Strategies Combination of Multi-strategy Ontology Mapping based on Entropy Decision-making

Rujuan Wang
College of Computer Science Jilin University, Changchun, China
wangrujuan_1108@sina.com

Lei Liu
College of Computer Science Jilin University
liulei@mail.jlu.edu.cn

Gang Wang
Northeast Normal University Archives, Changchun, China
wg08@sina.com

Abstract—Most of current ontology mapping methods can not treat different mapping tasks in different ways referred to the features of the input ontology. And they combine different mapping strategies without full consideration of the influences on mapping results caused mapping strategies. In view of the above questions, this paper proposes a dynamic mapping policy selection method which can pre-select mapping strategy and use entropy decision-making method to determine the combined weight of the selected strategy. Experiments show that this method can maintain the stability and the commonality, and improve the recall ratio and the precision ratio at the same time.

Index Terms—ontology mapping, entropy decision-making, strategy, similarity

I. INTRODUCTION

These years, ontology has become a hot topic in the field of artificial intelligence, knowledge representation, Semantic Web, data integration and information retrieval. But because the creators of ontologies use different methods, there must be disparity between ontologies created by different domain experts. The goal of ontology mapping is to solve the knowledge sharing and reuse problems of different ontologies.

Ontology mapping policies are exploited according to entities similarity computing of different ontologies, and these entities have infinite variety types of information (e.g. semantic information, structure information).

All the information can be regarded as the features of the ontology, besides the unitary mapping methods can’t get the whole information about entities of the ontology, so multi-strategy is widely used by present mapping methods.

But most of the methods combined strategies simply, they do not give due consideration to the features of ontology and the similar properties inside the mapping entity pair. These methods accumulate the results produced by different strategies to accomplish strategies integrated, and give a weight to every similarity, and then combine every similarity produced by different strategies with the weighted average method or Sigmoid function. These methods do not consider the different importance of every strategy. So if the mapping ontologies lost some features or some features changed, it will influence the quality of the mapping result.

To remedy these defects, this paper proposes a dynamic mapping policy which analyzes the similar information of the mapping ontology entities first, and select the mapping strategy, then use entropy decision-making method to determine the combined weight of the selected strategy, here we named the method as Automated Ontology Mapping based Entropy Decision-making (AOMED), the overall view of the ontology mapping framework architecture in Fig. 1. And give the mapping result finally. This method avoids the influences on the mapping result caused by worthless strategies, and there will be an upturn of efficiency, too. We begin with a discussion of the current state of the art in ontology matching. Following, we present a brief definition of the problem. Next, we provide details of the strategy selection and introduced four matchers of ontology mapping. In the next section, we explained the strategies combination of multi-strategy based on entropy decision-making. Then, we provide the results of two sets of experiments. Finally, our conclusions are stated.

II. RELATED WORK

Ontology matching is an active field of current research, with a vigorous community proposing numerous solutions. Euzenat and Shvaiko [15] present a comprehensive review of current approaches, classifying them along three main dimensions: granularity, input interpretation, and kind of input. The granularity dimension distinguishes between element-level and structure-level techniques. The input interpretation dimension is divided into syntactic, which uses solely the structure of the ontologies; external, which exploits auxiliary resources outside of the ontologies; and
semantic, which uses some form of formal semantics to justify results. The kind of input dimension categorizes techniques as terminological, which works on textual strings; structural, which deals with the structure of the ontologies; extensional, which analyzes the data instances; and semantic, which makes use of the underlying semantic interpretation of ontologies. A combination of several strategies can take full advantage of ontology information, so the result will be better than produced by single strategy.

There are a lot of multi-strategy methods at the present time. For example GLUE [14] system use machine learning methods to complete different ontology matching, the system use the information in the ontology entities to find the peculiar instance mode matching rules. RiMOM method proposes the ontology mapping model with minimum risk based on Bayesian decision theory, it converts mapping discovery problem to minimization risk problem, and provide a multi-strategy ontology mapping methods [5].

Yuzhong Zhai, et al, Southeast China University, design and put the Falcon-AO system [6] into practice, the system provide a matcher database which can chose different mapping strategies according to different mapping tasks. There are three kinds of matchers in the database, LMO, GMO and BMP. The FOAM system [16] aims at defining a framework for ontology mapping. It provides some basic strategies based on rule, machine learning and selection of candidate mappings. It also allows human intervention in the presence of vague mappings.

Similarly to these systems, AOMED adopts a combination of mapping strategies. However, it takes into account (through the strategies prediction) a crucial problem related to the configuration of a mapping strategy in terms of selection and parameter values assignment. Moreover, AOMED considered the influence of different strategies to the quality of the mapping result, so used entropy decision-making method to determine the combined weight of the selected strategy.

III. DEFINITION

This section provides preliminary definitions used throughout the paper.

A. Ontology

Definition 1 Ontology

Ontology is a six-tuple of the form:

$$O = \{C, P, H^c, H^p, A, I\}$$

consisting of a set of concepts $C$ and a set of properties $P$, respectively arranged in the hierarchies $H^c$ and $H^p$ that associate each concept $c$ with its sub-concepts $Sub(c)$ and each property $p_i$ with its sub properties $Sub(p_i)$, respectively $A$ is a set of axioms. The set $I$ contains instances of concepts and properties.

In order to clarify the terminology introduced in the next sections, in Fig. 2 an example of ontology is shown. Circles are used to represent concepts, solid rectangles to represent object properties (properties for which the value is an individual) and dotted rectangles to represent datatype properties (properties for which the value is a data literal). Moreover, an incoming arrow indicates the domain of a property while the range is indicated by an outgoing arrow. Inheritance relations are represented as solid arrows. Finally, instances of concepts and instantiation of relations are depicted as dotted arrows.

The considered ontology is meant to model the domain of person. As can be observed, two types of person, Man, and Woman. Moreover, a Person has a country, an age and to eat food. Finally, a food has a flavour. An excerpt of the OWL representation of this ontology is represented in the Fig. 2.
In more detail, the linguistic information included in the definition of a concept, property or instance encompasses the set of metadata in its name, comment(s), and label(s); this information is referred to as the linguistic description (LD):

$$LD(e) = \bigcup_{i} w_i$$

where \(e\) generally denotes a concept, a property or an instance and each \(w_i\) is a word in the name, label(s) or comment(s) of \(e\).

The structural information of a class \(c\) or property \(p\) indicates how it is arranged within its hierarchical structure (i.e., \(H^c\) or \(H^p\)). \(Sub(c)\) is indicated as the set of sub-concepts of \(c\) whereas \(Sub(p)\) refers to the set of sub-properties of \(p\). More formally:

$$Sub(c) = \bigcup_{i} c_i$$

where \(c_i\) is a sub concept of \(c\). On the other hand, for properties:

$$Sub(p) = \bigcup_{i} p_i$$

where \(p_i\) is a sub property of \(p\). Besides, structural information of properties can be further characterized since each property \(p\) is also defined in terms of its domain and range. With \(Dom(p)\) is indicated the set of concepts in which each concept has the property \(p\):

$$Dom(p) = \bigcup_{i} c_i$$

where each \(c_i\) has the property \(p\). On the other hand, \(Rang(p)\) is the set of concepts whose instances can be the value of \(p\):

$$Rang(p) = \bigcup_{i} c_i$$

where each \(c_i\) is a concept whose instances can be the value of \(p\).

Overall, the structural description (SD) can be defined for concepts and properties as follows:

$$SD(c) = Sub(c)$$

$$SD(p) = Sub(p) \cup Dom(p) \cup Rang(p)$$

Extensional information can be defined for concepts and properties as follows:

$$ED(i) = Inst(c) \cup Inst(p)$$

$$Inst(c) = \bigcup_{i} i_c$$

$$Inst(p) = \bigcup_{i} i_p$$

where \(i_c\) is an instance of \(c\).

$$OD(c) = LD(c) \cup Sub(c) \cup Inst(c)$$

$$OD(p) = LD(p) \cup Sub(p) \cup Dom(p) \cup Rang(p) \cup Inst(c)$$

$$OD(i) = ED(i)$$

B Ontology mapping

Up to now, ontology mapping problem has caught worldwide attention in the scientific community, but there is no standard definition of ontology mapping modality. The purpose of this article is to find 1:1 mapping, so uses the definition in literature cite [1], this definition is more convenient and flexible.

Definition 2 Ontology mapping

$$m = \{e, e, r, k\}$$

\(e\), represent entity in \(O\), \(e\), represent entity in \(O\), \(r\) represent a relationship between \(e\) and \(e\), (e.g..equivalence, inclusion, Included, Partitive, partial overlapping  ), \(k \rightarrow [0,1]\) represent the similarity of \(e\) and \(e\), \(k\) gets closer to 1 then \(e\) is more similar to \(e\).

C Similar function

The ontology enteritis is described with different ontology information, so first every mapping strategy
computes a similarity according to different information, and then combined these similarities to find the final similarity between source ontology and target ontology.

**Definition 3** Similarity function

Given two ontologies entities \( e_s \in O_s \), \( e_t \in O_t \), their overall similarity is computed by the following function:

\[
Sim(e_s, e_t) := F \left( \sum_i \text{sim}_i(e_s, e_t) \right)
\]  

(16)

where each \( \text{sim}_i \) is the \( i \)th similarity function implemented by an individual matcher and \( F \) is a function that combines the different similarity scores, for instance, through a weighted sum.

### IV. Strategies Prediction

Most of current methods use multi-strategy combination, it can help avoid recall ratio phenomenon cause by using unitary strategy, but at the same time we will be confronted with problems of strategy selection, strategy combination and parameter adjustment. This paper introduces a solution.

In the strategy selection, this paper thinks over the information of mapping ontology and the similarity information of mapping entity pair, and then give a mapping strategy selection proposal, and compute correlation parameters with this information (weights and threshold etcetera), the whole procedure is called strategy selection here in.

The basic thought of strategy selection is divided into three types, mapping strategy based on entity, mapping strategy based on structure and mapping strategy based on term, analysis information of the source mapping ontology and target mapping ontology, then compute the affinity coefficients of ontology with these three strategies, when one intimacy is larger than a particular value, then this strategy will be included in the pre-select strategies. Several affinity coefficients design procedure.

**A Lexical affinity coefficient**

The definition of the lexical affinity coefficient between two ontologies is as follows:

\[
L_s(O_s, O_t) = \frac{\#\text{common entities}}{\min(|S|, |T|)}
\]  

(17)

\#common entities represent all the similar entities in the description of semantic information of source ontology and target ontology (when the similarity of an entity pair is larger than the value of \( Th_s \)), semantic information of every entity includes language descriptions about name, labels and instructions.

**B Structural affinity coefficient**

Structural affinity coefficient aim to estimate the structure similarity between two ontologies, to compute Structural affinity coefficient, we should compute Structural information content of every entities (concepts or properties) in the two ontologies, it is computed according to the place in its own level, generally computed by the formulas in literature[2].

\[
Ic_s(e) = 1 - \frac{\log(Sub(e) + 1)}{\log(|E|)}
\]  

(18)

\( Sub(e) \) represent the number of Sub-entities entity \( e \) has in the hierarchy, and \( |E| \) represent the number of all entities in this level. Therefore, Structural affinity coefficient between two ontologies \( S_s \) defined as follows:

\[
S_s(O_s, O_t) = \frac{\#\text{common entities}}{\min(|S|, |T|)}
\]  

(19)

This \#common entities means two entities have very similar \( Ic_s \) (larger than a particular value of \( Th_s \)) and there are the same depth of level they belong to.

**C Instances affinity coefficient**

Generally speaking, strategies based on entity aim at ontologies with a large number of entities, if the number of entities is limited or none, then result produced by this type of strategies is low accuracy, here we use instances affinity coefficient \( I_s \) to estimate the similar of ontology entities.

\[
I_s = \frac{\#\text{common instances}}{\min(|S|, |T|)}
\]  

(20)

This \#common instances represents all similar entities in source ontology and target ontology semantic Information Description, when \( \min(|S|, |T|) = 0 \), \( I_s = 0 \).

**D On determining the optimal threshold values**

The basic idea in suggesting the threshold values is that the higher are \( L_s \), \( S_s \) and \( I_s \), the more results can be safely assessed. Therefore, as the affinity coefficients increase the thresholds values have to decrease. We argue that the exponential decreasing function is appropriate to reach this goal. The strategy predictor determines the optimal thresholds values as:

\[
Th_s = e^{-\alpha_s}, Th_s = e^{-\beta_s}, Th_s = e^{-\gamma_s}
\]  

(21)

where \( \alpha < 1, \beta < 1, \gamma < 1 \) are constants quantifying the monotonic decreasing rate.

### V. Strategies of Ontology Mapping

The pre-selection Strategy method proposed in this paper is of great versatility, the mapping Strategies provided by every mapping system are different, the strategy selection methods here supplies users a very good quantization parameter, and can make good decision to select mapping strategies automatically, so
here presented only several representative strategies. Every strategy correspond an input and output, the input is all entities of the two ontologies, the output is entity similarity vectors computed by each strategy, finally returns a similarity matrix \( S \). According to the pre-select strategies, where \( m \) is the number of entity pairs, \( n \) is the number of strategies.

\[
S = \{ \text{Sim}_y (e_i, e_j) \}_{i,j=1}^n
\]  
(22)

where \( \text{Sim}_y (e_i, e_j) \) is similarity of entity pair computed by each strategy, \( m \) is the number of entity pairs, \( n \) is the number of strategies. Several strategies are given as follows:

1) The lexical ontology matcher (LOM)
2) Latent Features Matcher (LFM)
3) The structural ontology matcher (SOM)
4) The instance-based matcher (IBM)

A The lexical ontology matcher (LOM)

As pointed out in [6], the lexical feature space consists of all the human-readable information provided in ontology. Three such lexical features are considered in OWL ontologies: the id, the label, and the comment. Let \( \Sigma \) denote a thesaurus, and \( \text{syn}(l) \) the set of synonyms and \( \text{ant}(l) \) the set of antonyms of label \( l \); the lexical similarity measure between the labels of \( e_i \) and \( e_j \), \( S_L (e_i, e_j) \) is then given as follows:

\[
S_L (e_i, e_j) = \begin{cases} 
1.0, & \text{if } l = l' \\
0.99, & \text{if } l, l' \in \text{syn}(l) \\
0.0, & \text{if } l, l' \in \text{ant}(l) \\
\frac{\text{Lin}(l, l')}{\text{max} \{ \text{tok}(l), \text{tok}(l') \}}, & \text{otherwise} 
\end{cases}
\]  
(23)

\( \text{Lin}(l, l') \) denotes the information theoretic similarity proposed by Lin in [7]; it provides a good measure of closeness of meaning between concepts within a thesaurus. The tokenization function \( \text{tok}(l) \) extracts a set of tokens from the label \( l \), \( S_{id} (e_i, e_j) \) is the ids similarity measure used the same way as with labels, except that the Lin function is not used. The lexical similarity for comments is compute used the following equation:

\[
S_L (e_i, e_j) = 1 - \frac{\text{op}(x_i, x_j)}{\text{max}\{\text{tok}(x_i), \text{tok}(x_j)\}}
\]  
(24)

\( S_L (e_i, e_j) \) as a variation of Levenshtein distance but applied to tokens. Let \( x_i, x_j \) be the comments of \( e_i, e_j \) respectively, and let \( \text{op}(x_i, x_j) \) denote the number of token operations needed, and \( \text{tok}(x) \) denote the number of tokens in a comment. The lexical similarity measure is calculated as the weighted average of the label, id, and comment similarities with the following similarity function:

\[
\text{Sim}_{\text{lex}} = F_c (S_L (e_i, e_j), S_{id} (e_i, e_j), S_{id} (e_i, e_j))
\]  
(25)

The function \( F_c \) defines a weighted sum of the similarity values as detailed in the next section.

B Latent Features Matcher (LFM)

The other matching method applied is the semantic matching one detailed in [8]. This method aims at discovering and exploiting latent features that reveal the intended meaning of ontology entities. The latent features matcher workflow is show in Fig.3. This is done by applying the reverse generative process of the Latent Dirichlet Allocation (LDA) [9] model. Doing so, each element is represented as a distribution over latent features, and similarities between entities’ pairs of the two ontologies is computed by means of the Kulback-Leibler divergence [10] measure. This measure estimates the divergence of distributions over latent features i.e., the divergence of entities’ approximated intended meaning. In more detail, the similarity between these entities is defined as follows:

\[
\text{Sim}_{\text{lf}} (e_i, e_j) = 1 - \frac{\text{KL}(e_i, e_j)}{\text{Max} \{ \text{KL}(i, j) \}} \forall i \in O, j \in O
\]  
(26)

The symmetric variant is defined as follows:

\[
\text{KL}(e_i, e_j) = I(e_i, e_j) + I(e_j, e_i)
\]  
(27)

The asymmetric KL-divergence between the entities is:

\[
I(e_i, e_j) = \sum_{x \in \Sigma} e_i \log \left( \frac{e_i}{e_j} \right)
\]  
(28)

![Figure 3. The LFM mapping process](image)

C The structural ontology matcher (SOM)

As pointed out in Section 3, an entity includes a structural description \( \text{SD}(e) \), which provides information about its neighbourhood. The SOM matcher aims at
exploiting structural descriptions of ontology entities. SOM translates the ontology mapping problem into a graph matching problem as OWL ontologies can be naturally seen as graphs in which concepts and properties are represented as nodes and OWL built-in properties as edges. SOM is based on the graph matching algorithm described in [11] and follows the work presented in [12]. According to Tous et al. [12], built-in RDF(S) and OWL properties can be modeled by a vector space model in which each property is represented as a dimension of a k-dimensional space where k is the number of built-in properties considered. Similarities between entities are collected into a similarity matrix \( S \) whose values are calculated by iterating the following updating equation:

\[
S_{k+1} = S_k + A^T \cdot S_k + S_k \cdot A^T, \quad k = 0,1,...n
\]  

(29)

where each element \( s_{ij} \) of \( S_k \) represent the similarity between a source entity \( e \in O_1 \) and a target entity \( e' \in O_2 \) at iteration \( k \). \( A \) and \( B \) are the adjacency matrices of \( O_1 \) and \( O_2 \), respectively. The algorithm stops when a predefined difference between \( S_k \) and \( S_{k+1} \) is reached.

D The instance-based matcher (IBM)

In instance-based mapping semantic relations between concepts of two ontologies are determined based on the overlap of their instance sets. This is a very natural approach, as in most ontology formalisms the semantics of the relations between concepts is defined via the set of their instances. The idea for mapping is then simply that the higher the ratio of co-occurring instances for two concepts, the more related they are. Similarity measure can define a similarity of any two ontologies involved. The well-known formula of similarity measure is Jaccard’s coefficient which appears in (Doan et al, 2002) as shown following:

\[
\text{Jaccard’s sim}(e, e') = \frac{P(e \cap e')}{P(e \cup e')} = \frac{P(e, e')} {P(e, e') + P(e, e') + P(e', e')} \]

(30)

To detail the formula, \( P(e, e') \) is a number of elements of the \( e \in O_1 \) and \( e' \in O_2 \) (concepts or properties) which are similar. \( P(e, e') \) is a number of elements of the entity \( e \) which are similar with elements of the entity \( e' \). \( P(e, e') \) is a number of elements of the entity \( e \) which are similar with elements of the entity \( e' \), the other details described in [13].

VI. STRATEGIES COMBINATION

Entropy in information system is the measure of information disorder, greater the entropy is, higher Information disorder is, the utility value of information is smaller; conversely, smaller the entropy is, lower Information disorder is, the utility value of information is greater. The combination of selected Strategies should especially consider the influence on mapping results caused by mapping strategy itself, this paper use the similarities produced by different strategies as strategy samples according to decision theory, and use information entropy to evaluate the effectiveness value of each strategy which will be used to compute the weight of each strategy in the system.

A Information entropy and Weighting functions in strategy combination

Using the similarity matrix \( S \) mentioned in the formula of previous section, \( m \) entity pairs as samples, \( n \) strategies as evaluating indicator, and this paper aims to evaluate the effectiveness value \( n \) strategies. Because the similarity matrix \( S \) is the same in dimension and quantity, here we pass over the standardization, and use \( y_i \) instead of \( \text{Sim}_j(e, e') \) to facilitate the description.

Using the formula (31) to compute information entropy value of strategy \( j \) based on information entropy theory.

\[
E_j = -k \sum_{i=0}^{m} y_i \ln y_i
\]

(31)

Where constant \( k \) related to system sample size \( m \), a system with completely disordered information, the degree of order is zero, and its entropy is maximum, \( E = 1 \). when \( m \) samples are in completely disordered distribution \( y_s = \frac{1}{m} \), calculated by the formula (32) and (33):

\[
E = -k \sum_{i=0}^{m} \frac{1}{m} \ln \frac{1}{m} = k \sum_{i=0}^{m} \frac{1}{m} \ln m = k \ln m = 1
\]

(32)

\[
k = \frac{1}{\ln m}
\]

(33)

Because information entropy \( E_j \) is used to measure the utility value of strategy \( j \), when completely disordered distribution, \( E_j = 1 \). Here, the information of \( E_j \) (data of strategy \( j \) target) utility value of overall evaluation is zero. So, the information utility value of an index is dependent on its difference value \( h_j \) of information entropy \( E_j \) and 1.

\[
h_j = 1 - E_j
\]

(34)

It is clear that using entropy method to estimate the weight of each strategy, its essence is to compute with the cost coefficient of the strategy information (similarity), if
the cost coefficient is higher, the more important it means to the final result, so weight of strategy \( j \) is:

\[
w_j = \frac{h_j}{\sum_{j=1}^N h_j}
\]

(35)

Entropy method is based on the difference of information order degree of each index, videlicet the Utility value of information to determine the weight of each index. So, this method can determine the weight of strategy according to the embodied effectiveness of similarity different strategies, and it can avoid the results deviation caused by human intervention or weighting simply in traditional methods to ensure the mapping precision ratio.

VII. EXPERIMENTAL RESULTS

This section discusses in detail the results of the evaluation of the strategy predictor module, which aims at automatically determining the most suitable mapping strategy in terms of parameter values. The first set of experiments was done using the 2008 benchmark series of tests created by the Ontology Alignment Evaluation Initiative (OAEI) [4], in order to determine the accuracy of the method. The second set of experiments was performed using the NCI Thesaurus (describing the human anatomy) and the Adult Mouse Anatomy ontologies, which are also part of the OAEI 2008 contest, in order to analyze the algorithm using different thesauri. All experiments were performed on an Intel Core 2 running at 2.0 GHz with 2 GB of memory.

A Parameter tuning

The strategy predictor uses some smoothing factors and threshold values in its definition. Table I summarizes the values of these parameters used during the evaluations. Values for these parameters have been experimentally obtained by using a subset of ontologies in the considered dataset. In particular, ten ontologies belonging to the first set were used to fine-tune these parameters.

In order to investigate further the performance of AOMED, we used the provided partial alignment as a gold standard. We then ran AOMED using two different combinations of weights: the first set is the one used for the OAEI contest, which had to be the same set as those used for the benchmark, and the second set was derived experimentally to improve accuracy. The results of this further investigation are presented in Table II, the standard weights used were \( w_{\text{LMS}} = 0.3, w_{\text{SOM}} = 0.3, w_{\text{IBM}} = 0.2, w_{\text{Sa}} = 0.2 \). The entropy weights computed were \( w_{\text{LMS}} = 0.4, w_{\text{SOM}} = 0.304, w_{\text{IBM}} = 0.167, w_{\text{Sa}} = 0.129 \).

B Evaluation of accuracy

The goal of ontology matching is to generate an alignment that discovers all correct correspondences, and only correct correspondences, where correctness is judged with respect to a human interpretation of meaning. In some cases, either incorrect correspondences are discovered, or correct correspondences are not. To evaluate accuracy of ontology matching, it is necessary to quantify both the number of correct correspondences not found, and the number of incorrect correspondences found. This is done by using a gold standard alignment between two ontologies previously derived by human experts, running the algorithm on the ontologies, and then calculating precision \( P \), the percentage of gold standard correspondences that exist within the extracted alignment, recall \( R \), the percentage of correct extracted correspondences that exist within the gold standard, and \( F_m \) the harmonic mean of precision and recall. Let \( G \) be the gold standard alignment, and \( A \) be the alignment extracted by the ontology matching algorithm,

\[
P = \frac{|A \cap G|}{|A|}, \quad R = \frac{|A \cap G|}{|G|}, \quad F_m = \frac{2PR}{P + R}
\]

(36)

| TABLE I. PARAMETER VALUES AFTER EXPERIMENTAL TUNING. |
|----------------|----------------|
| Parameter      | Usage          | Value |
| \( \alpha \)   | Smoothing factor in Eq. (21) | 0.4 |
| \( \beta \)    | Smoothing factor in Eq. (21) | 0.4 |
| \( \gamma \)   | Smoothing factor in Eq. (21) | 0.4 |
| \( T_{S_n} \)  | Threshold used in the La computation | 0.6 |
| \( T_{S_a} \)  | Threshold used in the Sa computation | 0.6 |
| \( T_{L_n} \)  | Threshold used in the La computation | 0.6 |
| \( T_{L_a} \)  | Threshold used to decide whether LMS has to be exploited | 0.4 |
| \( T_{SOM} \)  | Threshold used to decide whether SOM has to be exploited | 0.4 |
| \( T_{IBM} \)  | Threshold used to decide whether IBM has to be exploited | 0.4 |

| TABLE II. TOTAL NUMBER OF CORRESPONDENCES FOUND FOR PARTIAL ALIGNMENT IN ANATOMY TESTS |
|---------------------------------|-----------------|-----------------|
|                                | Standard weights | Entropy weights |
| Correct correspondences found   | 845              | 893             |
| Correspondences found but not in gold standard | 412              | 424             |
| Correspondences in gold standard not found | 124              | 76              |
| Precision                       | 0.672            | 0.678           |
| Recall                          | 0.872            | 0.921           |

We have evaluated the accuracy of AOMED using the well established OAEI benchmark series of tests. The benchmark tests start from reference ontology to a multitude of alterations. As ontologies may be modeled in a different manner by different developers, the variations between the tests highlight how well the algorithm would perform in the real world. Fig. 4 graph’s these values for
AOMED against all other entrants in the OAEI 2008 campaign.

![Accuracy of AOMED vs. OAEI 2008 entrants](image)

AOMED was compared with other ontology mapping algorithms on the OAEI test sets in Table III.

<table>
<thead>
<tr>
<th>System</th>
<th>P</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIMOM</td>
<td>0.8</td>
<td>0.81</td>
</tr>
<tr>
<td>Lily</td>
<td>0.89</td>
<td>0.79</td>
</tr>
<tr>
<td>ASMOV</td>
<td>0.81</td>
<td>0.77</td>
</tr>
<tr>
<td>AROMA</td>
<td>0.82</td>
<td>0.71</td>
</tr>
<tr>
<td>SAMBO</td>
<td>0.95</td>
<td>0.8</td>
</tr>
<tr>
<td>AOMED</td>
<td>0.8</td>
<td>0.82</td>
</tr>
</tbody>
</table>

Table III reports the partial results on the OAEI 2008. It can be observed that the three systems, which on the OAEI 2007 obtained the best results, even in these tests obtained good results. In particular, AOMED obtained the same value of P. Note that as compared to RIMOM, AOMED obtained better results even if the best system is SAMBO. SAMBO is a system including several matchers that may use knowledge from different sources. The system presents some alignment suggestions that are determined by combining and filtering the results generated by one or more matchers. The suggestions are then presented to the user who accepts or rejects them. The acceptance and rejection of a suggestion may influence further suggestions. Further, a conflict checker is used to avoid conflicts introduced by the alignment relationships. AOMED through the strategy predictor decreases the burden to the user by suggesting the most appropriate matching strategy on the basis of the particular mapping task.

**VIII. CONCLUSIONS**

This paper proposes a dynamic mapping policy which analyzes the similar information of the entities, which by relying on three affinity coefficient obtained by scrutinizing commonalities between the two ontologies under consideration is able to suggest values for cut-off threshold, we use entropy decision-making method to determine the combined weight of the selected strategy, the combination should especially consider the influence on mapping results caused by mapping strategy itself. Overall, this method can maintain the stability and the commonality, and improve the recall ratio and the precision ratio at the same time.

**ACKNOWLEDGMENT**

This paper supported by the National Natural Science Foundation of China under Grant No.60873044.

**REFERENCES**