Journal of Southeast University (English Edition)　Vol. 23, No. 3, pp. 389 – 393　Sept. 2007　ISSN 1003—7985

Ontology mapping based on hidden Markov model

Yin Kangyin[1]　Song Zilin[1]　Xu Ping[2]

([1] Institute of Command Automation, PLA University of Science and Technology, Nanjing 210007, China)
([2] EMC Research and Measurement Center of Navy, Shanghai 200235, China)

Abstract: The existing ontology mapping methods mainly consider the structure of the ontology and the mapping precision is lower to some extent. According to statistical theory, a method which is based on the hidden Markov model is presented to establish ontology mapping. This method considers concepts as models, and attributes, relations, hierarchies, siblings and rules of the concepts as the states of the HMM, respectively. The models corresponding to the concepts are built by virtue of learning many training instances. On the basis of the best state sequence that is decided by the Viterbi algorithm and corresponding to the instance, mapping between the concepts can be established by maximum likelihood estimation. Experimental results show that this method can improve the precision of heterogeneous ontology mapping effectively.

Key words: ontology heterogeneity; ontology mapping; hidden Markov model; semantic web

Currently, although a great many web pages have been built in the world-wide web, the vast majority of them are in human-readable format only (e. g. HTML). As a consequence, software agents cannot understand and process this information, and much of the potential of the web has so far remained untapped[1]. In response, researchers have created the vision of the semantic web[2], where data has structure and ontologies describe the semantics of the data. Ontologies allow users to organize information into taxonomies of concepts, each with their own attributes, and to describe relationships between concepts. When data is marked up using ontologies, software agents can better understand the semantics and, therefore, more intelligently locate and integrate data for a wide variety of tasks.

Ontologies are playing an important role in the semantic web, however, as a result of the decentralization of the web, the heterogeneous ontologies in different domains and even in the same domain come out. Heterogeneous ontologies block the intelligence interoperation between computers and impair the validity of the interoperation. But researchers have put forward ontology mapping mechanisms to establish mapping between heterogeneous ontologies. In the past, ontology mapping was performed manually, but manual mapping is tedious work. Hence, the task of finding mapping (semi-) automatically has been an active area of research in the ontology community. Currently, the major approach to (semi-) automatic ontology mapping is based on structure, synset, semantic neighbor[3], heuristics, machine learning techniques or formal concept analysis. The anchor-PROMPT[4] system exploits the general heuristic that paths between matching elements tend to contain other matching elements. Cupid[5] implements a hybrid matching algorithm comprising linguistic and structure schema matching techniques. RiMON[6] which is based on the Bayesian decision theory treats the mapping problem as a decision problem and formalizes mapping discovery in the same way as that of risk minimization. Glue[1] uses multiple learners and a probabilistic model to perform ontology mapping.

Instances play an important role in establishing ontology mapping and hold interrelated structure information. According to statistical theory, a method which is based on the hidden Markov model (HMM)[7] is brought out to establish ontology mapping. Our method considers the attributes, relations, hierarchies, siblings and rules as the states of HMM, and HMM is built by virtue of training instances. Mapping between the concepts can be established by searching the best HMM model that the instances belong to with the Viterbi algorithm[8].

1 Primary Concept

Ontology is a formal, explicit specification of a shared conceptualization[9]. When data are expressed by an ontology, computers can understand their “meanings” and interoperate with each other. According to this definition which is accepted, in our opinion, a for-

Received 2007-05-18.
Foundation item: The Weaponry Equipment Foundation of PLA Equipment Ministry (No. 51406020105JB8103).
Biographies: YinKangyin(1979—), male, graduate; Song Zilin(corresponding author), male, professor, zilinsong@ sina. com.

mal definition concerning ontology is as follows:

Definition 1 Ontology $O = \left(\sum, C, F, R, \text{Axiom}, r, I \right)$, where $\sum$ is a symbol list, C is the set of concepts, F is the set of attributes, R is the set of relations (e. g. R: $C_1 \times C_2 \ldots C_N$), Axiom$\subset R$ is the set of axioms, $r = \bigcup_{n=2}^{\infty} 2^{R^n}$ is the set of rules where R^n is n-Cartesian products (e. g. r: $R_1, R_2, \ldots, R_{n-1} \mid = R_n$), I is the set of instances and $C \cup A \cup R \cup I \subseteq \sum^*$.

Concept C, the abstract of the individuals in the world, is the set of the individuals with the same attributes; attribute F, which is the intension of a concept, decides the differences between the concepts; R, which is the relationship between the concepts, connects the concepts to each other; Axiom is the assertion that is objective and true in the domain; rule r connects the relationships between the concepts to each other; instance I is the concrete individual in the world and the extension of the concept. In this paper, according to the formal definition of ontology, the mapping is established from five aspects: instances, attributes, relationships, hierarchies and rules.

The HMM is a finite-state automata[9] with double stochastic processes. One is the Markov chain which is expressed with transition probability and describes the state transition, and the other is a general stochastic process which is expressed with probability of observation and describes the relationships between the states and the sequences of observation.

The HMM is expressed with a 3-tuple $\lambda = (A, B, \pi)$, where $\pi = \{\pi_i\}$is the initial state distribution with $\pi_i = P\{q_1 = s_i\}, 1 \leqslant i \leqslant N, \sum_{i=1}^{N} \pi_i = 1$ and N is the number of states; $\boldsymbol{A} = \{a_{ij}\}$ is the matrix of state transition probability of being in state s_i at time t and going to state s_j at time $t+1$ with $a_{ij} = P\{q_{t+1} = s_j \mid q_t = s_i\}, 1 \leqslant i, j \leqslant N$ and $0 \leqslant a_{ij} \leqslant 1, \sum_{j=1}^{N} a_{ij} = 1; B = \{b_j(v_k)\}$ is the observation output probability distribution of state and $V = \{v_1, v_2, \ldots, v_M\}$ is the observation list, $o_t = v_k$ expresses the observation at time t.

There are three basic HMM problems:

① Given the observation sequence $O = o_1 o_2 \ldots o_T$ and an HMM model $\lambda = (A, B, \pi)$ with T being the length of the observation sequence, how do we compute the probability $P(O \mid \lambda)$ of O given the model?

② Given the observation sequence $O = o_1 o_2 \ldots o_T$ and an HMM model $\lambda = (A, B, \pi)$, how do we find the state sequence that best explains the observations?

③ Given the training data, how do we adjust the model parameter $\lambda = (A, B, \pi)$ to maximize $P(O \mid \lambda)$?

Instances are the extension of concepts, and concepts are the set of instances which have the same features. Ontology mapping based on instances judges whether the instance of one concept belongs to another concept. A concept is expressed by a model, and the model which best explains the instance corresponds to the concept the instance belongs to. When the HMM model is trained by instances, it can solve the problem of ontology mapping, and when the three basic HMM problems are solved, the ontology mapping is built. Problem ① can be solved by the forward chain algorithm or the backward chain algorithm[8]; problem ② can be solved by the Viterbi algorithm[8]; and problem ③ can be solved by the Baum-Welch algorithm[10].

2 Model Expression

Instances have labels and a set of attributes, maintain interrelationships with other instances, and are the individuals of some concept in the hierarchy. Because there are interrelationships among attributes, relations, hierarchies, rules and siblings, one of them can influence the others and vice versa. Instances are the carriers of attributes, relations, hierarchies, siblings and rules. Concepts are considered as HMM models $\lambda = (A, B, \pi)$. Attributes, relations, hierarchies, siblings and rules are considered as the states of the model $\lambda = (A, B, \pi)$. The interrelationships among attributes, relations, hierarchies, siblings and rules are considered as the state transition. Similarities between the concepts are calculated according to instance recognition.

The states are defined according to the following features:

- Attributes Attributes are the intension of concepts, decide the differences between the concepts and reflect the inherent and essential features. If one instance has the same attributes as another instance, they belong to the same concept. The similarity according to attributes is concerned with label (L), range (Range), and restriction (rest), and is defined as follows:

$$a = \text{avg}(\text{sim}(L) + \text{sim}(\text{Range}) + \text{sim}(\text{rest})) \quad (1)$$

- Relations Relations are the connections between the concepts. The connection between the instances is its extension. The similarity according to relations is concerned with labels; the neighbor instance (I), and it is defined as follows:

$$r = \text{avg}(\text{sim}(L) + \text{sim}(I)) \quad (2)$$

- Hierarchies Hierarchies decide the hierarchy among the concepts. The instance which belongs to a concept belongs to its super-concept at the same time. The similarity according to the concepts is concerned with the instances and the labels of super-concepts (sup), the instances and the labels of sub-concepts (sub), and is defined as follows:

$$c = \mathrm{avg}(\mathrm{sim}(\mathrm{sup}) + \mathrm{sim}(\mathrm{sub}) + \mathrm{sim}(I)) \qquad (3)$$

- Siblings Siblings locate in the same layer of the hierarchy. The similarity according to the siblings is concerned with instances and label, and is defined as follows:

$$\mathrm{sibling} = \mathrm{avg}(\mathrm{sim}(L) + \mathrm{sim}(I)) \qquad (4)$$

- Rules Rules are the connection between the relations. The instances that comply with the same rules belong to their corresponding concepts. The similarity according to the rules is defined as follows:

$$\mathrm{rule} = \frac{1}{n}\sum_{j=1}^{n} \mathrm{sim}(I_j) \qquad (5)$$

3 Similarity Calculation

3.1 Ontology mapping

According to the definition of the ontology mapping[11]: "Given two ontologies O_1 and O_2, mapping one ontology with another means that for each concept in ontology O_1, we try to find a corresponding concept, which has the same or similar semantics in ontology O_2 and vice versa". An ontology mapping function can be described as follows:

$$\mathrm{sim}: C_1, C_2 \rightarrow [0, 1] \qquad C_1 \in O_1; C_2 \in O_2 \qquad (6)$$

If $\mathrm{sim}(C_1, C_2) \geqslant t$ with $\mathrm{sim}(C_1, C_2)$ being the similarity coefficients between C_1, C_2, t being the threshold, then C_1, C_2 have the same or similar semantics.

When two concepts have shared instances, the more shared instances they are, the more similar they are. But when they do not have shared instances, how do we calculate the similarity between them. According to the mapping, the more instances which are mapped to another concept, the more similar they are. Therefore, the similarity between the concept C_1 in O_1 and the concept C_2 in O_2 is defined as follows:

$$p = \frac{n_2(\lambda_1) + n_1(\lambda_2)}{n_1 + n_2} \qquad (7)$$

where $n_2(\lambda_1)$ is the number of instances that belong to C_1 and is mapped to the model $\lambda_2 = (A, B, \pi)$ corresponding to C_2; $n_1(\lambda_2)$ is the number of instances that belong to C_2 and is mapped to $\lambda_1 = (A, B, \pi)$ corresponding to C_1.

3.2 Procedures of similarity calculation

Instances are the carriers of attributes, relations, hierarchies, siblings and rules. Each instance is the sample of its concept and composes an observation sequence. According to the observation sequence, the HMM model corresponding to each concept of one ontology can be trained. For any instance of the other ontology, we search its best model. In summary, we estimate the similarity of C_1 and C_2 as follows:

Step 1 Build the training model $\lambda = (A, B, \pi)$ of each concept of one ontology O_2 respectively. The initial values of the parameters A, B, π are produced at random and the final values are calculated according to the Baum-Welch[12] algorithm.

Step 2 Decide the state sequence of each instance according to the Viterbi algorithm [10].

Step 3 Transform each instance into an observation sequence $O = o_1 o_2 o_3 \ldots$ that is composed of attributes, relations, hierarchies, siblings and rules and calculate its probabilities $P(O \mid \lambda)$ according to the maximum likelihood decision rule. The concept which corresponds to $\lambda' = \arg\max_{\lambda}(P(O \mid \lambda))$ is the concept that the instance belongs to.

Step 4 Calculate the number $n_1(\lambda_2)$ of instances that belong to C_1 and are mapped to the model $\lambda_2 = (A, B, \pi)$ corresponding to C_2.

Step 5 Repeat step 1 to step 3 until the instance number $n_1(\lambda_2)$ does not change any more.

Step 6 Repeat step 1 to step 4, calculate the number $n_2(\lambda_1)$ of instances that belong to C_2 and are mapped to the model $\lambda_1 = (A, B, \pi)$ corresponding to C_1.

4 Experiments

The matching for ontologies is implemented as an important tool for aligning the ontology. This is called OMHM. In order to verify the idea of similarity measure that is proposed above, we conduct experiments and analyze its performance. The program language in the experiments is Java whose platform and VM are Eclipse and JDK1.4.1, respectively; the CPU of the system is Intel Pentium Ⅳ 2.4 GHz; the RAM is 1 GB; the operation system is Windows 2003 Server.

We have evaluated OMHM on several real-world domains. We tested the effectiveness of OMHM on EON2004 (http://co4.inrialpes.fr/align/Contest/) benchmark test cases (see Tab. 1), and compared it with the content learner of Glue on three ontologies about three institutes in one college (see Tab. 2). In the experiments and mappings are established between the reference ontology and the other ontologies, and the results are as follows from two aspects: the precision and recall of the query.

Tab. 1 EON2004 benchmark test case

Ontology	Concept	Attribute	Instance
Reference ontology	33	59	76
101	33	61	111
103	33	61	111
104	33	61	111
201	34	62	111
202	34	62	111
204	33	61	111
205	34	61	111
221	34	61	111
222	29	61	111
223	68	61	111
225	33	61	111
228	33	0	35
230	25	54	83

Tab. 2 Test data about three colleges

Taxonomies		Concept	Instance	Predicate
College Ⅰ	ICA	51	2 135	98
	ICE	47	1 912	87
College Ⅱ	ICA	51	2 135	98
	EIEC	89	3 547	132
College Ⅲ	ICE	47	1 912	87
	EIEC	89	3 547	132

4.1 Experiment Ⅰ

The experimental results are demonstrated in Fig. 1 and Fig. 2 by precisions and recalls in all the test cases, excepting a few ontologies(such as 224, 301, 302, 303, 304) which do not have instances and are not listed in the table.

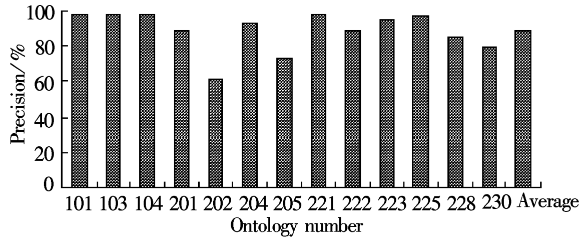

Fig. 1 Precision on several ontologies

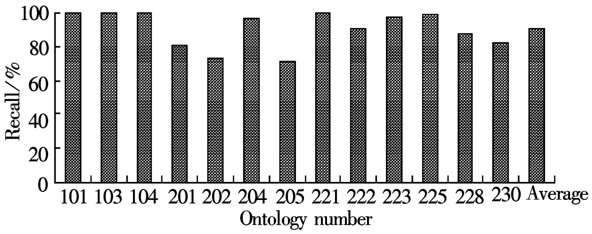

Fig. 2 Recall on several ontologies

As we can see from the results in Fig. 1 obtained by the HMM, for a majority of ontologies, the precision is higher than 80%, even up to completely matching a few ontologies. The precision is a little lower in a minority of ontologies such as 205, 230, but it is also close to 80%; only in ontology 202 is it 60%. In Fig. 2, the recall is higher than 80% in a majority of ontologies, even up to perfection in a few ontologies such as 101, 103, 104, 221, 225. Only in ontology 202, 205 is it a little lower, about 70%.

4.2 Experiment Ⅱ

The concrete information of these three institutes is as described in Tab. 2. These three ontologies characterize three institutes of one college. One is ICA, and the other two are ICE and EIEC, respectively. They describe resources (such as courses, teachers, students, papers, scientific researches, entertainments, etc.) of three institutes, where the research domain of institutes ICA and ICE are similar to some extent, and are a little different from EIEC. The taxonomies of College Ⅰ are fairly similar to each other, while the taxonomies of College Ⅱ and College Ⅲ are not very similar to each other.

As shown in Tab. 3, the precision and recall of our method are higher than those of the CL of Glue, especially regarding College Ⅱ and College Ⅲ, because our method makes full use of the interrelationships between the features. Though there are not many shared instances, the precision and recall of our method are all greater than 70%. In these three ontologies our method outperforms the CL of Glue to some extent.

Tab. 3 Test result %

Taxonomies		Precision	Recall
College Ⅰ	OMHM	95.00	92.00
	CL	85.00	94.00
College Ⅱ	OMHM	86.00	88.00
	CL	74.00	79.00
College Ⅲ	OMHM	76.00	73.00
	CL	63.00	65.00

5 Conclusion and Future Work

Our work which applies machine learning techniques to create semantic mapping is similar to Glue. But Glue uses the Naïve Bayes theory and assumes that the features of the concepts are independent of each other, while our approach which is based on the HMM and assumes that the features are interrelated with each other is closer to reality.

In this paper, attributes, relations, hierarchies, siblings and rules are represented as the states of the HMM, and the interrelationships among them are described by virtue of the state transition of the HMM. The mapping which is based on the HMM improves the precision of ontology mapping which is based on instances. This method can be combined with other strategies such as structures and reasoning to improve the precision of ontology mapping. And it is our future research work.

References

[1] Doan A, Madhavan J, Domingos P, et al. Learning to map between ontologies on the semantic web[C]//*Proceedings of the* 11*th International World Wide Web Conference*. Hawaii, 2002: 662 – 673.

[2] Berners-Lee Tim, Hendeler James, Lassila Ora. The semantic web[J]. *Scientific American*, 2001, **284**(5): 35 – 43.

[3] Rodriguez M A, Egenhofer M J. Determining semantic similarity among entity classes from different ontologies[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2003, **15**(2): 442 – 456.

[4] Noy N F, Musen M A. The PROMPT site: interactive tools for ontology merging and mapping[J]. *International Journal of Human-Computer Studies*, 2003, **59**(6): 983 – 1024.

[5] Madhavan Jayant, Bernstein Philip A, Rahm Erhard. Generic schema matching with cupid[C]//*Proceedings of the* 27*th International Conference on Very Large Databases*. Rome, Italy, 2001: 129 – 138.

[6] Tang Jie, Liang Bangyong, Li Juanzi, et al. Automatic ontology mapping in semantic web[J]. *Chinese Journal of Computers*, 2006, **29**(11): 1956 – 1976. (in Chinese)

[7] Rabiner Lawrence R. A tutorial on HMM and selected applications in speech recognition[J]. *Proceedings of the IEEE*, 1989, **77**(2): 257 – 286.

[8] Forney G D. The viterbi algorithm [J]. *Proceedings of the IEEE*, 1973, **61**(3): 268 – 278.

[9] Studer R, Benjamins V R, Fensel D. Knowledge engineering: principles and methods[J]. *Data and Knowledge Engineering*, 1998, **25**(1/2): 161 – 197.

[10] Baum L E. An inequality and associated maximization technique in statistical estimation of probabilistic function of a Markov processes [J]. *Inequalities*, 1972, **3**(1): 1 – 8.

[11] Su Xiaomeng. A text categorization perspective for ontology mapping[R]. Norway: Department of Computer and Information Science of Norwegian University of Science and Technology, 2002.

基于隐马尔可夫模型的本体映射

尹康银[1]　宋自林[1]　徐　平[2]

([1]解放军理工大学指挥自动化学院,南京 210007)
([2]海军电磁兼容研究检测中心,上海 200235)

摘要:当前本体映射方法主要考虑结构映射而且映射精度较低,根据统计理论思想,提出了一种基于隐马尔可夫模型的异构本体映射方法. 该方法将概念表示为隐马尔可夫模型、概念的特性、关系、上下文、兄弟、规则等表示为隐马尔可夫模型的状态,通过对实例的学习建立隐马尔可夫模型. 利用 Viterbi 算法确定实例所对应的状态序列,然后采用极大似然估计法确定该实例所对应的模型,从而建立异构本体之间的映射. 实验表明,该方法有效地提高了异构本体映射的精度.

关键词:本体异构;本体映射;隐马尔可夫模型;语义 web

中图分类号:TP311