Journal of Southeast University (English Edition) Vol. 23, No. 3, pp. 352 – 356 Sept. 2007 ISSN 1003—7985

Ontology mapping approach using web search engine

Li Keyue[1] Xu Baowen[1,2] Wang Peng[1]

([1] School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

([2] Jiangsu Institute of Software Quality, Nanjing 210096, China)

Abstract: A new mapping approach for automated ontology mapping using web search engines (such as Google) is presented. Based on lexico-syntactic patterns, the hyponymy relationships between ontology concepts can be obtained from the web by search engines and an initial candidate mapping set consisting of ontology concept pairs is generated. According to the concept hierarchies of ontologies, a set of production rules is proposed to delete the concept pairs inconsistent with the ontology semantics from the initial candidate mapping set and add the concept pairs consistent with the ontology semantics to it. Finally, ontology mappings are chosen from the candidate mapping set automatically with a mapping select rule which is based on mutual information. Experimental results show that the F-measure can reach 75% to 100% and it can effectively accomplish the mapping between ontologies.

Key words: semantic web; ontology; ontology mapping; web search engine

Ontology mapping is an effective approach for establishing interoperation between applications using different ontologies by forming the relationships between ontology elements. Much research has been done to pursue good algorithms and tools for (semi-) automatic ontology mapping[1-2]. Though structural similarity is commonly used in ontology mapping approaches, the ontology mapping should be based on semantic information because the ontology is a knowledge representation model. However, the question of how to provide the necessary formal metadata in an effective and efficient way has still not been solved to a satisfactory extent[3]. Cimiano thinks that acquiring collective knowledge form the world wide web using a web search engine is a potential way out of the problem. Our mapping method exploits this idea for obtaining subclass relations among concept names of different ontologies, and then initializes a candidate mapping set which is a collection of mapping concept pairs. Since ontology mappings are formal relations between ontology concepts rather than ontology concept names, the results of using this web search engine method directly to generate ontology mappings is inaccurate. Therefore, we build a set of production rules to correct and complete the candidate mapping set according to concept hierarchies in ontologies. Finally, the ontology mappings are selected from the candidate mapping set automatically based on a mapping select rule.

1 Lexico-Syntactic Patterns for Hyponymy

Learning relationships between concepts through the web is the methodology based on the idea that certain lexico-syntactic patterns matched in texts convey a specific semantic relation[3]. For a pair of ontology concepts, the collection of patterns for hyponymy is called the pattern library, written as $P_{(C_i, D_j)}$. O_1 and O_2 are a pair of heterogeneous ontologies, concepts $C_i \in O_1$ and concept $D_j \in O_2$. If a pattern in $P_{(C_i, D_j)}$ is discovered in a text, we can derive that $C_i \supseteq D_j$. For example, for the sentence "Such countries as US, UK and Canada", we can extract the relationships: country $\supseteq$ US, country $\supseteq$ UK and country $\supseteq$ Canada. The following is a list of patterns for hyponymy used in our method. $N(C_i)$ indicates the name of C_i. Pattern P_1 to P_4 are proposed by Hearst[4] while the other two patterns are defined in PANKOW[3].

P_1: $N(C_i)$ such as $N(D_j)$;

P_2: such $N(C_i)$ as $N(D_j)$;

P_3: $N(C_i)$, (especially | including) $N(D_j)$;

P_4: $N(C_i)$ (and | or) other $N(D_j)$;

P_5: $N(D_j)$, an $N(C_i)$;

P_6: $N(D_j)$, is an $N(C_i)$

Patterns can not only be discovered in a text or a corpus, but also in the world wide web[3,5-7]. In our method, we make use of Google(/Yahoo) API (With Google API, users can implement the search function of Google in their own application programs.) for ob-

Received 2007-05-18.

Foundation items: The National Natural Science Foundation of China (No. 60425206, 90412003), the Foundation of Excellent Doctoral Dissertation of Southeast University (No. YBJJ0502).

Biographies: Li Keyue (1983—), female, graduate; Xu Baowen (corresponding author), male, doctor, professor, bwxu@seu.edu.cn.

taining a particular pattern and counting the times the pattern appears on the web. We call this procedure pattern validation. In our approach, when the sum of Google results of all the patterns in $P_{(C_i, D_j)}$ is greater than or equal to a threshold, the subclass relationship between C_i and D_j can be derived and this concept pair passes pattern validation.

2 Automatic Generation of Ontology Mappings

Definition 1 Suppose that O_1, O_2 are two different ontologies, the similar relationship R between their concepts is $R = \{\langle C_i, D_j\rangle \mid C_i \in O_1 \wedge D_j \in O_2 \wedge C_i \supseteq D_j \wedge \neg \exists C_m(C_m \in O_1 \wedge C_i \supseteq C_m \wedge C_m \supseteq D_j) \wedge \neg \exists D_n(D_n \in O_2 \wedge C_i \supseteq D_n \wedge D_n \supseteq D_j)\}$.

According to definition 1, we define a candidate mapping set M in definition 2.

Definition 2 Suppose that O_1, O_2 are two different ontologies. $M = \{\langle C_i, D_j\rangle \mid C_i \in O_1 \wedge D_j \in O_2 \wedge C_i \supseteq D_j\}$.

M includes the concept pairs that have passed pattern validation. We also set a weight w for every concept pair in a candidate mapping set in order to estimate the possibility of a concept pair being a mapping pair. The w of a concept pair $\langle C_i, D_j\rangle$ is written as w_{ij}. Under the initial situation, the value of w_{ij} is the sum of Google results of all the patterns in $P_{(C_i, D_j)}$.

2.1 Implicit concept

In an ontology, semantic information can often be found in definitions, labels, comments and hierarchies of concepts. Some concepts and concept hierarchies are not defined explicitly in ontologies which can be found by analyzing the comments and labels. A concept like this is called an implicit concept. Usually, the comment of a concept is the definition of the concept. In English, people are used to starting a definition with the superclass of the described concept[8]. If a clause as follows appears in the beginning of a comment of an ontology concept C, we consider noun N to be the superclass of C.

P'_1: N such as; P'_2: such N; P'_3: N, (especially | including); P'_4: N (and | or) other; P'_5: N; P'_6: an N; P'_7: is an N; P'_8: is N.

2.2 Production rules

The web search engine mapping method introduced above makes use of natural semantics of the concept names. In an ontology, a concept name usually indicates the meaning of the concept partly but not equally. The dependency between ontology concepts and concept names make it possible to generate ontology mappings by the web search engine method, and the inequality between them causes the performance of finding ontology mappings using web search engines directly to be poor. For the purpose of achieving good ontology mapping results, we propose a set of production rules based on the concept hierarchies in the ontologies. Due to the transitivity of the subclass relations between concepts, we can obtain two theorems as follows:

Theorem 1 Suppose that O_1, O_2 are a pair of ontologies, concept $C_i \in O_1$ and concept $D_j \in O_2$, if $C_i \supseteq D_j$ and $C_m \supseteq C_i$(concept $C_m \in O_1$), then $C_m \supseteq D_j$.

Theorem 2 Suppose that O_1, O_2 are a pair of ontologies, concept $C_i \in O_1$ and concept $D_j \in O_2$, if $C_i \supseteq D_j$ and $D_j \supseteq D_n$(concept $D_n \in O_2$), then $C_i \subseteq D_n$.

Let O_1, O_2 be a pair of heterogenous ontologies, M be a candidate mapping set, C_m, C_i, C_n be the concepts of O_1, and D_m, D_j, D_n be concepts of O_2. We build generation rules for concept pair $\langle C_i, D_j\rangle$ as below("$\supseteq$" and "$\subseteq$" in production rules indicate a direct subclass relation):

R_1: $C_m \supseteq C_i$, $C_i \supseteq C_n$, $\langle C_i, D_j\rangle \in M$, $\langle C_m, D_j\rangle \notin M$, $\langle C_n, D_j\rangle \in M \rightarrow M = M \vee \{\langle C_m, D_j\rangle\}$, $w_{mj} = \theta$;

R_2: $C_m \supseteq C_i$, $\langle C_i, D_j\rangle \in M$, $\langle C_m, D_j\rangle \notin M$, initial M includes the concept pairs that contain concept $C_m \rightarrow M = M - \{\langle C_i, D_j\rangle\}$;

R_3: $C_m \supseteq C_i$, $\langle C_i, D_j\rangle \in M$, $\langle C_m, D_j\rangle \notin M$, initial M excludes the concept pairs that contain concept $C_m \rightarrow M = M \vee \{\langle C_m, D_j\rangle\}$, $w_{mj} = \theta$;

R_4: $\langle C_i, D_j\rangle \in M$, $\langle C_n, D_j\rangle \in M$, $C_i \supseteq C_n \rightarrow w_{nj} = w_{nj} + w_{ij}$;

R_5: $D_m \subseteq D_j$, $D_j \subseteq D_n$, $\langle C_i, D_j\rangle \in M$, $\langle C_i, D_m\rangle \notin M$, $\langle C_i, D_n\rangle \in M \rightarrow M = M \vee \{\langle C_i, D_m\rangle\}$, $w_{im} = \theta$;

R_6: $D_m \subseteq D_j$, $< C_i, D_j > \in M$, $< C_i, D_m > \notin M$, initial M includes the concept pairs that contain concept $D_m \rightarrow M = M - \{\langle C_i, D_j\rangle\}$;

R_7: $D_m \subseteq D$, $\langle C_i, D_j\rangle \in M$, $< C_i, D_m > \notin M$, initial M excludes the concept pairs that contain concept $D_m \rightarrow M = M \vee \{< C_i, D_m >\}$, $w_{im} = \theta$;

R_8: $\langle C_i, D_j\rangle \in M$, $\langle C_i, D_n\rangle \in M$, $D_j \subseteq D_n \rightarrow w_{in} = w_{in} + w_{ij}$.

1) Reasonableness of production rules

It may happen that "$C_m \supseteq C_i$, and $\langle C_i, D_j\rangle \in M$ and $\langle C_m, D_j\rangle \notin M$" with an original candidate mapping set. This leads to a contradiction according to theorem 1. The contradiction appears under two situations: ① C_i has ambiguity and has relative semantic with D_j in natural language ($\langle C_i, D_j\rangle \in M$), but if C_i is chosen the semantic related to C_m ($C_m \supseteq C_i$) in an ontology, and C_m is unrelated to D_j($\langle C_m, D_j\rangle \notin M$). ② The concept name of D_j is an uncommon word and its patterns are hard to be found in web. In case ① we should delete

$\langle C_i, D_j \rangle$ from M because the semantic of C_i is unconcerned with D_j in the ontology. In case ②, we should add $\langle C_m, D_j \rangle$ to M because of the incorrectness of $\langle C_m, D_j \rangle \notin M$. If $C_i \supseteq C_n$ and $\langle C_n, D_j \rangle \in M$, we can obtain that the semantic of C_i is related to D_j in the ontology and this situation belongs to case ②. If a concept pair containing C_n exists in the original M, it indicates that the concept name of C_n is a normal word with natural semantics and this situation belongs to case ①. Similarly, if a concept pair containing C_n does not appear in the original M, this situation belongs to case ②. Thus, we establish three production rules R_1, R_2 and R_3. Similarly, we obtain production rules R_5 to R_7.

Due to the meanings of definition 1 and weight w, $\langle C_i, D_j \rangle \in M$, $\langle C_n, D_j \rangle \in M$ and $C_i \supseteq C_n$, w_{nj} should be greater than w_{ij}. However, under initialization w_{ij} may be greater than w_{nj} because the original value of w is from the web search engine using the natural meanings of concepts. We propose R_4 to eliminate the contradictions shown above. For the same reason, R_8 is derived.

2) Collision elimination

When R_1 and R_2 are evoked at the same time, they cause a collision. The information in M is not complete as it is missing some concepts whose names are abnormal words. Therefore, we set the priority level of R_1 higher than R_2. The priority level of R_5 is higher than R_6 for the same reason. Based on the principle that using production rules having more constraints preferentially, we set the priorities of R_1 to R_3 higher than R_4, and R_5 and R_6 higher than R_8.

3) Algorithm of using production rules

```
PD(CandidateMappingSet M)
{   for(int i=n; i>=0; i--) {//n is the number of concept pairs in M
      conceptPair P=M[i];
      R1(P); R2(P); R3(P); //use R1 to R3 for P}
      for Mc in M do {//k is the number of concept pairs in Mc
      Sort(Mc);
      for(int i=k; i>=0, i--) {conceptPair P=Mc[i];
            R5(P); R6(P); R7(P); }
      delImpliedConcept(M);
      for(int i=0; i<=n; i++) {conceptPair P=M[i]; R4(P); }
      for Mc in M do{
      for(int i=0; i<=k, i++)
      {   conceptPair P=Mc[i];
      R8(P); }
      }
}
```

Because the "$\supseteq$" and "$\subseteq$" in production rules indicate direct subclass relationships, using them in a candidate mapping set is a cyclic procedure that uses the production rules from one concept pair to another in a particular order according to the concept hierarchies of the ontology. Sort(M) sorts concept pairs in M in a top-down order of concept hierarchies in O_1. Sort(M_c) sorts the concept pairs in M_c in a bottom-up order of concept hierarchies in O_2. Here, M_c is the set of concept pairs which have the same concept C of O_1. Since the implicit concepts are not included in ontology mapping pairs, we delete concept pairs which contain an implicit concept, and add the weights of them to the concept pairs containing their direct subclass concepts.

2.3 Selecting mapping pairs

Because w indicates the occurrence of a pattern in the web, the w of concept pairs consisting of common concepts is obviously greater than the w of concept pairs consisting of uncommon concepts. Thus, an absolute high w may not represent a correct mapping, and given a particular w_{ij}, its relative value (magnitude) among all w for concept pairs including D_j and concept pairs including C_i is more important than its absolute value. We use mutual information to solve the problem of selecting mapping pairs and give the definition of the mutual information between two concept pairs[9].

Definition 3 Given a candidate mapping set M and a concept pair $\langle C_i, D_j \rangle$ in M, the estimated mutual information (EMI) between C_i and D_j is

$$\mathrm{EMI}(C_i, D_j) = \frac{w_{ij}}{\mathrm{W}} \log \frac{w_{ij}/W}{(w_i/W) \cdot (w_j/W)}$$

with w_{ij} being the weight w of concept pair $\langle C_i, D_j \rangle$, W being the sum of w of all the concept pairs in M, w_i being the sum of w of all the concept pairs that contain C_i, and w_j being the sum of w of all the concept pairs that contain D_j.

Accordingly, the mapping select rule can be given as below.

Definition 4 Let M be a candidate mapping set, M' be the set of ontology mapping pairs. $\langle C_i, D_j \rangle \in M$ ($C_i \in O_1$, $D_j \in O_2$). If M excludes a concept pair p which contains concept C_i or D_j and EMI(p) is greater than EMI(C_i, D_j), the concept pair $\langle C_i, D_j \rangle$ is a mapping pair and $M' = M' \vee \{\langle C_i, D_j \rangle\}$.

3 Algorithm

For finding mappings between different the ontologies O_1 and O_2, the complete procedure of our method has four steps: Step 1 is the process that changes the ontology concept names into their base forms, and inserts implicit concepts into ontology concept hierarchies; Step 2 is the pattern validation through which we can obtain candidate mapping set M and mapping set M'; Step 3 is processing M using production rules;

Step 4 is selecting mapping pairs from M and putting them into M'. The algorithm of our method is shown as follows:

```
Input: two different ontologies O1, O2;
Output: ontology mapping set M'.
void main()
{Ontology O1' = Preprocess (O1);
 Ontology O2' = Preprocess (O2);
 for concept Ci in O1' do {
 for concept Dj in O2' do{
 /* Put concept pair ⟨Ci, Dj⟩ into M' */
    if(N(Ci) = N(Dj)) Put(M', Ci, Dj);
    else {PatternValidate(N(Ci), N(Dj));
      if (S(P) > = θ) Put (M, Ci, Dj); }
    }
 } M = Sort(M);
 PD(M); //use production rules on M
 while(M! = ∅) Put(M', SelectMapping(M));
}
```

Particularly, we need to reduce the Google search times by avoiding unnecessary search operations. Given a concept pair $\langle C_i, D_j \rangle$, if the concepts have a same name, $C_i \supseteq D_j$ and $D_j \subseteq C_i$ can be derived using the web search engine. We consider that C_i is equal to D_j, and that $\langle C_i, D_j \rangle$ is a mapping pair. We extract these kinds of concepts surrounding the subclass concepts from the ontologies to be mapped, and establish mappings among them before doing so with the other concepts in the ontologies

4 Experiments

In this section, we perform two experiments on ontology pairs from OAEI 2005 (http://oaei.ontologymatching.org/2005. OAEI provides a public testing ontology set.) benchmark tests with $\theta = 50$ to test our method. In experiment 1, we perform three tests on the ontology pair $\langle 101, 205 \rangle$. Ontology 205 discards some linguistic features of ontology 101 by replacing local names of a concept with synonyms. The first test ignores implicit concepts and selects mapping pairs directly from initial M using the mapping generation rule. In the second test we still ignore implicit concepts, and then generate mapping pairs from M that have been processed by production rules. The last test uses our mapping method completely to find ontology mappings automatically. We use both Google and Yahoo as the web search engines for testing whether our approach is unaffected by selecting different web search engines. In experiment 2, we test our mapping approach on the ontology pairs $\langle 101, 205 \rangle$, $\langle 101, 301 \rangle$, $\langle 101, 302 \rangle$ and $\langle 101, 304 \rangle$ using Google. The ontologies 301 to 304 are four real-life ontologies of bibliographic references found on the web and left untouched.

Generally, the performance of ontology mapping is evaluated using Precision, Recall and F-measure. Some of the mappings generated by our method do not have the relation R defined in definition 1 between concepts, but have a right subclass relation. We call concept pairs like this non error mapping pairs which are helpful in forming ontology mapping manually. We give a weaken precision with respect to non error mapping pairs.

$$\text{Precision} = \frac{\text{correct\_mapping\_pairs\_in\_}M'}{\text{concept\_pairs\_in\_}M'}$$

$$\text{Recall} = \frac{\text{correct\_mapping\_pairs\_in\_}M'}{\text{existing\_mapping\_pairs}}$$

$$\text{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{WeakenPrecision} = \frac{\text{nonError\_mapping\_pairs\_in\_}M'}{\text{concept\_pairs\_in\_}M'}$$

The results of experiment 1 are shown in Tab. 1. Comparing test 1 with test 2, we find that the performance of the mapping method is much better after considering the information of the concept hierarchies. The comparison between tests 2 and 3 demonstrates that when the number of concept levels in the ontology increases, the recall and precision grow. Furthermore, the results of tests using Google and using Yahoo are similar. Thus the results of experiment 1 demonstrate that implicit concepts and production rules can improve the performance of mapping significantly and our approaches are little affected by the selection of search engine.

The results of experiment 2 in Tab. 2 show that our approaches are effective for finding good ontology

Tab. 1 Results of experiment 1 %

Tests	Web search engine	Precision	Recall	F-measure	Weaken precision
Mappings from initial M	Google	62. 5	45. 5	52. 6	66. 7
	Yahoo	54. 2	39. 4	43. 9	62. 5
Using production rules	Google	77. 3	51. 5	62. 0	77. 3
	Yahoo	72. 8	48. 5	58. 2	72. 7
Using implicit concepts and production rules	Google	91. 3	63. 6	75. 0	91. 3
	Yahoo	91. 3	63. 6	75. 0	91. 3

mapping automatically. Furthermore, the recall is low for the ontologies which contain many concepts having an uncommon name such as "Misc". For 〈101, 302〉, recall is higher than precision, because most of the concepts in ontology 302 are normal words which lead to high recall and because the deepest concept level is 2 which leads to low precision. Obviously, our method achieves a good effect on the ontologies having normal concept names and many concept levels.

Tab. 2 Results of experiment 2 %

Ontology pairs	Precision	Recall	F-measure
〈101, 205〉	91.3	63.64	75
〈101, 301〉	100	100	100
〈101, 302〉	81.82	90	85.72
〈101, 303〉	100	100	100
〈101, 304〉	100	100	100

5 Conclusion

We proposed a new ontology mapping method in this paper. This method is based on the idea that the web is a large knowledge base from which we can obtain metadata for ontologies through the web search engine. The implicit concepts and a set of production rules were recommended for correcting and completing the information obtained from the web. The experimental results demonstrate that our method is effective. Our next work is to try to obtain more information about ontologies using not only subclass relations but also ontology semantic such as ontology property to correct the information from the web.

References

[1] Kalfoglou Y, Schorlemmer M. Ontology mapping: the state of the art[J]. *The Knowledge Engineering Review*, 2003, **18**(1): 1 - 31.

[2] Shvaiko P, Euzenat J. A survey of schema-based matching approaches [J]. *Journal on Data Semantics*, 2005, **4**: 146 - 171.

[3] Cimiano P, Staab S. Learning by googling[J]. *ACM SIGKDD Explorations Newsletter*, 2004, **6**(2): 24 - 33.

[4] Hearst M A. Automatic acquisition of hyponyms from large textcorpora [C]//*Proceedings of the 14th International Conference on Computational Linguistics*. Nantes, France, 1992: 539 - 545.

[5] Cimiano P, Handschuh S, Staab S. Towards the self-annotating web[C]//*Proceedings of the 13th World Wide Web Conference*. ACM Press, 2004: 462 - 471.

[6] Cui H, Kan M-Y, Chua T-S. Unsupervised learning of soft patterns for generating denitions from online news[C]//*Proceedings of the 13th World Wide Web Conference*. ACM Press, 2004: 90 - 99.

[7] Etzioni O, Cafarella M, Downey D, et al. Web-scale information extraction in knowitall: (preliminary results) [C]//*Proceedings of the 13th World Wide Web Conference*. ACM Press, 2004: 100 - 110.

[8] van Hage W R, Katrenko S, Schreiber G. A method to combine linguistic ontology mapping techniques[C]//*Proceedings of the 4th International Semantic Web Conference*. Galway, Ireland, 2005: 732 - 744.

[9] Wang J, Wen J, Lochovsky F, et al. Instance-based schema matching for web databases by domain-specific query probing[C]//*Proceedings of the 30th International Conference on Very Large Data Bases*. Toronto: Morgan Kaufmann Press, 2004: 408 - 419.

一种利用搜索引擎实现本体映射的方法

李珂玥[1]　徐宝文[1,2]　汪　鹏[1]

([1] 东南大学计算机科学与工程学院，南京 210096)
([2] 江苏省软件质量研究所，南京 210096)

摘要：提出了一种利用 web 搜索引擎如 Google 自动完成本体映射的方法. 该方法通过构造句法模式，利用 web 搜索引擎获得异构本体概念间的上下义关系，产生由本体概念对组成的初始候选映射集. 根据本体的概念层次建立一个产生式规则集，从初始候选映射集中去除不符合本体语义的概念对，同时加入符合本体语义但未被初始候选映射集包含的概念对. 最后，按照基于互信息的映射选取规则从候选集映射集中自动产生本体映射. 实验结果表明，该方法的 F-measure 可达到 75% ~100%，能有效地完成本体之间的映射.

关键词：语义 web；本体；本体映射；web 搜索引擎
中图分类号：TP391