similarity factors to estimate the characteristics of ontologies, which is only suitable to some special situations. For example, its linguistic similarity factor only concerns elements that have the same label.

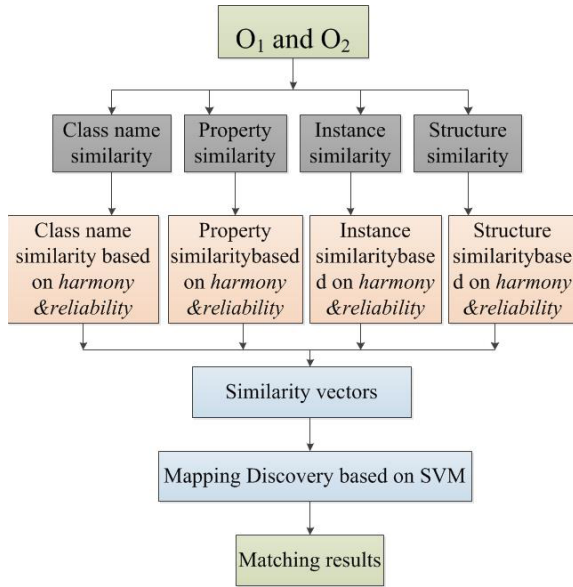


Fig. 1 Process of ontology concepts mapping.

3. The Algorithms of Ontology Mapping

Using similarity calculation, we need to select an artificial threshold to determine whether two concepts can be mapped or not. In the process of the parameter setting and the threshold value selection we require the users to participate. Furthermore, different threshold values will be taken when calculating similarity from different areas. To solve this problem, we use a combination of similarity calculation method and SVM.

The process from calculating similarities to discovering ontology mapping is shown in Fig. 1. In Fig. 1, O_1 , O_2 are two ontologies. Firstly, four individual similarities are computed. Secondly, the similarity vectors consisting of four weighted individual similarities are built, and the weights are decided by both of harmony and reliability. Here, each concept pairs between two ontologies is represented by a similarity vector. Lastly, SVM is used to accomplish mapping discovery by classifying the similarity vectors. our system is called as SHRS.

3.1 The Algorithms of Individual Similarities

3.1.1 The Algorithm of Concept Name Similarity

When two names are definitely different in syntax but synonymous, the performance of syntactic similarity calculation methods is very poor. Therefore, in this paper the semantic similarity of two concept names is firstly computed in WordNet to be taken as concept name similarity. if the similarity computed in WordNet is zero, or the concept names may not exist in WordNet, then the syntactic similarity of two concept names will be calculated by edit distance algorithms to be taken as concept name similarity.

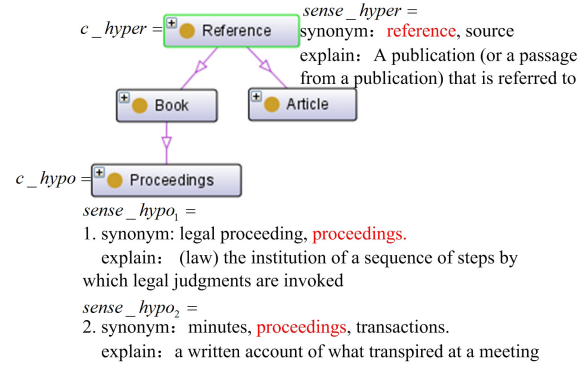


Fig. 2 The sense localization based on the hypernym concept.

Below we introduce our computing method of concept name similarity based on WordNet. A word may have multi-senses in WordNet, but in traditional WordNet similarity algorithms, the similarity between the most similar senses of two words is regarded as the word similarity, which is improper in our method, because that the true meanings of concept names in the ontological context are not considered, which results in the falling of the precision of concept name similarity calculation and ontology mapping. This paper proposes the algorithm of word sense localization based on WordNet to compute the concept name similarity. In this paper, we regard the right sense of the hypernym concept and the description of the ontology concept as two characteristics of ontology concept. For each feature, this method finds the sense which has the highest similarity with ontology concept in WordNet, and regards that sense as the right sense of ontology concept in WordNet.

Specifically speaking, 1) with the real meaning of the hypernym concept of concept c in the ontology, find the right sense of c in WordNet. For example, Fig. 2 is an ontology fragment, the top concept is “Reference”, the right sense of the concept “Proceedings” need be determined; the right sense of the top concept “Reference” is Known in advance and is *sense_hyper*, after similarity computing, the similarity value is highest between the second sense *sense_hypo2* of “Proceedings” and *sense_hyper*, so the right sense of “Proceedings” is *sense_hypo2*.

2) with the description of concept c in the ontology, through the improved latent semantic analysis (LSA), find the right sense of c in WordNet; and 3) using the right sense of c got in (1) and (2) respectively, calculate the semantic similarity between the concept names, and take the weighted sum of the two similarity as the final similarity of the concept names. This chapter puts forward two methods of sense localization of ontology concept. One is the sense localization algorithm based on the ontology hypernym - hyponym concept; and another is the sense localization algorithm based on the sentence similarity between the description of the ontology concept and its explanations in WordNet. We’ll apply the two senses determined by the two kinds of methods respectively to the calculation of similarity between the concept names. We mark the resulted similarities

between the two concept names $sim_hyperHypo(cn_1, cn_2)$ and $sim_commentSense(cn_1, cn_2)$. The final similarity formula of concept names is:

$$\begin{aligned} sim_{CN}(c_1, c_2) \\ = weight_hyperHypo \cdot sim_hyperHypo(cn_1, cn_2) \\ + weight_commentSense \\ \cdot sim_commentSense(cn_1, cn_2) \end{aligned} \quad (1)$$

Where, $weight_hyperHypo$ and $weight_commentSense$ are respectively the weights of the two algorithms. Make

$$\begin{aligned} weight_hyperHypo = \\ \frac{sim_hyperHypo(cn_1, cn_2)}{sim_hyperHypo(cn_1, cn_2) + sim_commentSense(cn_1, cn_2)}, \\ weight_commentSense = \\ \frac{sim_commentSense(cn_1, cn_2)}{sim_hyperHypo(cn_1, cn_2) + sim_commentSense(cn_1, cn_2)}, \end{aligned}$$

and then we can get $weight_hyperHypo + weight_commentSense = 1$.

Using the weighted sum of the similarities calculated by the two methods, this approach not only combines the advantages of two methods, but also complements their shortcomings. Meanwhile it eliminates the inaccurate factors resulted from the effects of the word ambiguity on ontology mapping.

3.1.2 The Algorithm of Property Similarity

(1) The similarity calculation between a pair properties

A. Calculate the weighted sum of property name and property type

Suppose P_{1i} is the i -th property of $c_1 \in O_1$, and P_{2j} is the j -th properties of $c_2 \in O_2$. The weighted sum of the property name similarity and property type similarity between two properties is:

$$weight_{pn} \times sim(pn_{1i}, pn_{2j}) + weight_{pt} \times sim(pt_{1i}, pt_{2j}) \quad (2)$$

In Eq. (2), pn and pt mean respectively the the property name and property type, $sim(pn_{1i}, pn_{2j})$ and $sim(pt_{1i}, pt_{2j})$ are similarities of the property name and property type. Here, the similarities of property name and property type are both computed through concept name similarity calculation algorithm (seen in Sect. 3.1.1). $weight_{pn}$ and $weight_{pt}$ represent the importance of property name, property type in property.

B. Calculate the importance of property in concept

$$importance_{c(p)} = tf_{c(p)} \times idf_p, \text{ where, } tf_{c(p)} = \frac{1}{pnum_c}; idf_p = \log \frac{cnum}{cnum_p}$$

Among them, $importance_{c(p)}$ means the importance of the property p in the concept c , the calculation of the importance references $tf-idf$ algorithm, in which $pnum_c$ is the

number of properties in concept c , and $cnum$ and $cnum_p$ are the numbers of the all concepts and the concepts including property p among ontologies respectively.

Then merge the importance of the two properties, the equation is as follows:

$$\begin{aligned} importance_{c1(p_i), c2(p_j)} \\ = \frac{importance_{c1(p_i)}}{importance_{c1(p_i)} + importance_{c2(p_j)}} \cdot importance_{c1(p_i)} \\ + \frac{importance_{c2(p_j)}}{importance_{c1(p_i)} + importance_{c2(p_j)}} \cdot importance_{c2(p_j)} \\ = \frac{importance_{c1(p_i)}^2 + importance_{c2(p_j)}^2}{importance_{c1(p_i)} + importance_{c2(p_j)}} \end{aligned}$$

Where, the $importance_{c1(p_i)}$ means the importance of the i property of the concept $c1$, the $importance_{c2(p_j)}$ means the importance of the j property of the concept $c2$, the $importance_{c1(p_i)} / (importance_{c1(p_i)} + importance_{c2(p_j)})$ is the weight of the $importance_{c1(p_i)}$, the $importance_{c2(p_j)} / importance_{c1(p_i)} + importance_{c2(p_j)}$ is the weight of $importance_{c2(p_j)}$.

C. Calculate the similarity between two properties

Composite the results above, the equation of calculating property p_{1i} and property p_{2j} similarity is as follows:

$$\begin{aligned} sim(p_{1i}, p_{2j}) \\ = importance_{c1(p_{1i}), c2(p_{2j})} \times weight_{pn} \times sim(pn_{1i}, pn_{2j}) \\ + importance_{c1(p_{1i}), c2(p_{2j})} \times weight_{pt} \times sim(pt_{1i}, pt_{2j}) \end{aligned} \quad (3)$$

(2) Calculate property similarity between two concepts

The number of properties in concept c_1 and c_2 is defined as $pnum_{c1}$ and $pnum_{c2}$, respectively. Calculate similarity of any two properties which are all emerged in concepts c_1 and c_2 . Then get a matrix of $pnum_{c1} \times pnum_{c2}$ similarity. Firstly traverse the matrix, and find the max values of every row and every column, then computer the average value of the row max values and the average of the column max values, finally the property similarity is the average of the above two average values. The following is the equation of property similarity between two concepts.

$$\begin{aligned} sim_P(c_1, c_2) = sim_P(P_{c1}, P_{c2}) \\ = \left(\frac{\sum_{i=1}^{pnum_{c1}} \max_{j=1}^{pnum_{c2}} (sim(P_{1i}, P_{2j}))}{pnum_{c1}} + \frac{\sum_{j=1}^{pnum_{c2}} \max_{i=1}^{pnum_{c1}} (sim(P_{1i}, P_{2j}))}{pnum_{c2}} \right) / 2 \end{aligned} \quad (4)$$

3.1.3 The Algorithm of Instance Similarity

For calculating the instance similarity, if the similarity of every property value pair between two instances needs to be computed, then the time cost of similarity calculation would be very high. So, in this paper, the improved Vector Space

Model (VSM) is used to compute the instance similarity between two ontology concepts. It can decrease dramatically the time cost. The improved VSM method is as follow:

In ontology, every concept has many instances and every instance has several property values. All property values belonging to a concept are seen as a text set. Therefore, every concept is represented as a text set, expressed as a weight vector, and denoted by $(weight_{ci(v_1)}, weight_{ci(v_2)}, \dots, weight_{ci(v_s)}, \dots)$, where $s = 1, 2, \dots, n$. Here, c_i means the i -th concept; n means the number of property values in c_i ; and $weight_{ci(v_s)}$ means the weight of the property value V_s in the text set of c_i . The instance similarity between $c_1 \in O_1$ and $c_2 \in O_2$ can be computed by cosine of angle between two weight vectors, and its equation is as follows:

$$sim_I(c_1, c_2) = \frac{\sum_{s=1}^n (weight_{c_1(v_s)} \cdot weight_{c_2(v_s)})}{\sqrt{\sum_{s=1}^n weight_{c_1(v_s)}^2} \cdot \sqrt{\sum_{s=1}^n weight_{c_2(v_s)}^2}} \quad (5)$$

Where

$$weight_{ci(v_s)} = \sum_{s'=1}^n (tf_{ci(v_{s'})} \cdot idf_{v_{s'}} \cdot sim(v_s, v_{s'})) \quad (6)$$

In Eq. (6), the calculation of the importance references the tf-idf algorithm. Among them $tf_{ci(v_{s'})} \cdot idf_{v_{s'}}$ means the initial importance of the property value $V_{s'}$ in concept c_i . $tf_{ci(v_{s'})}$ means the number of occurrence of $V_{s'}$ in c_i ; $idf_{v_{s'}} = \log(cnum/cnum_{v_{s'}})$, here $cnum$ means the number of concepts and $cnum_{v_{s'}}$ is the number of concepts including property $V_{s'}$.

In traditional VSM methods, $weight_{ci(v_s)} = tf_{ci(v_s)} \times idf_{v_s}$, and terms in concepts are identified to be equivalent fully or be unequal absolutely. But, in fact, non-identical terms may be very similar, and this situation cannot be dealt with in traditional methods. So, in this paper, if a term is highly similar with other terms in c_1 and c_2 , then it shows that the importance of other terms in concepts can be granted partly to this term. The degree of the granted part depends on similarities between this term and other terms.

3.1.4 The Algorithm of Structure Similarity

Semantic similarity between respective neighbor node sets of two concepts is called as structure similarity and can impact on concept similarity. The neighbor nodes of a concept mean that they have structural relations with this concept. The structural relations between two concepts in the same ontology include: hierarchical and non-hierarchical structural relations. Super, sub and sibling relations are hierarchical structure relations and the rest are non-hierarchical relations. In our approach, the structure similarity consists of the similarity between the adjacent concept sets and the similarity between the structure parameters.

Computing the similarity between the adjacent concept sets as follows:

$$sim(C_SR_1, C_SR_2)$$

$$= weight_{sim(C_SUPSET)} \times sim(C_SUPSET_1, C_SUPSET_2) \\ + weight_{C_SUBSET} \times sim(C_SUBSET_1, C_SUBSET_2) \\ + weight_{C_NHSET} \times sim(C_NHSET_1, C_NHSET_2) \quad (7)$$

where $sim(C_SUPSET_1, C_SUPSET_2)$ is the similarity between $sim(C_SUPSET_1)$ and $sim(C_SUPSET_2)$, $sim(C_SUPSET_1)$ and $sim(C_SUPSET_2)$ is the set of super classes of concept c_1 and c_2 , respectively. $sim(C_SUBSET_1, C_SUBSET_2)$ is the similarity between C_SUBSET_1 and C_SUBSET_2 , C_SUBSET_1 and C_SUBSET_2 are the sets of subclasses of concept c_1 and c_2 , respectively. $sim(C_NHSET_1, C_NHSET_2)$ is the similarity between C_NHSET_1 and C_NHSET_2 . Here, C_NHSET_1 and C_NHSET_2 are the two non-hierarchical relation sets of the concept c_1 and c_2 . $Weight_{C_SUPSET}$, $weight_{C_SUBSET}$, $weight_{C_NHSET}$ represents the importance of super concept, sub concept and adjacent concept respectively, and every weight will be got by computing its harmony.

Computing the similarity between structure parameters. The structure parameters include: the number of concept property, the number of instance, the number of sub concept and the conceptual depth in the ontology. For example, the number of concept property for c_1 is $pnum_{c_1}$, the number of instance, the number of sub concept and the concept of depth in the ontology O_1 for c_1 is $inum_{c_1}$, $subnum_{c_1}$ and $depth_{c_1}$, respectively. The equation of calculating similarity between the structure parameters of c_1 and c_2 is:

$$sim(pnum_{c_1}, pnum_{c_2}) = 1 - \frac{|pnum_{c_1} - pnum_{c_2}|}{\max(pnum_{c_1}, pnum_{c_2})} \quad (8)$$

$$sim(inum_{c_1}, inum_{c_2}) = 1 - \frac{|inum_{c_1} - inum_{c_2}|}{\max(inum_{c_1}, inum_{c_2})} \quad (9)$$

$$sim(subnum_{c_1}, subnum_{c_2}) = 1 - \frac{|subnum_{c_1} - subnum_{c_2}|}{\max(subnum_{c_1}, subnum_{c_2})} \quad (10)$$

$$sim(depth_{c_1}, depth_{c_2}) = 1 - \frac{|depth_{c_1} - depth_{c_2}|}{\max(depth_{c_1}, depth_{c_2})} \quad (11)$$

The equation of calculating structure parameter similarity between c_1 and c_2 is:

$$sim(SP_{c_1}, SP_{c_2}) \\ = weight_{pnum} \times sim(pnum_{c_1}, pnum_{c_2}) \\ + weight_{inum} \times sim(inum_{c_1}, inum_{c_2}) \\ + weight_{subnum} \times sim(subnum_{c_1}, subnum_{c_2}) \\ + weight_{depth} \times sim(depth_{c_1}, depth_{c_2}) \quad (12)$$

Structure parameter similarity is the similarity weighted sum of the number of properties, the number of instances, the number of sub-concepts and conceptual depth between the two concepts. The weights are respective proportion of four similarities.

After computing the similarity between the adjacent concept sets and the similarity between the structure parameters, we compute their weighted sum, and the weights are

Table 1 Similarity matrix.

	e21	e22	e23	e24
e11	1	0.99	0.99	0.2
e12	0.99	1	0.99	0.1
e13	0.99	0.99	1	0.1
e14	0.1	0.3	0.1	0.1

respective proportion of the two similarities, then we can get the structure similarity. The equation is as follows:

$$\begin{aligned} \text{sim}_s(c1, c2) = & \text{weight}_{c_sr} \text{sim}(C_SR_1, C_SR_2) \\ & + \text{weight}_{sp} \text{sim}(SP_{c1}, SP_{c2}) \end{aligned} \quad (13)$$

3.2 Ontology Mapping Discovery Based on SVM Combined with Harmony and Reliability

In this paper, the method of SVM classification combined with the linear function of harmony [20] and reliability [21] was adopted to achieve ontology mapping discovery. Harmony can be used to assess the importance of different individual similarities. But, sometimes, with the same harmony, the difference between the maximum value and other values in the same row and column is tiny, which decreases the credibility of results, shown in Table 1. So, reliability needs to be applied with harmony to increase the credibility. Before SVM classification method is carried out, individual similarities need to be weighted by the linear function of harmony and reliability.

The method of SVM classification combined with the linear function of harmony and reliability is as follows:

1) Harmony

Each of individual similarity matrixes has a harmony, which is defined as:

$$h = \frac{\max \text{num}_{\text{row} \cap \text{column}}}{\min(\text{enum}_1, \text{enum}_2)} \quad (14)$$

In Eq. (14), $\max \text{num}_{\text{row} \cap \text{column}}$ is the number of maximum in the same rows and same columns in the matrix, enum_1 is the number of concepts e_{1i} in O_1 , and enum_2 is the number of concepts e_{2j} in O_2 . Harmony represents the ratio of concept pairs which can be mapped to all concept pairs, that is, the ability to discover the mapped concept pairs.

2) Reliability

Reliability (rel) is defined as:

$$\text{rel} = \begin{cases} 0 & \text{if } \max \text{num}_{\text{row} \cap \text{column}} = 0 \\ 1 & \text{if } \text{enum}_1 = \text{enum}_2 = 1 \\ \frac{\sum_{(i,j) \in \text{MAXSET}_{\text{row} \cap \text{column}}} \left(\sum_{k=1}^{\text{enum}_1} |a_{ij} - a_{kj}| + \sum_{k'=1}^{\text{enum}_2} |a_{ij} - a_{ik'}| \right)}{\min(\text{enum}_1, \text{enum}_2) \times (\text{enum}_1 + \text{enum}_2 - 2)} & \text{otherwise} \end{cases} \quad (15)$$

Where, $\text{MAXSET}_{\text{row} \cap \text{column}}$ is the set of elements which are maximum in the same rows and same columns in the matrix. The reliability value is decided by the difference between maximum and other values, which must be from the same row and column. Reliability represents similarity gap between reasonable mapping elements and unreasonable mapping elements in the similarity matrix of some

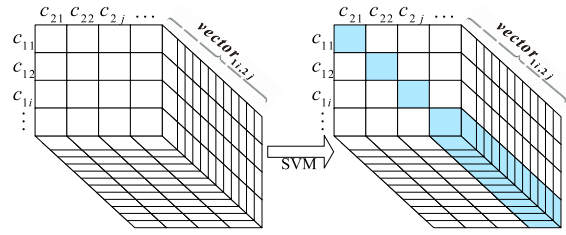


Fig. 3 Ontology mapping method based on SVM.

features. Reliability is able to objectively reflect the important level of certain feature.

3) Ontology mapping discovery based on SVM combined with harmony and reliability

Traditional classification methods about machine learning require a great deal of samples. In most cases the model learned from machine learning is fit excessively for a function, it results in poor expansion capability of the model. However, SVM has its unique advantages, and SVM is a machine learning method based on structural risk minimization, and its result is the unique global optimal result.

Figure 3 is the illustration of using SVM classification to achieve ontology mapping input vector. Each cell in cubes of a two-dimensional plane represents a pair of ontology concepts, while in three-dimensional space, this cell represents the condition input vector of this pair of concepts. Each vector has 12 parameters. All similarity vectors representing concept pairs are classified by SVM as two categories: one includes concept pairs which can be mapped (blue vector on the right) and another includes concept pairs which cannot be mapped (white vector on the right).

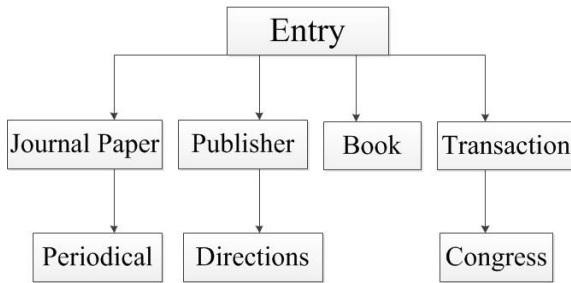
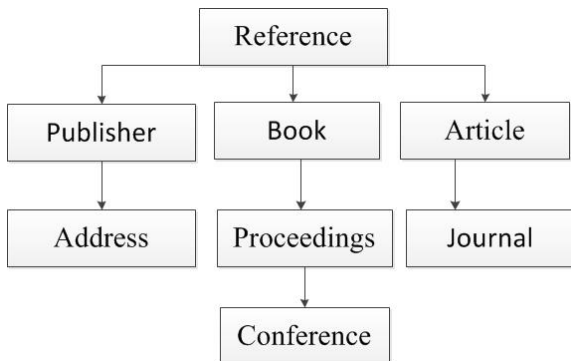
The input vector is a 12-dimensional similarity vector. Calculated four sub-similarities include concept name, property, instance, and structure. Conditioned input vectors of 12 parameters are the production after each sub-similarity multiplied by the respective harmony, reliability, and f-measure. Similarity vector is defined as:

$$\begin{aligned} \text{vector}_{1,2j} = & (h_{CN} \cdot \text{sim}_{CN}(c_{1i}, c_{2j}), h_P \cdot \text{sim}_P(c_{1i}, c_{2j}), \\ & h_I \cdot \text{sim}_I(c_{1i}, c_{2j}), h_S \cdot \text{sim}_S(c_{1i}, c_{2j}), \\ & \text{rel}_{CN} \cdot \text{sim}_{CN}(c_{1i}, c_{2j}), \text{rel}_P \cdot \text{sim}_P(c_{1i}, c_{2j}), \\ & \text{rel}_I \cdot \text{sim}_I(c_{1i}, c_{2j}), \text{rel}_S \cdot \text{sim}_S(c_{1i}, c_{2j}), \\ & f_{CN} \cdot \text{sim}_{CN}(c_{1i}, c_{2j}), f_P \cdot \text{sim}_P(c_{1i}, c_{2j}), \\ & f_I \cdot \text{sim}_I(c_{1i}, c_{2j}), f_S \cdot \text{sim}_S(c_{1i}, c_{2j})) \end{aligned} \quad (16)$$

In Eq. (16), $\text{sim}_{CN}(c_{1i}, c_{2j})$, $\text{sim}_P(c_{1i}, c_{2j})$, $\text{sim}_I(c_{1i}, c_{2j})$ and $\text{sim}_S(c_{1i}, c_{2j})$ are the similarities of concept name, property, instance and structure, respectively; h_{CN} , h_P , h_I and h_S are the harmony of similarities of concept name, property, instance and structure, respectively; rel_{CN} , rel_P , rel_I and rel_S are the reliability of similarities of concept name, property, instance and structure, respectively; f_{CN} , f_P , f_I and f_S are the f-measure of similarities of concept name, property, instance and structure, respectively; the expression of f-measure refers to the formula (19).

Table 2 The mapping results between two ontologies.

Ontology 1 concept	Ontology 2 concept	Conditioned input vector	Mapping
Conference	Congress	(0.9524, 0.8254, 0.6414, 0.6351, 0.8543, 0.713, 0.4742, 0.5391, 0.9547, 0.8233, 0.6344, 0.6702)	Yes
Journal	Periodical	(0.9524, 0.8505, 0.6094, 0.7646, 0.8543, 0.7346, 0.4506, 0.6491, 0.9547, 0.8483, 0.6028, 0.8069)	Yes
Book	Book	(0.9524, 0.8957, 0.7143, 0.7933, 0.8543, 0.7737, 0.5281, 0.6735, 0.9547, 0.8934, 0.7065, 0.8372)	Yes
Reference	Entry	(0.9524, 0.8776, 0.6421, 0.7627, 0.8543, 0.7581, 0.4748, 0.6474, 0.9547, 0.8753, 0.6351, 0.8048)	Yes
Address	Directions	(0.9524, 0.7871, 0.5714, 0.7933, 0.8543, 0.6799, 0.4225, 0.6735, 0.9547, 0.7851, 0.5652, 0.8372)	Yes

**Fig. 4** Ontology 1 model.**Fig. 5** Ontology 2 model.

3.3 Example

The following example illustrates the proposed ontology mapping algorithms. Figure 4 and Fig. 5 are two ontologies to be mapped, which are different in language and structure. In accordance with SVM we combine the input parameters to get ontology mapping algorithm, draw two ontology concepts for each input vector and mapping result, as shown in Table 2.

4. Experiment and Result Evaluation

4.1 Experimental Data sets and Criterion

In this paper, the bibliographic ontology set of benchmark data sets[†] from OAEI ontology matching campaign is firstly applied as data sets, because the organization is authoritative in the field of ontology matching all over the world and it provides a unified benchmark test data for all researchers. The participants can use the data set to test their tools, but the data set are not used in formal match. The ontology sets contain 110 ontologies, and provide us with mapping answers.

In our experiments, all ontologies on the average were divided into five parts and numbered 1-5. Then we took four of them as the training set, and the rest one as a test set. We first trained number 1-4 ontologies, and got a training model. Then we trained No.1, 2, 3, 5 ontologies, and got second training model, and so on. We can get 5 training models, and the final result is the average result coming from the five models. The above method can guarantee the reliability of the training. Among the data sets, O_{101} is reference ontology in bibliographic domains. The others were variants and linguistics or structure of the variants were different from O_{101} . The mapping operations between O_{101} and these variants were implemented, and the results of mapping were evaluated. In this paper OAEI criteria (precision, recall and f-measure) were used to evaluate the results of mapping. Their equations are as follows:

$$\text{precision} : p = \frac{\text{num_correctmapping}}{\text{num_allfoundmapping}} \times 100\% \quad (17)$$

$$\text{recall} : r = \frac{\text{num_correctmapping}}{\text{num_alltruemapping}} \times 100\% \quad (18)$$

$$\text{f-measure} : f = \frac{2 \times p \times r}{p + r} \quad (19)$$

[†]<http://oaei.ontologymatching.org/2012/benchmarks/index.html>

In the above three equations, the *num_correctmapping* means the number of the concept pairs mapped correctly; *num_allfoundmapping* means the number of all discovered mapping concept pairs including the true and the false; and *num_alltremapping* means the number of all concept pairs which should be mapped.

4.2 Experiment Design

Ontology mapping methods related to similarity calculation have been discussed in many studies. We compare the mapping results by precision, recall and f-measure. The 4 sub-similarity methods with reasonable index (harmony), reliability index (reliability) and f-measure act as the 12 input parameters of SVM machine learning. The vector made up of these 12 parameters vector can reflect the degree of importance that each sub-similarity determines the concepts whether they can be mapped. To evaluate our approach, all mapping methods mentioned in the Table 3 are compared.

Experimental steps are as follows:

- (1) For all ontological concept pairs, the four individual similarities are calculated;
- (2) These similarities are aggregated and mappings between ontologies are extracted by using all methods of Table 3 except “Ours” and “SVM”;
- (3) For “SVM”, the four individual similarities of all ontology concept pairs are calculated and ontology mappings are extracted by SVM;
- (4) For our approach, the process is the same as

Table 3 Different ontology mapping methods.

Method	Formula or Model
1. Ours	$vector_{1i,2j} = (h_{CN} \cdot sim_{CN}(c_{1i}, c_{2j}), h_P \cdot sim_P(c_{1i}, c_{2j}),$ $h_I \cdot sim_I(c_{1i}, c_{2j}), h_S \cdot sim_S(c_{1i}, c_{2j}),$ $rel_{CN} \cdot sim_{CN}(c_{1i}, c_{2j}), rel_P \cdot sim_P(c_{1i}, c_{2j}),$ $rel_I \cdot sim_I(c_{1i}, c_{2j}), rel_S \cdot sim_S(c_{1i}, c_{2j}),$ $f_{CN} \cdot sim_{CN}(c_{1i}, c_{2j}), f_P \cdot sim_P(c_{1i}, c_{2j}),$ $f_I \cdot sim_I(c_{1i}, c_{2j}), f_S \cdot sim_S(c_{1i}, c_{2j}))$
2. SVM	$vector_{1i,2j} = (sim_{CN}(c_{1i}, c_{2j}), sim_P(c_{1i}, c_{2j}),$ $sim_I(c_{1i}, c_{2j}), sim_S(c_{1i}, c_{2j}))$
3. Neural network	$vector_{1i,2j} = (sim_{CN}(C_{ij}, c_{wj}), sim_P(c_{1i}, c_{2j}),$ $sim_I(c_{1i}, c_{2j}), sim_S(c_{1i}, c_{2j}))$
4. Harmony	$sim(c_{1i}, c_{2j}) = \left(\sum_{t=CN, P, I, S} h_t \cdot sim_t(c_{1i}, c_{2j}) \right) / 4$
5. Reliability	$sim(c_{1i}, c_{2j}) = \left(\sum_{t=CN, P, I, S} rel_t \cdot sim_t(c_{1i}, c_{2j}) \right) / 4$
6. Sigmoid	$sim(c_{1i}, c_{2j}) = \left(\sum_{t=CN, P, I, S} sigmoid(sim_t(c_{1i}, c_{2j})) \right) / 4$
7. Avg	$sim(c_{1i}, c_{2j}) = \left(\sum_{t=CN, P, I, S} sim_t(c_{1i}, c_{2j}) \right) / 4$
8. Min	$sim(c_{1i}, c_{2j}) = \min_{t=CN, P, I, S} (sim_t(c_{1i}, c_{2j}))$
9. Max	$sim(c_{1i}, c_{2j}) = \max_{t=CN, P, I, S} (sim_t(c_{1i}, c_{2j}))$
10. Class name	$sim(c_{1i}, c_{2j}) = sim_{CN}(c_{1i}, c_{2j})$
11. Property	$sim(c_{1i}, c_{2j}) = sim_P(c_{1i}, c_{2j})$
12. Instance	$sim(c_{1i}, c_{2j}) = sim_I(c_{1i}, c_{2j})$
13. Structure	$sim(c_{1i}, c_{2j}) = sim_S(c_{1i}, c_{2j})$

Sect. 3.2. After four individual similarities and their respective harmony and reliability were worked out, similarity vectors consisting of our weighted individual similarities are built, and the weights are the linear function of harmony and reliability. A part of vectors as training sets is used by SVM to get the optimal classification hyper-plane, and the remaining vectors as test sets are classified by SVM to two categories: the category includes concept pairs which can be mapped and another includes concept pairs which cannot be mapped. Here, the vector represents the similarity between the concept pairs belonging to different ontologies;

(5) For all ontology pairs, precision, recall and f-measure of ontology mapping discovery are calculated in every method mentioned in Table 3. And then the experimental results are evaluated.

4.3 Experimental Result and Analysis

The experimental results are shown in Fig. 6:

(1) Methods 10-13: The results of using one of individual similarities as concept similarity are relatively low, and both recall and f-measure are lower than 0.4, and precision are lower than 0.6. These four algorithms have lower f-measure which is no more than 40%, especially the “Instance” algorithm whose recall value is less than 20%, because the concepts including instances are not much. They use only a single similarity algorithm, which cannot adapt to all changes of heterogeneous ontologies, so the result is very low.

(2) Methods 6-9: The results of the methods of similarity aggregation are higher than the above, in which three indicators are all more than 0.5. It indicates that the methods of similarity aggregation perform better. “Sigmoid” algorithm is the most optimistic, whose f-measure is more than 70%, and the “Max” and “Min” algorithm are the most pessimistic, because the “Max” and “Min” only consider one sub-similarity, which cannot reflect the impact from ontology factors on the ontology mapping comprehensively.

(3) Our approach, “SVM”, “Neural network”, “Harmony” and “Reliability” outperforms the other fundamen-

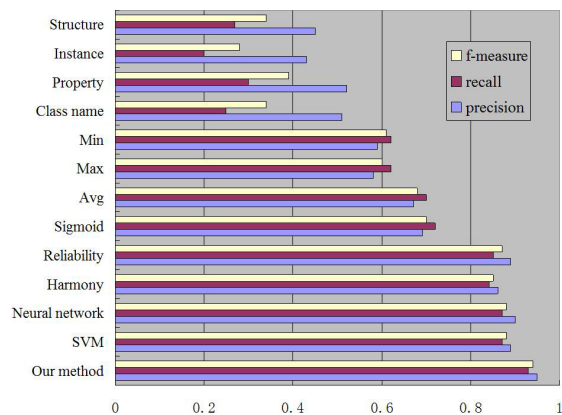


Fig. 6 The experimental results comparison of ontology mapping methods.

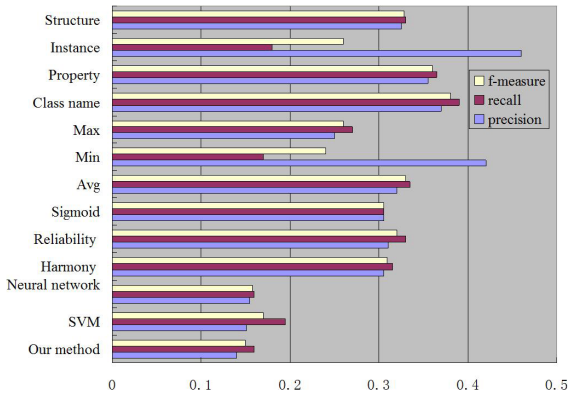


Fig. 7 Comparison of standard deviation of ontology mapping methods results.

tal methods, in which three indicators are all more than 0.8. But the precision, recall and f-measure of “SVM”, “Neural network”, “Harmony” and “Reliability” are all lower than 0.9.

(4) Our approach: precision, recall and f-measure in our approach reach 0.95, 0.935 and 0.9424, respectively, and are the highest, which can validate that the results of mapping discovery are more accurate after harmony and reliability is added into SVM, and also can show that our approach outperforms than the others dramatically.

The comparison of standard deviation is shown in Fig. 7:

The standard deviations of “10 Class name”, “11 Property”, “12 Instance” and “13 Structure” algorithms except recall and f-measure of “Instance” algorithm are all more than 30%, especially the precision of “Instance” algorithm which is more than 45%. Because a single similarity algorithm is good for heterogeneous ontology of specific types, and has poor performance for ontology of other types, when a variety of change appears in the language or the structure of the heterogeneous ontologies, a single similarity algorithm results fluctuate greatly and are instability. “8 Min” and “9 Max”: the results are more than 25% except recall and f-measure of “Min” algorithm, especially the precision of “Min”, which is more than 40%, and is very unstable. “4 Harmony”, “5 Reliability”, “6 Sigmoid” and “7 Avg” algorithms: the results of these methods are range from 30% to 35%, these results have a more stable fluctuation range compared with using a single sub-similarity algorithm. The results of the Neural network method, SVM method, Our method are lower than 0.2. And the results compared to other algorithms have smaller volatility. The standard deviations of precision and f-measure of “ours method” are the lowest of three algorithms, and they are less than 0.15.

Among them, the ontology mapping algorithms using integrated similarity and machine learning are better than those using only a single similarity algorithm. And the mapping algorithm using machine learning performs better compared with the integrated similarity algorithms.

The algorithm of concept name similarity can get high similarity values in the name of semantic or syntactic simi-

Table 4 The results of our method and OM2013 F-measure top5.

Matching system	Prec.	F-m	Rec.
Our system	0.93	0.87	0.82
CroMatcher	0.95	0.88	0.82
YAM++	0.97	0.89	0.82
CIDER-CL	0.85	0.75	0.67
IAMA	0.99	0.73	0.57
ODGOMS	0.99	0.71	0.55

larity. Such as ontology #101, #103, #104, #203-208, #221-247, #301-304 (The digital is the ontology label in benchmark). When the concept name is meaningless random code substitution, the algorithm has poor performance, such as ontology #202, #248-266.

The algorithm of property similarity can get high similarity values in adequate ontology property information and useful property name. For example, ontology #101, #103, #104, #203-208, #224-231, #236-238. When ontology has little or no property information, the algorithm has poor performance, such as ontology #221-223, #239, #246, #248, #249, #252, #253, #260.

The algorithm of instance can get high similarity values when ontology information is sufficient. For example, ontology #101, #103, #104, #201-210, #221-223, #251, #252. But when the number of instances of the concept is not much, the algorithm has poor performance.

The algorithm of structure similarity can get high similarity values in adequate ontology structure information. Such as ontology #103, #104, #201-210. When ontology has little or no structure information, the algorithm has poor performance, such as ontology #221-223, #232-248, #252-254, #258-261, #301-304.

Since the other algorithms are based on algorithms 10-13, in all ontology mapping operations, the other algorithms and algorithms 10-13 have the trend of consistent results to the same ontology pair. Among them, using the comprehensive similarity of ontology mapping algorithm and the use of machine learning ontology mapping algorithm are higher and more stable than using only a single similarity algorithm. In the above ontology mapping algorithms, our algorithm has better performance than the other algorithms.

In order to further test our algorithms, we use the benchmark data sets for the 2013 OAEI campaign[†] to re do the ontology matching experiments, and compare our results with the results of the methods mentioned in the 2013 OAEI campaign [36]. Given that OM2013 has a total of 22 results, we choose the F-measure top5 results. The mapping results of our method and OM2013 F-measure top5 are shown in Table 4. It can be seen from Table 4 that the precision, recall and f-measure of our method are better than the most of the results of OM2013.

In this paper, by using our proposed algorithm, we conducted two experiments on the two different benchmark test sets. There are differences between the two experimental re-

[†]<http://oaei.ontologymatching.org/2013/results/benchmarks/index.html>

sults. Among them, the latter recall decreased significantly. After the analysis, the reasons are as follows:

(1) The two experiment sets are named 2012 experiment set and 2013 experiment set, respectively. And comparing the same number of ontologies of two sets, 2013 experiment set has a higher degree of random code in concepts and properties. When calculating the similarities more random codes lead to a decreased recall. Our algorithms have specially processed the random codes in 2012 experiment set, while 2013 experiment set has more and more complex random codes, so it is clear that the processing strategies of our algorithms for the random codes are not applicable to the 2013 experiment set.

(2) Due to the affection of random codes, the structural similarity of 2013 experiment set becomes inaccurate. Since the concept names are not the same, although the same structure, we still cannot identify that the two concepts are similar on structure and semantic. Because the calculation of structural similarity centers on concept code and compares the concept similarity of parent codes and child nodes, thus, if the random codes of parent nodes and child nodes increase, the similarity of the conceptual node will be affected.

(3) Some ontologies in 2013 experiment set increase some redundant concepts, which leads to the decreased performance, especially in the calculation of structure similarity.

(4) Some of ontologies in 2013 experiment set has less elements. Such cases like property has no property value and there are less instances, make similarity decrease, and make results that should be similar dissimilar, which leads to a decline in performance.

The proposed algorithms consider the concept, instances, properties, structure of ontologies, use a variety of semantic computing strategies, and join a variety of factors such as harmony, reliability, f-measure, etc., as SVM classification data, which make the SVM classification data rich and reasonable, and achieve good results. But there are also some other deficiencies, such as a lack of parameters to adjust for users, and the shortage of the processing strategy of random code in ontologies.

5. Conclusions

A new similarity algorithm between concepts is proposed in this paper which fully considers the semantic and grammatical relations between individual elements. The addition of *harmony* and *reliability* increases the availability and accuracy of the algorithm. The algorithm can be used in application of semantic retrieval and knowledge management, and can increase the effectiveness of these applications. Experimental results show that, in similarity-based algorithms, the method of SVM combined with harmony and reliability has the better performance. The method can be used for RDF and OWL ontology matching. Future works include further research on ontology matching algorithm based on OWL ontology features and ontology merging.

Acknowledgments

This paper is supported by the China National Nature Science Foundation (No.61371194).

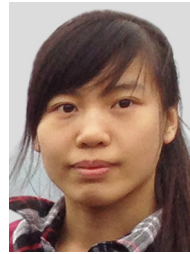
References

- [1] P. Shvaiko and J. Euzenat, "Ontology matching: State of the art and future challenges," *IEEE Trans. Knowl. Data Eng.*, vol.25, no.1, pp.158–176, 2013.
- [2] Li Huang, Guoxiong Hu, and Xinghe Yang, "Review of ontology mapping," 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), pp.537–540, Yichang, China, May 2012.
- [3] V. Cross and X.R. Yu, "Investigating ontological similarity theoretically with fuzzy set theory, information content, and tversky similarity and empirically with the gene ontology," *Proc. 5th International Conference on Scalable Uncertainty Management*, pp.387–400, Dayton, OH, USA, Oct. 2011.
- [4] A.L. Ginsca and A. Iftene, "Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment," *Proc. 9th RoEduNet IEEE International Conference, RoEduNet 2010*, Sibiu, Romania, pp.118–122, June 2010.
- [5] M.A. Rodriguez and M.J. Egenhofer, "Determining semantic similarity among entity classes from different ontologies," *IEEE Trans. Knowl. Data Eng.*, vol.15, no.2, pp.442–456, 2003.
- [6] G. Stoilos, G. Stamou, and S. Kollias, "A string metric for ontology alignment," *The Semantic Web-ISWC 2005, 4th International Semantic Web Conference, ISWC 2005. Proceedings*, pp.624–637, Galway, Ireland, Nov. 2005.
- [7] V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol.10, pp.707–710, 1966.
- [8] I. Akbari and M. Fathian, "A novel algorithm for ontology matching," *J. Information Science*, vol.36, no.3, pp.324–334, 2010.
- [9] P. Giuseppe and D. Talia, "UFOme: An ontology mapping system with strategy prediction capabilities," *Data & Knowledge Engineering*, vol.69, no.5, pp.444–471, 2010.
- [10] A. Genkin, D. Lewis, and D. Madigan, "Large-scale Bayesian logistic regression for text categorization," *Technometrics*, vol.49, no.3, pp.291–304, 2007.
- [11] H. Kim, P. Rowland, and H. Park, "Dimension reduction in text classification with support vector machines," *J. Machine Learning Research*, vol.6, pp.37–53, 2005.
- [12] A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy, "Learning to match ontologies on the semantic web," *VLDB Journal*, vol.12, no.4, pp.303–319, 2003.
- [13] J. Tang, J. Li, B. Liang, X. Huang, Y. Li, and K. Wang, "Using Bayesian decision for ontology mapping," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol.4, no.12, pp.243–262, 2006.
- [14] Y.R. Jean-Marya, E.P. Shironoshita, and M.R. Kabuka, "Ontology matching with semantic verification," *J. Web Semantics: Science, Services and Agents on the World Wide Web*, vol.7, no.3, pp.235–251, 2009.
- [15] X. Dong, J. Madhavan, and A.Y. Halevy, "Mining structures for semantics," *SIGKDD Explorations*, vol.6, no.2, pp.53–60, 2004.
- [16] A.G. Maguitman, F. Menczer, F. Erdinc, H. Roinestad, and A. Vespignani, "Algorithmic computation and approximation of semantic similarity," *World Wide Web-Internet and Web Information Systems*, vol.9, no.4, pp.431–456, 2006.
- [17] A. Doan, P. Domingos, and A. Halevy, "Reconciling schemas of disparate data sources: A machine-learning approach," *Proc of the ACM SIGMOD International Conference on Management of Data*, California, pp.509–520, 2001.

- [18] R. Trillo, J. Gracia, M. Espinoza, and E. Mena, "Discovering the semantics of user keywords," *J. Universal Computer Science*, vol.13, no.12, pp.1908–1935, 2007.
- [19] Q.V. Tran, R. Ichise, and B.Q. Ho, "Cluster-based similarity aggregation for ontology matching," *Proc. 6th International Workshop on Ontology Matching (OM-2011)*, pp.142–147, Bonn, Germany, CEUR-WS, Oct. 2011.
- [20] M. Mao, Y. Peng, and M. Spring, "An adaptive ontology mapping approach with neural network based constraint satisfaction," *J. Web Semantics*, vol.8, no.1, pp.14–25, 2010.
- [21] M. Houshmand, M. Naghibzadeh, and S. Araban, "Reliability-based similarity aggregation in ontology matching," *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2010)*, pp.744–749, Xiamen, China, 2010.
- [22] J. Gracia, J. Bernad, and E. Mena, "Ontology matching with CIDER: Evaluation report for OAEI 2011," *Proc. 6th International Workshop on Ontology Matching (OM-2011)*, Bonn, Germany: CEUR-WS, pp.126–133, Oct. 2011.
- [23] M. Rubiolo, M.L. Calusco, G. Stegmayer, M. Coronel, and M.G. Fabrizi, "Knowledge discovery through ontology matching: An approach based on an artificial neural network model," *Information Sciences*, vol.194, pp.107–119, 2012.
- [24] R. Ichise, "Machine learning approach for ontology mapping using multiple concept similarity measures," *ACIS-ICIS*, pp.340–346, 2008.
- [25] B.T. Le, R. Dieng-Kuntz, and F. Gandon, "On ontology matching problems-for building a corporate semantic web in a multi-communities organization," *Proc of 11th International Conference on EnterPrise Information Systems*, pp.236–243, Milan, Italy, 2009.
- [26] J. Euzenat and P. Valtehev, "Similarity-based ontology alignment in OWL-lite," *Proc 15th European Conference on Artificial Intelligence*, pp.333–337, Valencia, 2004.
- [27] S. Castano, A. Ferrara, and S. Montanelli, "Matching ontologies in open networked systems: Techniques and applications," *J. Data Semantics*, pp.25–63, 2006.
- [28] M. Ehrig and S. Staab, "QOM–quick ontology mapping," *Proc. International Semantic Web Conference 2004*, pp.683–697, 2004.
- [29] S. Melnik, H. Garcia-Molina, and E. Rahm, "Similarity flooding: A versatile graph matching algorithm and its application to schema matching," *Proc. 18th International Conference on Data Engineering (ICDE)*, pp.117–128, 2002.
- [30] N. Noy and M. Musen, "The PROMPT Suite: Interactive tools for ontology merging and mapping," *Int. J. Human–Computer Studies*, vol.59, no.6, pp.983–1024, 2003.
- [31] Y. Qu, W. Hu, and G. Cheng, "Constructing virtual documents for ontology matching," *Proc. 15th International Conference on World-WideWeb*, pp.23–31, 2006.
- [32] P. Wang, B. Xu, LILY, "the results for the ontology alignment contest OAEI 2007," *Proc. ISWC 2007 Ontology Matching Workshop*, Busan, pp.179–187, Korea, 2007.
- [33] J. Madhavan, P. Bernstein, and E. Rahm, "Generic schema matching with Cupid," *Proc 27th International Conference on Very Large Data Bases*, pp.48–58, Roma, 2001.
- [34] Y. Jean-Mary and M. Kabuka, "ASMOV results for OAEI 2007," *Proc. ISWC 2007 Ontology Matching Workshop*, pp.150–159, Busan, Korea, 2007.
- [35] W. Hu, N.S. Jian, Y.Z. Qu, and Y.B. Wang, "GMO: A graph matching for ontologies," *K-Cap 2005 Workshop on Integrating Ontologies*, pp.43–50, 2005.
- [36] B.C. Grau, Z. Dragisic, K. Eckert, et al., "Results of ontology alignment evaluation initiative," *OM2013*.



Jie Liu is an associate professor at College of Information and Engineering, Capital Normal University, Beijing, P.R. China. He received PH.D degree in Computer Application Technology at Beijing Institute of Technology. His main research interests are Semantic Web, Ontology Application and Natural Language Processing. Email: liujxxy@126.com.



Linlin Qin is a Master degree candidate whose specialty is Computer Application at College of Information and Engineering, Capital Normal University, Beijing, P.R. China. She received the degree of Bachelor of Management Studies in Computer Application at Anyang Normal College. Her main research interests are Ontology Application and Natural Language Processing.



Jing Gao is currently an assistant professor in the Computer Science and Engineering Department of the University at Buffalo. I got my PhD from Computer Science Department at University of Illinois at Urbana Champaign in 2011. My thesis is on exploring the power of heterogeneous information sources. I received M.E. and B.E. from the Computer Science and Technology Department at Harbin Institute of Technology in China. My Website: <http://www.cse.buffalo.edu/~jing/>



Aidong Zhang Department of Computer Science and Engineering, University at Buffalo, The State University of New York. Department Chair. IEEE Fellow. Dr. Aidong Zhang is appointed to the rank of SUNY Distinguished Professor in 2014. This is the highest academic rank available for any faculty member in the State University of New York system, conferred by a vote of the Board of Trustees of the State University of New York. As stated on the SUNY website, the Distinguished Professorship is conferred upon individuals who have achieved national or international prominence and a distinguished reputation within a chosen field. This distinction is attained through significant contributions to the research literature or through artistic performance or achievement in the case of the arts. The candidates' work must be of such character that the individuals' presence will tend to elevate the standards of scholarship of colleagues both within and beyond these persons' academic fields. Dr. Aidong Zhang also holds UB Distinguished Professor and Furnas Memorial Chair Professor positions at State University of New York at Buffalo. Website: <http://www.cse.buffalo.edu/faculty/azhang/>