Proceedings of the 6th World Congress on Intelligent Control
and Automation, June 21 - 23, 2006, Dalian, China

Ontology Mapping Approach Based on Concept Partial Relation*

Pengfei Qian and Shensheng Zhang

Department of Computer Science & Technology
Shanghai Jiaotong University
800 Dong Chuan Rd, Shanghai, China

{qianpf & sszhang}@sjtu.edu.cn

***Abstract* - A kind of ontology mapping approach based on concept partial relation is introduced. In this approach, the traditional mapping relation among the different ontology concepts will be extended to compound relation, including 'intersection', 'equal', 'disjunction', 'subsumption', 'be subsumed' relation. In this paper, 2 questions about ontology mapping approach based on concept partial relation should be answered, the first is how to automatically reason and compute the complex semantic partial relation among the ontologies, the second is how to use the semantic partial relation and the existing mapping results to instruct the following mapping, and then improve the mapping accuracy and efficient. To solve these 2 problems, concept partial mapping (CPM) and CPM-based ontology mapping approach are introduced respectively. Finally, the mapping accuracy and time complexity of CPM-based ontology mapping approach are introduced compared to traditional mapping approach.**

***Index Terms* –ontology mapping, partial relation, mapping accuracy, time complexity**

I. Introduction

For implementing the automatic matching between the similar ontologies of different domains, the first problem is how to realize the automatic matching among the concepts of different ontologies. General speaking, probability inference will be used to compute the similarity between 2 different concepts (such as GLUE system [1][2]), but the existing matching approach is so coarseness that the relations among the concepts are confined as 'Equal' or 'Not Equal' [3], consequently, the basic set characters of ontology concept (such as the subsumption and intersection relation among the sets) are ignored.

How to use different semantic partial relations to finish the entire ontology mapping process, how to use the existing mapping results and constraints to instruct the realization of the following mappings, those are key problems discussed in this paper. Therefore, a kind of ontology mapping approach based on concept partial relation is presented, in this approach, the traditional mapping relation among the ontology concepts will be extended to compound relation, including 'Equal', 'Subsumption', 'Be subsumed' 'disjunction', and 'Intersection' [4][5], the complex semantic partial relation among ontology concepts will be imported, the existing mapping results will be used to shorten the mapping range of the unknown concept. In the end, the mapping accuracy and time complexity of CPM-based ontology mapping approach are introduced associated with the GLUE system.

The remained part of this paper is structured as following. Section 2 mainly introduces the concept partial mapping (CPM), section 3 mainly introduces ontology mapping based on CPM, performance evaluation will be introduced in section 4 and section 5 is a conclusion.

II. Concept Partial Mapping (CPM)

The core of this ontology mapping approach is concept partial mapping (CPM) that will be introduced detailedly in the below:

$\forall$ ontology mapping concept-pair $< A \in O_1, B \in O_2 >$, U_1 and U_2 are the respective sample spaces of the concept A and B (see Fig 1)(In the sample space, the sample instance selection relies on concrete ontology concept, sample instances should be representative and even in the distribution, the number of positive instances is basically equal to the number of negative instances).

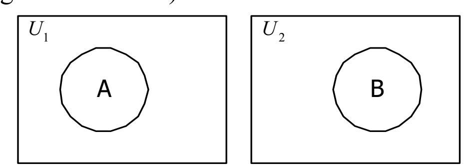

Fig 1 sample space U_1 and U_2

For implementing the mapping between these 2 concepts, sample space U_1 and U_2 overlap each other (see Fig 2).

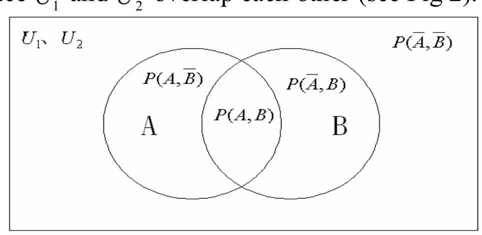

Fig 2 sample space U_1 and U_2 overlap each other

After the overlapping of the sample space U_1 and U_2, 4 domains come into being. The probabilities that random

---

* **Found Item:** Supported by National High Technology Research and Development Program of China (863) (No.: 2002AA411420); National Natural Science Foundation (60374071).

sample instances (the instances in the U_1 and U_2) fall into the 4 domains can be represented as $P(A,B), P(\overline{A},B), P(A,\overline{B}), P(\overline{A},\overline{B})$.

Distribution-based similarity measures are chose to compute the similarity of all kinds of semantic relation among different ontology concepts. According to the overlapping of sample spaces (see Fig 2), the computation formula of the "equal" relation similarity is:

$$Jaccard\_sim(A=B)=\frac{P(A\cap B)}{P(A\cup B)}=\frac{P(A,B)}{P(A,B)+P(A,\overline{B})+P(\overline{A},B)} \quad (1)$$

After making sure $A != B$, the computation formulas of the "subsumption" and "be subsumed" relation similarity are:

$$Jaccard\_sim(A\subseteq B|A\neq B)=1-\frac{P(A\cap\overline{B})}{P(A)}=\frac{P(A,B)}{P(A,B)+P(A,\overline{B})} \quad (2)$$

$$Jaccard\_sim(B\subseteq A|A\neq B)=1-\frac{P(B\cap\overline{A})}{P(B)}=\frac{P(A,B)}{P(A,B)+P(\overline{A},B)} \quad (3)$$

The range of $Jaccard\_sim$ is [0,1] which represents the value of certain semantic relation similarity (such as '=', '$\subseteq$', '$\supseteq$') among different ontology concepts.

From mentioned above, the computation of all kinds of semantic relation among ontology concepts can be transformed to the computation of the 4 probability values: $P(A,B)$, $P(\overline{A},B)$, $P(A,\overline{B})$, $P(\overline{A},\overline{B})$.

The computation of these 4 probability values will be introduced in the below:

Suppose: $N(U_i)$ is the number of instances in the sample space U_i; $N(U_i^{A,B})$ is the number of instances that belong to concept A and also belong to B; $N(U_i^{\overline{A},B})$ is the number of instances that belong to concept B and do not belong to A; $N(U_i^{A,\overline{B}})$ is the number of instances that belong to concept A and do not belong to B; $N(U_i^{\overline{A},\overline{B}})$ is the number of instances that neither belong to concept A nor B. Considering the concept A and B have the same level, the computation formulas of the $P(A,B)$, $P(\overline{A},B)$, $P(A,\overline{B})$, $P(\overline{A},\overline{B})$ are in the below:

$$P(A,B)=\frac{N(U_1^{A,B})+N(U_2^{A,B})}{N(U_1)+N(U_2)} \quad (4);$$

$$P(A,\overline{B})=\frac{N(U_1^{A,\overline{B}})+N(U_2^{A,\overline{B}})}{N(U_1)+N(U_2)} \quad (5);$$

$$P(\overline{A},B)=\frac{N(U_1^{\overline{A},B})+N(U_2^{\overline{A},B})}{N(U_1)+N(U_2)} \quad (6);$$

$$P(\overline{A},\overline{B})=\frac{N(U_1^{\overline{A},\overline{B}})+N(U_2^{\overline{A},\overline{B}})}{N(U_1)+N(U_2)} \quad (7);$$

In order to finish the overlapping of the sample space U_1 and U_2, the above formulas (4)-(7) can be computed using the $N(U_i^{A,B})$, $N(U_i^{\overline{A},B})$, $N(U_i^{A,\overline{B}})$, $N(U_i^{\overline{A},\overline{B}})$. A computation approach composed of 6 steps is in the below:

Step1: Partition U_1 into U_1^A and $U_1^{\overline{A}}$, the set of instances that do and do not belong to A respectively (see Fig 3).

Step2: Train a learner L that can identify the concept A, using U_1^A and $U_1^{\overline{A}}$ as the sets of positive and negative training instances (see Fig 3).

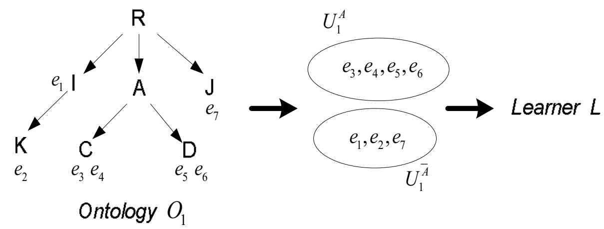

Fig 3 Step1 and Step2

Step3: Partition U_2 into U_2^B and $U_2^{\overline{B}}$, the set of instances that do and do not belong to B respectively (see Fig 4).

Step4: Apply learner L to partition U_2^B into $U_2^{A,B}$ and $U_2^{\overline{A},B}$, and partition $U_2^{\overline{B}}$ into $U_2^{A,\overline{B}}$ and $U_2^{\overline{A},\overline{B}}$ (see Fig 4).

Step5: Repeat Steps 1-4, but with the roles of sample space U_1 and U_2 being reversed, to obtain the $U_1^{A,B}$, $U_1^{\overline{A},B}$, $U_1^{A,\overline{B}}$, $U_1^{\overline{A},\overline{B}}$

Step6: Finally, compute $P(A,B)$, $P(\overline{A},B)$, $P(A,\overline{B})$, $P(\overline{A},\overline{B})$ using the formulas (4)-(7).

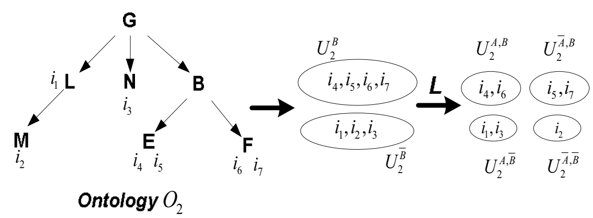

Fig 4 Step3 and Step4

III. CPM-BASED ONTOLOGY MAPPING APPROACH

Considering the sample instance is uncertain and random, the complex semantic partial relation can't be distinguished wholly using automatic machine learning. The semi-automatic ontology mapping approach which needs user's intervention is still used in the CPM-based ontology mapping approach.

General speaking, CPM-based ontology mapping approach is composed of 3 steps:(see Fig 5)

Step1：Similarity Computation

$\exists$ O_1 and O_2, for a predefined concept c of O_1, serval concepts in the ontology O_2 that are similar to the concept c are obtained in this algorithm (the value of similarity is more than Δ_k). The corresponding algorithm [6][7] is in the below:

Algorithm 1: **Similarity Computation**

Void SingleMapping(Array * pResultArray, Concept c, OntoTree T, int depth)
{
  if($\text{Similarity}(c, T.root)_{'='} > \Delta_k$)
/* Δ_k is a predefined threshold parameter such as 0.9 */
  { pResultArray→add(T.root , $\text{Similarity}(c, T.root)_{'='}$);
    For each TChild in T.root.ChildrenTrees
      SingleMapping(pResultArray, c, Tchild, depth+1);
  }

  if (($\text{Similarity}(c, T.root)_{'\subseteq'} > \Delta_k$) or ($depth \le 1$))
  { For each TChild in T.root.ChildrenTrees
      SingleMapping(pResultArray, c, Tchild, depth+1);
  }
}

See the Algorithm 1, $\text{Similarity}(c, T.root)_{'='}$ is the similarity that c is equal to T.root, and $\text{Similarity}(c, T.root)_{'\subseteq'}$ is the similarity that c is subsumed in T.root.

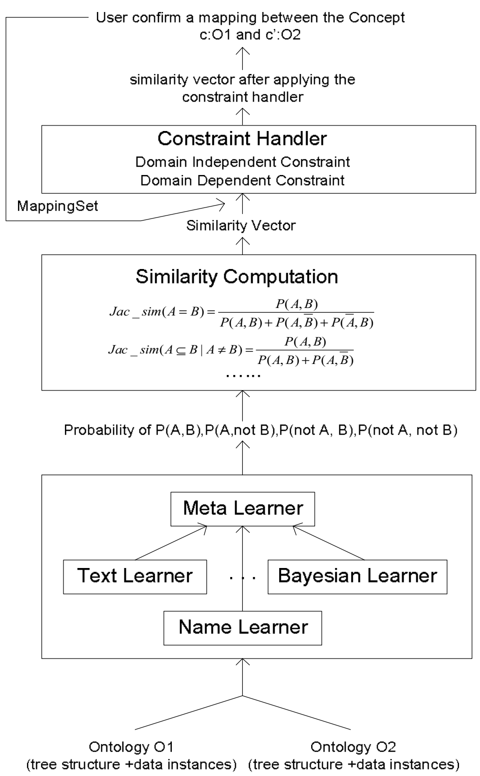

Fig 5 CPM-based ontology mapping approach

The output of the algorithm 1 is a set of concepts in another ontology that are similar to the concept c (the value of similarity is more than Δ_k). This ontology concepts set is named 'Similarity Vector' (see Fig 5).

Step2： Applying Constraints

In order to use the existing mapping results to instruct the realization of the following ontology mappings, the output of step1: 'Similarity Vector' can be modified or changed by applying related constraint relations or constraint rules on it. The constraint relations and constraint rules can be classified 2 types: Domain-independent and Domain-dependent, Domain-independent constraints convey our general knowledge about the interaction between related concepts; Domain-dependent constraints convey our knowledge about the interaction between specific concepts in the ontologies. Table I shows examples of the two types of constraints.

TABLE I
DOMAIN-INDEPENDENT AND DOMAIN-DEPENDENT CONSTRAINTS

Constraint Types		Examples
Domain-independent	**Combination**	If all children of concept A is subsumed in concept B, then A is also subsumed in B, or A matches B.
	Sibling	Two nodes match if their children also match; Two nodes match if their parents match and at least x% of their children also match; Two nodes match if their parents match and some of their descendants also match.
Domain-dependent	**Subsumption**	If node Y is a descendant of node X, and Y matches TEACHER, then it is unlikely that X matches ASSISTANT-TEACHER; If node Y is NOT a descendant of node X, and Y matches TEACHER, then it is unlikely that X matches FACULTY.
	Frequency of concept	There can be at most one node that matches UNIVERSITY-COMMITTEE.
	Adjacency	If a node in the neighborhood of node X matches ASSOCIATE-TEACHER, then the chance that X matches TEACHER is increased.

Step3: Confirming Matching

The end user chooses a most-fitted concept from 'similarity vector' after applying the constraints to finish a concept mapping, simultaneously, the concept mapping is inserted into the mapping set.

The related algorithm of step2 and step3 is in the below:

Algorithm 2: **Partial Mapping**

MappingSet partialMapping($OntoTree\ T_1$, $OntoTree\ T_2$)
{ MappingSet resultMappingSet;
  Queue temp;

```
temp.add(T1.root);
while ( NOT( temp.IsEmpty() ) )
{    Array    ResultArray, UserResultArray;
             Concept  c= temp.getfirst();
     SingleMapping( ResultArray, c, T2 ,1);
     ApplyConstraint(                     resultMappingSet,
UserResultArray, getKMax ( ResultArray ) );
//apply constraints into the K best mapping results
     Users confirm a mapping between the Concept c:T1
and the Concept c':T2;
     resultMappingSet.add( (c:T1,c':T2) );
     For each Concept cchild in c.Children
             temp.add(cchild);
}
return resultMappingSet;
}
```

The output of the algorithm is a mapping set between the two ontologies.

Ⅳ Performance Evaluation

We have evaluated the approach on several domains associated with the GLUE system. In this section, our goal is to evaluate the **matching accuracy** and **time complexity** of the approach.

Matching Accuracy

We have evaluated the matching accuracy of this approach on two applications whose characteristics are shown in Table II: (1) Matching between the English pizza ontology and the Chinese Bopomofo pizza ontology; (2) Matching between the CMS ontology in Donghua University and the CMS ontology in Jiaotong University.

TABLE II
Two Application Experiments

Ontologies		Concepts	non-leaf concepts	depth	instances	max children of a concept	confirmed mapping
Pizza	Chinese Bopomofo	43	12	4	1826	7	43
	English	46	13	4	2012	7	43
CMS	Donghua Univ	437	121	8	18557	38	236
	Jiaotong Univ	467	132	8	20447	39	236

The result of matching accuracy (GLUE, CPM without constraint, CPM with constraint) is in the below (see Fig 6, CPM with and without constraint stand for CPM-based ontology mapping approach with and without constraint, respectively).

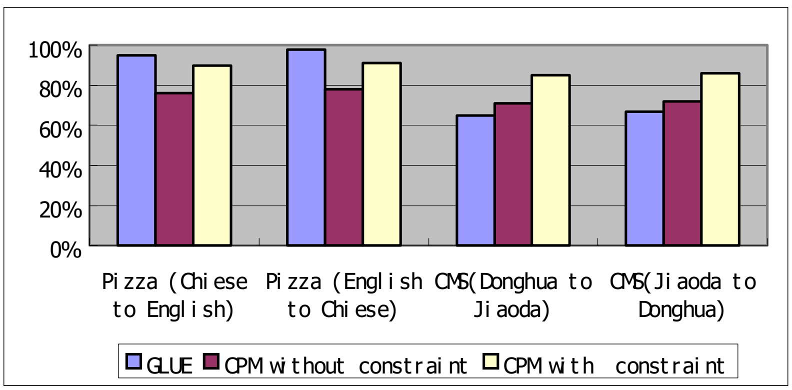

Fig 6 Matching accuracy

Time Complexity

In this section, the time complexity of GLUE and CPM-based mapping approach will be compared. For example, a mapping between the ontology O_1 that has m concepts, and the ontology O_2 that has n concepts, the mapping time of GLUE is $T_{glue} = m(nt_s + T_u)$, the time of CPM-based mapping approach is $T_{cpm} = m(\frac{K^{Log_L n}}{2} t_s + T_u)$ (t_s is the time of a single similarity computation; T_u is the time of user confirming; L is the average number of child concepts under a non-leaf concept; K is the average number of child concepts under a non-leaf concept, that have the "="or"$\supseteq$" relation with certain concept in ontology O_1.).

If ontology O_2 is a large-scale ontology, $nt_s >> T_u$, $\frac{K^{Log_L n}}{2} t_s >> T_u$, the time complexity of these 2 approaches is:

$$O(T_{glue}) = O(m \cdot n) \text{ and } O(T_{cpm}) = O(\frac{m \cdot K^{Log_L n}}{2}).$$

General speaking, $K << L$, and then $K^{Log_L n} << n$, and $O(T_{cpm}) << O(T_{glue})$.

If ontology O_2 is a lightweight ontology, $nt_s << T_u$, $\frac{K^{Log_L n} \cdot}{2} t_s << T_u$, then the time complexity of these 2 approaches is: $O(T_{glue}) = O(T_{cpm}) = O(m)$.

From the matching accuracy (see Fig6) and time complexity, we can learn that CPM-based ontology mapping approach is better than GLUE in the large-scale ontology mapping.

V CONCLUSION

In the traditional ontology mapping approach [8][9], the relation among the concepts is 'equal' or 'not equal', but in this ontology mapping approach, the mapping relation among the different ontology concepts will be extended to compound relation, including 'intersection', 'equal', 'disjunction', 'subsumption', 'be subsumed'. Concept partial mapping that can be represented as reasoning and computing the complex semantic partial relation among the different ontology concepts is put forward based on the extension, and then CPM-based ontology mapping approach also comes into being, and the mapping accuracy and efficient are improved rapidly compared to the traditional ontology mapping approach such as GLUE system.

In the end, we have done a lot of performance evaluation on the CPM-based ontology mapping approach, including mapping accuracy and time complexity, and find that this approach is very fit for the large-scale ontology mapping. This approach has been applied in two projects. One is funded by the National High Technology Research and Development Program of China (863:2002AA411420), and the other is supported by the Natural Science Funds of China (60374071).

REFERENCES

[1] AnHai Doan, Jayant Madhavan, Pedro Domingos. " Learning to Map between Ontologies on the Semantic Web". WWW 2002. May 7-11, 2002, Honolulu,Hawaii,USA.

[2] A.Doan. "Learning to Map between Structured Representations of Data". Ph.D. Dissertation, Univ. of Washington-Seattle, 2002.

[3] Erhard Rahm, Philip A.Bernstein. "A survey of approaches to automatic matching". The VLDB Journal. pp334-350, Oct, 2001.

[4] Fausto Giunchiglia, Pavel Shvaiko. "S-Match: an Algorithm and an Implementation of Semantic Matching". European Semantic Web Symposium(2004).pp61-75, May 2004.

[5] Fausto Giunchiglia, Mikalai Yatskevich. "Element Level Semantic Matching ". Technical Report # DIT-04-035. June,2004.

[6] Robin Dhamankar, Yoonkyong Lee, AnHai Doan, Alon Halevy, Pedro Domingos. "iMAP: Discovering Complex Semantic Matches between Database Schemas". SIGMOD(2004). pp13-18, June 2004.

[7] Alon Y.Halevy, Zachary G.Ives, Dan Suciu, Igor Tatarinov. "Schema mediation for large-scale semantic data sharing". VLDB Journal(2003)

[8] Alon Halevy, Oren Etzioni, AnHai Doan, Zachary Ives, Jayant Madhavan. " Crossing the structure Chasm". CIO Magazine. January 1, 1998

[9] Zachary G..Ives, Alon Halevy, Peter Mork, Igor Tatarinov. "Piazza: Mediation and Integration Infrastructure for Semantic Web Data". Journal of Web Semantics, Volume 1, Number.pp155-175, February,2004.