Fourth International Conference on Semantics, Knowledge and Grid

Concept Similarity Matching Based on Semantic Distance

Jike Ge, Yuhui Qiu

Faculty of Computer and Information Science, Southwest University
Chongqing, China
{gjkid, yhqiu}@swu.edu.cn

***Abstract*— With the application of semantic Web service and semantic Grid service, the similarity measure between services are more and more important in the processing of service matching. By formally defining the similarity of semantic services, useful information can be obtained about their similarity and compatibility. In this paper, we propose a concept similarity matching method based on semantic distance in service matching. Uses OWL-S to descript service, the algorithm computes semantic similarity of service in four macro steps. At last, we provide an experimental comparison of our method against traditional similarity measures, and prove empirically the benefits of our approach.**

I. INTRODUCTION

With the advance of the semantic Web, both the Web and Grid community have embraced the concepts of enriching distributed systems with machine-understandable semantic metadata. Semantic services matching (in this paper, we will use the general term semantic service to describe both Web and Grid service) is one of the emerging research areas that exploits the semantic metadata to reason about the similarity and functionality of the ontologies that are to be composed. The current standard for creating semantic service description is the OWL-S (Web Ontology Language Service ontology) [1], service matching can be considered as ontological concepts matching [2].

Many diverse solutions to the matching problems have been proposed so far, see [3]-[10] for recent surveys. A good survey of ontology-based matching approaches up to 2004 is provided in [3]. A survey of schema-based and a user-centric classification of matching systems are provided in [4], while the work in [5] considers [4] and [6] as well as some other classifications. [7]-[9] are good referenced methods in semantic matching. [10] exploits external resources of a domain and common knowledge, e.g., WordNet.

In the processing of ontological concepts matching, when dealing with the similarity between concepts, it not only considers inheritance (subclass of) relations, but also considers the distance relationship between concepts. In this paper, on the basis of comprehensive consideration of the heritance relations and semantic distance between concepts, we propose a concept similarity matching method based on semantic distance. The algorithm computes semantic similarity through four macro steps, and gains the more human intuition similarity between concepts.

The rest of the paper is organized as follows. Section 2 introduces the strategy of the algorithm. It also provides an overview of four main steps of the concept similarity matching method based on semantic distance. Section 3 is devoted to the technical details of those steps. Section 4 provides the experimental comparison with some concept similarity measures. Section 5 presents our conclusions and discusses future work.

II. THE DESIGN OF ALGORITHM

In our concept similarity matching method based on semantic distance, we comprehensively consider the inheritance relations and semantic distance relations between concepts, and measure the degree of matching between concepts through semantic similarity.

The algorithm takes two concepts as input and computes a semantic similarity as output in four macro steps:

Step 1: Weight allocation. Determine weight according to the relationship between root node of ontology and other concept nodes.

Step 2: Node routing table generation. Record all paths between root node and concept nodes, and generate node routing table.

Step 3: Semantic distance computation. Compute semantic distance according to the node routing table. The semantic distance is the sum of weight between the concepts that have the inheritance relationship.

Step 4: Semantic similarity computation. Construct similarity function, and compute semantic similarity between concepts based on semantic distance.

III. CONCEPT SIMILARITY MATCHING METHOD BASED ON SEMANTIC DISTANCE

A. Weight Allocation

In the processing of similarity measure, the method of allocating the weight value to concept node has been proposed in [11] [12]. We borrow their original thought and make some modifications to reflect our intention. In our method, we allocate the weight value to the edge between concepts, but not concept nodes.

Given two concepts C_1 and C_2, we take the following formula as weight allocation function,

$$w[sub(C_1, C_2)] = 1 + \frac{1}{k^{depth(C_2)}} \quad (1)$$

Where, depth(C) presents the depth of concept C from the root concept to node C in ontology hierarchy, k is a predefined factor larger than 1 indicating the rate at which the weight

978-0-7695-3401-5/08 $25.00 © 2008 IEEE
DOI 10.1109/SKG.2008.24

values decrease along the ontology hierarchy (currently we set *k* to 2). This formula has two desirable properties: (1) the semantic differences between upper level concepts are higher than those between lower level concepts, in other words, two general concepts are less similar than two specialized ones. (2) The distance between sibling concepts is greater than the distance between parent and child concepts. Specially, the depth of root concept is zero, and the depth of other concepts equal to their path length to root concept node. Fig. 1 shows the weight values of ontological concepts.

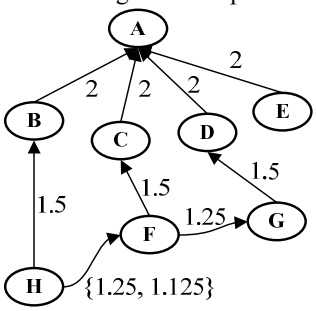

Fig. 1 The weight values of ontological concepts

In addition, if there exists multiple inheritance relation between concepts (such as, F is subclass of C and G), the depth of the concept node have multiple values, and the weight also have multiple values. For example, there are two weight values between H and F, for path (H, F, C, A), w[sub(H, F)] = 1+ $1/2^2$ = 1.25; for path (H, F, G, D, A), w[sub(H,F)] = 1+ $1/2^3$ = 1.125.

B. Node Routing Table Generation

To each concept node in ontology, we can get its all paths to root node, and can compute related weight values between nodes in every path, and we can use paths and weight values to generate the node routing table. We can compute and compare the length of path between two nodes through searching the node routing table. It is convenient to obtain the shortest semantic distance not traversing the whole ontology.

The weight construction method as follows: Breadth traversal the ontological concept graph from root node, the path record of child node is composed of the parent node's path record and itself, the child's weight is composed of the parent node's weight list and the weight between the parent-child nodes.

The node routing table of Fig. 1 shows as Table I.

TABLE I
THE NODE ROUTING TABLE

Nodes	Routing
A	root node
B	{(B,A)(2)}
C	{(C,A)(2)}
D	{(D,A)(2)}
E	{(E,A)(2)}
F	{(F,C,A)(1.5, 2)} {(F,G,D,A)(1.25, 1.5, 2)}
G	{(G,D,A)(1.5, 2)}
H	{(H,B,A)(1.5, 2)} {(H,F,C,A)(1.25, 1.5, 2)} {(H,F,G,D,A)(1.125, 1.25, 1.5, 2)}

Where, all routings of every node to root node are composed of some path records, and every path record include node path and related weight values. The node path is the path from current node to root node, and the weight value is weight between two nodes in the path.

When compute the weight values between indirect concept nodes, we only need to compute the sum of weight between two nodes in node routing table. For example, the weighted value of (G, A) is 1.5+2 =3.5.

C. Semantic Distance Computation

We can compute the semantic distance among any two nodes through the node routing table. Fig. 2 presents the pseudo-code of semantic distance solving algorithm.

Semantic Distance Solving Algorithm:

Input: two concepts C_1, C_2

Output: semantic distance: Sem_Dis(C_1,C_2)

For two ontological concepts: C_1, C_2

If C_1, C_2 are the same concept

Sem_Dis(C_1, C_2)=0

Else if there exists the direct path relation between C_1 and C_2

Sem_Dis(C_1, C_2) = w[sub(C_1, C_2)]

Else if there exists the indirect path relations between C_1 and C_2

$$Sem\_Dis(C_1,C_2) = \sum_{C \in SPath(C_1,C_2)} w_c[sub(C_1,C_2)]$$

Where, *SPatch* denotes the shortest path between C_1 and C_2

Else

Sem_Dis(C_1, C_2) = min{Sem_Dis(C_1,C_0)} + min{Sem_Dis(C_2,C_0)}

Fig. 2 Semantic distance solving algorithm

We can easily compute semantic distance between two ontological concept nodes through the semantic distance solving algorithm. For example, if we want to compute the semantic distance between B and F, then, Sem_Dis(B,F) = min{Sem_Dis(B,A)}+min{Sem_Dis(F,A)} = 2 + (2+1.5) = 5.5, therefore, the semantic distance between B and F is 5.5.

D. Semantic Similarity Computation

The greater semantic distance between two concepts, the less semantic similarity of them is. That is, semantic similarity and semantic distance have inverse relation [10]. But, the range of semantic distance is too great to measure intuitive semantic relation between concepts, and it is also a non-normalized description for semantic relation.

For above reasons, we exploit semantic similarity when measuring the semantic relation between concepts. It needs to construct a logical semantic similarity function after obtaining the semantic distances, and then the semantic similarity function can convert semantic distance to semantic similarity.

The semantic similarity function has some properties that confirm some of the common intuition regarding similarity.

1. $0 \leqslant \text{sim} \leqslant 1$
2. $\forall a : sim(a,a) = 1$
3. $\forall a,b,c$: if Sem_Dis(a, b)>Sem_Dis(a, c), then sim(a, b)<sim(a, c)

Property 1 gives the range of semantic similarity function. For identical object *a* and *b*, their similarity value is one. When two objects have nothing in common, their similarity value is zero. In other words, the output of similarity function should be in closed interval [0, 1].

Property 2 states that the semantic similarity function is reflexive. This follows the intuition that any object should be identical to itself.

Property 3 shows the relationship between semantic distance and semantic similarity. For any object *a, b* and *c*, if the semantic distance between *a* and *b* is more than that of *a* and *c*, the semantic similarity between *a* and *b* is less than that of *a* and *c*.

We can immediately obtain some referenced semantic similarity function according to above properties, such as:

$$SF1 = 1/(Sem\_Dis+1) \quad (2)$$

$$SF2 = 1/(Sem\_Dis^2+1) \quad (3)$$

$$SF3 = 1/e^{Sem\_Dis} \quad (4)$$

The mainly difference among these functions is that they have different rate of descent with the distance increasing.

SF1 is reciprocal descending function. It is a linear descending with the distance increasing.

SF2 is square descending function. It is an accelerating descending with the distance increasing.

SF3 is exponential descending function. It is a fast accelerating descending with the distance increasing.

The distinction of these function performance is less when Sem_Dis is small. But, with the increasing of Sem_Dis, the latter two functions will be accelerating attenuation. SF2 and SF3 are not well fit for measuring semantic similarity under the condition of multiple hierarchy ontology.

In this paper, by comprehensive consideration, we take $SF=1/(p*Sem\_Dis+1)$ $(0<p\leqslant 1)$ as our semantic similarity function. Where, *p* and Sem_Dis decide the impact degree of semantic distance to semantic similarity. The concrete value of *p* obtains through experiment or related domain experts.

IV. Experimental Evaluation

A. Evaluation Set-up

In this experimental evaluation, we are going to find the semantic similarity among **author, writer, creator, illustrator** and **person**. Consulting WordNet, we get the fragment of the ontology hierarchy concerning these concepts shows in Fig. 3.

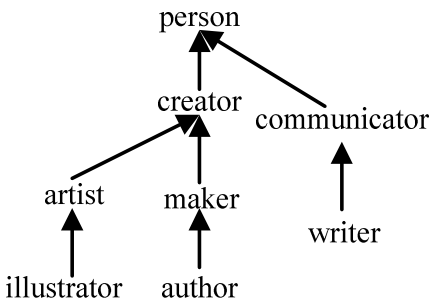

Fig. 3 The fragment of the ontology hierarchy

In this experimental comparison, we take SF=1/(0.2* Sem_Dis+1) as semantic similarity function, because it is pervasive in many conditions and can gain the valuable results than other two functions.

B. Evaluation Results

We present the performance and quality evaluation of our proposed method and other three similarity measures. The evaluation results shows as Fig. 4.

	illustrator	author	creator	person	writer
illustrator	1.0	0.0	0.0	0.0	0.0
author	0.0	1.0	0.0	0.0	0.0
creator	0.0	0.0	1.0	0.0	0.0
person	0.0	0.0	0.0	1.0	0.0
writer	0.0	0.0	0.0	0.0	1.0

(a) Synonymy similarity

	illustrator	author	creator	person	writer
illustrator	1.0	0.05	0.07	0.0	0.02
author	0.05	1.0	0.0	0.0	0.19
creator	0.07	0.0	1.0	0.06	0.02
person	0.0	0.0	0.06	1.0	0.04
writer	0.02	0.19	0.02	0.04	1.0

(b) Gloss overlap

	illustrator	author	creator	person	writer
illustrator	1.0	0.37	0.43	0.4	0.18
author	0.37	1.0	0.43	0.29	0.36
creator	0.43	0.43	1.0	0.4	0.18
person	0.4	0.29	0.4	1.0	0.25
writer	0.18	0.36	0.18	0.25	1.0

(c) Upward cotopic similarity

	illustrator	author	creator	person	writer
illustrator	1.0	0.48	0.65	0.51	0.38
author	0.48	1.0	0.65	0.51	0.38
creator	0.65	0.65	1.0	0.71	0.48
person	0.51	0.51	0.71	1.0	0.59
writer	0.38	0.38	0.48	0.59	1.0

(d) The proposed method

Fig. 4 The results of various similarity measures

In Fig. 4, (a) is the results of synonymy similarity [13], (b) is the results of gloss overlap [14], (c) is the results of upward cotopic similarity [15], and (d) is the results of our proposed method: concept similarity measure based on semantic distance.

As shows in Fig. 4, synonymy similarity measure can only find the similarity between the same concepts, and gloss overlap measure is better than the synonymy similarity measure. The upward cotopic similarity measure and our proposed method are better than the above two methods. The upward cotopic similarity measure can find the semantic similarity between concepts, but the similarity score is low. Our proposed method can also get the semantic similarity between concepts, and the similarity score is high.

C. Evaluation Summary

1) *Performance Measures:* Time is an important indicator, because when matching industrial-size ontologies (e.g., with hundreds and thousands of nodes, which is quite typical for e-business applications), it shows scalability properties of our proposed method and its potential to become industrial-strength systems, because it is easy to obtain the weight of ontological concepts and the node routing table. The computation complexity of our method is constrained to be polynomial.

2) *Quality Measures:* Most similarity matching systems return similarity coefficients, rather than semantic relations, and our proposed method is similar to them.

It is not exist that the absolutely precise criteria for measuring the semantic similarity between concepts, generally speaking, the larger similarity scores between concepts, the more semantic similarity. In addition, the experience of related domain experts is also a kind of recommended criterion.

In order to find how well our measure matching human intuition, we performed user studies on different people, all of whom are doctoral students, we considered them are all experts. Their major include philosophy, linguistics, artificial intelligence and information management. 85% of 20 accessed experts considered that our results closer to human intuition than other three methods.

V. Conclusions

In this paper, we present a concept similarity matching method based on semantic distance. It considers not only the inheritance relation between concepts, but also the level of concepts in ontology hierarchy. We conducted a comparative evaluation of our approach against three state of the art methods. The results empirically prove the strength of our approach.

Future work includes development of a robust semantic matching system. Also, we are planning to extend the semantic matching approach by computing the semantic distance among different ontologies, which might be more useful when different ontologies encode a domain of interest at different level of details. Developing a testing methodology which is able to estimate quality of the matching between ontologies with hundreds and thousands of nodes is also an interesting work. Here, the key issue is that in these cases, specifying expert matching manually is neither desirable nor feasible task, thus a semiautomatic approach is needed.

In addition, the Resource Space Model (RSM) is a semantic model for specifying, organizing and retrieving versatile resources such as image, text, webpage and link by classifying their contents according to different partition methods, organizing them into a multi-dimensional classification space, and normalizing the resource space for effective management [16] [17]. The semantic similarity matching between the resource space models or the semantic similarity matching between the resource space model and ontology, which are more interesting and researchable work, they are our next work.

Acknowledgment

This work is supported by the National Grand Fundamental Research 973 Program of China under Grant No.2003CB317008. The author thanks all team members of Semantic Grid Research Group of Southwest University for their help and cooperation.

References

[1] M. Burstein, J. Hobbs, O. Lassila. et al. (2002) "OWL-S: Semantic Markup for Web Services," [Online]. Available: http://www.w3.org/Submission/OWL-S/.

[2] S. Melnik, H. Garcia-Molina and E. Rahm, "Similarity flooding: a versatile graph matching algorithm and its application to schema matching," *Proceedings of 18th International Conference on Data Engineering*, pp. 117-128, 2002.

[3] N. Noy, "Semantic Integration: A survey of ontology-based approaches," *SIGMOD Record*, vol. 33, no. 4, pp. 65-70, 2004.

[4] P. Shvaiko and J. Euzenat, "A survery of schema-based matching approaches," *Journal on Data Semantics (IV)*, Springer, pp. 146-171, 2005.

[5] J. Euzenat and P. Shvaiko. *Ontology matching*, Berlin, Germany: Springer-Verlag, 2007.

[6] F. Giunchiglia, M. Yatskevich and P. Shvaiko, "Semantic Matching: Algorithms and Implementation," *Journal on Data Semantics (IX)*, Springer, pp.1-38, 2007.

[7] M. A. Rodriguez and M. J. Egenhofer, "Determining Semantic Similarity among Entity Classes from Different Ontologies," *IEEE Transactions on knowledge and data engineering*, vol. 15, no. 2, pp. 442-456, 2003.

[8] L. Kuang, J. Wu, S. Deng, Y. Li, W. Shi and Z. Wu, "Exploring Semantic Technologies in Service Matchmaking," *Third European Conference on Web Services (ECOWS'05)* , pp. 226-234, 2005.

[9] J. Hau, W. Lee and J. Darlington, "A Semantic Similarity Measure for Semantic Web Services," *In Workshop of WWW2005, Web Service Semantics: Towards Dynamic Business Integration*, 2005.

[10] A. Budanitsky and G. Hirst, "Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures," *In Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, 2001.

[11] J. Zhong, H. Zhu, J. Li and Y. Yu, "Conceptual graph matching for semantic search," *The 2002 International Conference on Computational Science (ICCS2002)*, Amsterdam, pp. 92-106, 2002.

[12] Y. Ganjisaffar, H. Abolhassani, M. Neshati and M. Jamali, "A Similarity Measure for OWL-S Annotated Web Services," *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 621-624, 2006.

[13] F. Giunchiglia, P. Shvaiko and M. Yatskevich, "S-Match: an algorithm and an implementation of semantic matching," *Proceedings of 1st European Semantic Web Symposium (ESWS),* volume 3053 of LNCS, Springer, pp. 61-75, 2004.

[14] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," *Proceedings of 5th Annual International Conference on Systems Documentation (SIGDOC)*, Toronto (CA), pp. 24-26, 1986.

[15] A. Madche and V. Zacharias, "Clustering ontology-based metadata in the semantic web," *Proceedings of 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pp. 348-360, 2002.

[16] H. Zhuge, "Resource Space Grid: Model, Method and Platform," *Concurrency and Computation: Practice and Experience*, vol. 16, no. 14, pp. 1385-1413, 2004.

[17] H. Zhuge, *The Web Resource Space Model,* Berlin, Germany: Springer-Verlag, 2007