Journal of Southeast University (English Edition) Vol. 26, No. 1, pp. 48 - 52 Mar. 2010 ISSN 1003—7985

Reducing semantic bias of annotations for semantic web service discovery

Wei Dengping[1] Wang Ting[1] Tang Jintao[1] Wang Ji[2]

([1]School of Computer, National University of Defense Technology, Changsha 410073, China)

([2]National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073, China)

Abstract: In order to improve the effectiveness of semantic web service discovery, the semantic bias between an interface parameter and an annotation is reduced by extracting semantic restrictions for the annotation from the description context and generating refined semantic annotations, and then the semantics of the web service is refined. These restrictions are dynamically extracted from the parsing tree of the description text, with the guide of the restriction template extracted from the ontology definition. New semantic annotations are then generated by combining the original concept with the restrictions and represented via refined concept expressions. In addition, a novel semantic similarity measure for refined concept expressions is proposed for semantic web service discovery. Experimental results show that the matchmaker based on this method can improve the average precision of discovery and exhibit low computational complexity. Reducing the semantic bias by utilizing restriction information of annotations can refine the semantics of the web service and improve the discovery effectiveness.

Key words: semantic web service discovery; semantic bias; context; restriction template; similarity measure

Most of the current discovery methods mainly consider the semantic matching based on a service profile rather than the service process model. A SWS profile describes the service capabilities in terms of several elements, including its inputs(I), outputs(O), preconditions/assumptions (P) and effects/postconditions(E)[1]. There are also various SWS matchmakers based on their respective profile elements: some perform logic-based semantic IOPE matching[2-3], and some others perform logic-based semantic service signature (input/output) matching[4-5]. Most of these methods decide the semantic similarity between two concepts through their semantic subsumption relationships described in the domain ontology. Generally, two resources with the same semantic annotations are considered to have the same semantics. However, sometimes there exists a semantic bias between the semantics of the annotation and the semantics that the annotated resource represents; i. e., the annotations are not accurate enough to represent the semantics of the annotated resources. This bias may make the matching results not reliable and seriously affect the discovery effectiveness. With the rapidly increasing number of semantic web services, such situations occur more and more frequently.

The key idea of this paper is to reduce the semantic bias by replacing the annotation with a concept expression which has finer semantics. The refined concept expression is dynamically extracted from the annotated context with the guide of the restriction template which is more general than the constraint type defined in Ref. [6] and it is directly obtained from the definition of the domain ontology. However, current subsumption-relationships-based similarity measures are not suitable for computing the similarity between such concept expressions. This paper also proposes a novel semantic similarity measure to compute the semantic similarity between two concept expressions.

1 Annotated Context

1.1 Semantic bias of semantic annotations

The semantic annotating task in the semantic web community can be presented by a process of establishing a function SA: $R \rightarrow C$, in which R is the set of resources to be annotated and C is the set of concepts in the domain ontology. $SA(r) = c(r \in R, c \in C)$ means that resource r is annotated with concept c.

When the web resources are annotated with the concepts in the ontology, the semantics of each resource can be considered as the same as the semantics that the corresponding concept represents.

The semantic bias of semantic annotations states the fact that the semantics of an annotated resource is inconsistent with the semantics of the annotation, or the semantics of the annotation cannot perfectly describe the semantics of the resource. From this point of view, most current annotated resources may have a more or less semantic deviation to their real-world semantics.

1.2 Annotated context

In the similarity measure community, context is often defined as any information that helps to specify the similarity between two entities more precisely by concerning the current situation[7]. This definition is especially useful for the similarity measurement and also supports the choice and weighting of context parameters. However, the annotated context in this paper is a bit different from the usual definition of the context in the similarity measure community.

Annotated context is defined as any information of the annotated resources that helps to clarify or refine the semantics of an annotation.

In the SWS community, the same concept c involved in

Received 2009-07-30.

Biographies: Wei Dengping (1981—), female, graduate; Wang Ting (corresponding author), male, doctor, professor, tingwang@nudt.edu.cn.

Foundation items: The National Basic Research Program of China (973 Program) (No. 2005CB321802), Program for New Century Excellent Talents in University (No. NCET-06-0926), the National Natural Science Foundation of China (No. 60403050, 90612009).

Citation: Wei Dengping, Wang Ting, Tang Jintao, et al. Reducing semantic bias of annotations for semantic web service discovery[J]. Journal of Southeast University (English Edition), 2010, 26(1): 48 - 52.

different SWS may represent the instances with different semantics, and all the information about the SWS such as the description text and the source code are all considered as the annotated context of c.

Example 1 Both web services A and B have an input parameter annotated with a concept Book and an output parameter annotated with a concept Price (a slice of the Book ontology is described in Fig. 1). Service A returns the price of the books which are published by Springer, and B returns the price of the books which are published by Elsevier. The same concept Book abstractly represents all the books published by Springer in service A, while it represents all the books published by Elsevier in service B.

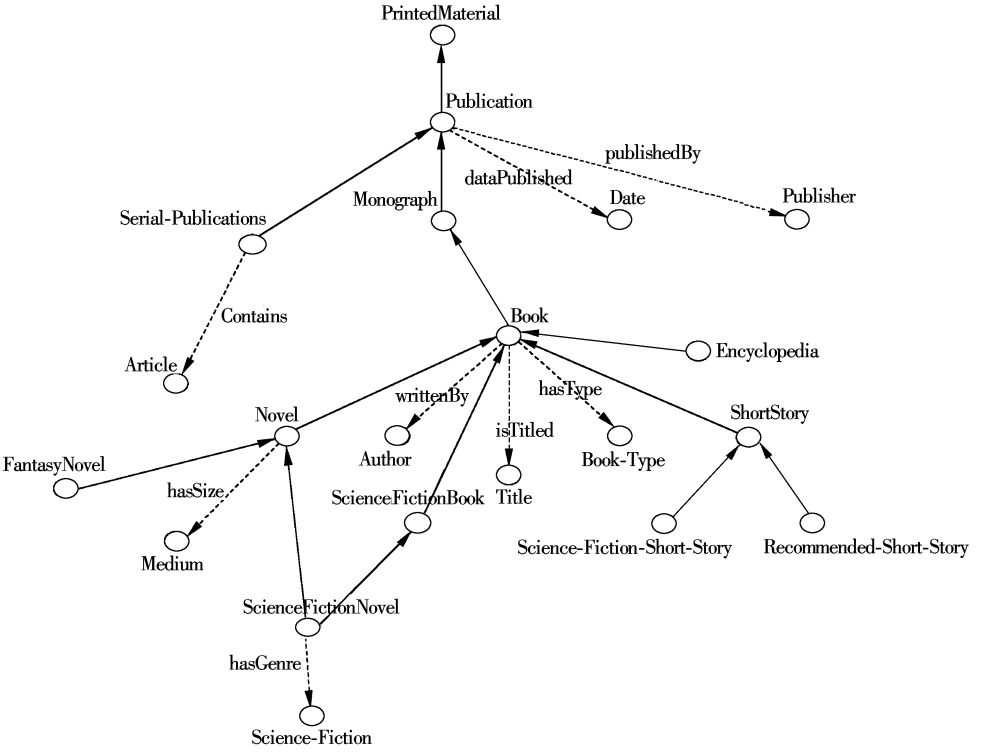

Fig. 1 The snippet of Book ontology

From the above example, we can easily see that service A can only accept the books published by Springer and return the corresponding prices, while service B can only accept those published by Elsevier and return their prices. They have the same interface definition but with different semantics. The main reason causing the bias is that the annotations are actually more general than the parameters of the web services should be; i. e., all the instances that services A and B can accept are subsumed by the instances of the concept Book.

1.3 Generating refined concept expressions

1.3.1 Restriction template

Let T be the terminology of the ontology o specified in OWL-Lite(SHIF(D)) or OWL-DL(SHOIN(D))[8]; SC(c) be the set of super-concepts of c; R be the set of all the properties specified in the ontology o. The restriction template of concept c annotated in a specific web service can be obtained from the ontology definition and represented by the following set:

$$\mathrm{RT}(c) = \{r \mid r \in R \wedge (c \in r.\,\mathrm{domain} \vee \exists c'(c' \in \mathrm{SC}(c) \wedge c' \in r.\,\mathrm{domain}))\} \quad (1)$$

in which each r is also called a template element of concept c. RT(c) represents all the properties whose domains include the concept c or its super-concepts.

Example 2 The Book ontology in Fig. 1 has several object properties such as publishedBy and data properties such as datePublished. Then some restriction templates are listed as follows:

RT("Book") = {datePublished; publishedBy; writtenBy; isTitled; hasType}

RT("Monograph") = {datePublished; publishedBy}

1.3.2 Generating refined concept expressions

Each concept c has its own restriction template according to the definition of the ontology it belongs to. When a parameter of the web service is annotated with concept c, the semantics of concept c in this service needs to be refined (confirmed) according to the service context and the restriction template. Actually, each template element in the restriction template corresponds to a constraint. If the range of the template element has a certain value, then the instances of the concept are filtered by this constraint.

In order to formally represent some complex descriptions, we use the notions of description logics[8]. Let ξ be the set of concepts; R be the set of roles. The semantics of a concept description is defined in terms of an interpretation $I = (\Delta^I, \cdot^I)$, which consists of a nonempty set Δ^I, i. e., the domain of the interpretation, and an interpretation function $\cdot^I$, which associates to each concept name $c \in \xi$ a subset c^I of Δ^I and to each role name $R \in \mathbf{R}$ a binary relation $R^I \in \Delta^I \times \Delta^I$.

Definition 1 (concept expression) A concept expres-

sion is a complex description that is built using primitive concepts, conjunction constructor, roles and the value restriction $\forall R.c$, in which $\forall R.c$ is interpreted as the set $\{x \in \Delta^I \mid \forall y \in \Delta^I((x, y)\ R^I \rightarrow y \in c^I)\}$.

Example 3 The concept expression Publication $\sqcap$ $\forall$ publishedBy.{Springer} represents all the instances of publications which are published by Springer. It is a sub-concept of the concept Publication.

The method for extracting the instances of the restriction template is based on the parse trees of the description text which is considered as the annotated context. It improves the semantic constraints extraction method that can only extract limited kinds of restrictions defined by users, which has been described in our previous work[6]. For each concept c, we can obtain several triples according to syntactic relationships in parse trees, each of which includes a subject representing concept c, a predicate and an object. Then, the similarity between the predicate name and the property name in each restriction template element is computed. If the similarity is over the threshold, then the corresponding object value is extracted as the value restriction about the property, i.e., the instance of the template element. Therefore, the method here can extract much more definite restriction information about a concept with the guide of the restriction template and also does not need the specific extracting heuristic rules. Algorithm 1 shows the pseudo-code of the process of generating refined concept expressions according to the annotated context and the restriction template.

Algorithm 1 Refining(c, ws) returns the refined concept expression c' by extracting instances of the restriction template of concept c from the annotated context ws.

$c' = c$

Obtain all the possible triples $T = \{\langle c, p_i, o_i \rangle\}$ about the concept c from the description text

for each template element $\forall r.D \in \mathrm{RT}(c)$ do

  for each triple $t \in T$ do

    if $\mathrm{Sim}(r, p_i) >= \mathrm{threshold}$ then $c' = c' \sqcap \forall r.\{o_i\}$

  end for

end for

1.3.3 Semantic web service with refined semantics

We define an SWS ws as a tuple $\mathrm{ws} = \langle I, O, P, E \rangle$, in which $I = \{i_1, i_2, \ldots, i_m\}$ represents the set of concepts that is annotated to the input parameters; $O = \{o_1, o_2, \ldots, o_n\}$ represents the set of concepts that is annotated to the output parameters; P and E represent the preconditions and effects of the service ws, respectively.

After refining the semantics of each concept annotated in the ws with the help of the restriction template, ws is represented as $\mathrm{ws}_r = \langle I', O', P', E' \rangle$, in which

$$
\begin{aligned}
I' &= \{\mathrm{Refining}(i_1, \mathrm{ws}), \mathrm{Refining}(i_2, \mathrm{ws}), \\
&\quad \ldots, \mathrm{Refining}(i_m, \mathrm{ws})\} \\
O' &= \{\mathrm{Refining}(o_1, \mathrm{ws}), \mathrm{Refining}(o_2, \mathrm{ws}), \\
&\quad \ldots, \mathrm{Refining}(o_n, \mathrm{ws})\} \\
P' &= \mathrm{Sub}(P, \mathrm{ws}) \\
E' &= \mathrm{Sub}(E, \mathrm{ws})
\end{aligned}
$$

where $\mathrm{Sub}(x, \mathrm{ws})$ represents the refined logic expression x' of x, which is obtained by substituting each concept c in x with $\mathrm{Refining}(c, \mathrm{ws})$. In this paper, we only consider the similarity measure for data semantics, i.e., we ignore the preconditions and effects.

2 SWS Discovery Based on Refined Concept Expressions

2.1 Semantic similarity measure for SWS discovery

Currently, most matchmakers for discovering the SWS decide the degree of matching according to the subsumption relationships between two concepts in domain ontologies. After refining the semantics of a concept c in different web services, different refined concept expressions are generated, which are the sub-concepts of concept c. Subsumption relationships usually cannot distinguish among these new concepts. Thus, it is desirable to design a novel semantic similarity measure for the SWS described in the precious section.

Definition 2 (semantic similarity) Let c^I be the set of instances of concept c. Given request concept R and service concept S, the semantic similarity between R and S is the degree that concept S satisfies concept R, which is defined as a function $\mathrm{Sim}: C \times C \rightarrow [0, 1]$,

$$\mathrm{Sim}(R, S) = \frac{|R^I \cap S^I|}{|S^I|} = \frac{|(R \sqcap S)^I|}{|S^I|} \tag{2}$$

Given a user request r and a semantic web service ws, we can obtain a new representation of request $r = \langle I_r, O_r \rangle$ and the web service $\mathrm{ws} = \langle I_{ws}, O_{ws} \rangle$ by refining the semantics of their parameters using the restriction templates and the annotated context. The similarity between r and ws can be computed as

$$\mathrm{Similarity}(r, \mathrm{ws}) = \alpha \mathrm{Sim}_s(I_{ws}, I_r) + \beta \mathrm{Sim}_s(O_r, O_{ws}) \tag{3}$$

where $\alpha + \beta = 1$, $0 \leqslant \alpha \leqslant 1$, and $\mathrm{Sim}_s(X, Y)$ means the degree that Y satisfies X, and it is defined as

$$\mathrm{Sim}_s(X, Y) = \sum_{x \in X} \max_{y \in Y} \frac{\mathrm{Sim}(x, y)}{m} \tag{4}$$

where m is the cardinality of set X.

2.2 Estimating semantic similarity of two concept expressions

Usually, it is difficult to compute all the instances of a concept. We can only estimate the rate between the numerator set and the denominator set in Eq. (2).

Let the conjunction concept expression of R and S be $I = c_1 \sqcap \forall P_1{}^I D_1{}^I \sqcap \ldots \sqcap \forall P_k{}^I D_k{}^I$. The estimation of the semantic similarity is defined as

$$
\begin{aligned}
\mathrm{Sim}(R, S) = \frac{|I^I|}{|S^I|} &\approx \mathrm{Sim}_e(I, S) = \\
&\sum_{i}^{k} \mathrm{Sim}_R(\forall P_i D_i, S) + \frac{\mathrm{Sim}_C(c_1, c_s)}{k+1}
\end{aligned} \tag{5}
$$

where $\mathrm{Sim}_e(I, S)$ represents the estimated similarity between I and S; $\mathrm{Sim}_C(c_1, c_s)$ represents the similarity between two concepts c_1 and c_s; $\mathrm{Sim}_R(\forall P_i D_i, S)$ represents

the semantic similarity between value restriction $\forall P_i D_i$ and S. There are several methods available to measure the semantic similarity between two concepts. This paper uses the loss-of-information similarity measure for unfolded conception expressions[9] to measure the semantic similarity between two concepts c_1 and c_s. There exist two situations when computing the value of $\mathrm{Sim}_R(\forall P_i D_i, S)$.

1) If there does not exist restriction $\forall P'_j D'_j$ in S such that P_i matches P'_j, let concept V be the range of property P_i and V^I be the set of individuals defined in ontology. Then $\mathrm{Sim}_R(\forall P_i D_i, S)$ is defined as

$$\mathrm{Sim}_R(\forall P_i D_i, \forall P'_j D'_j) = \frac{|D_i|}{|V^I|} \quad (6)$$

2) If there exists value restriction $\forall P'_j D'_j$ in S such that P_i matches P'_j, then $\mathrm{Sim}_R(\forall P_i D_i, S)$ is defined as

$$\mathrm{Sim}_R(\forall P_i D_i, \forall P'_j D'_j) = \begin{cases} \sqrt{\mathrm{Sim}_P(P_i, P'_j)\,\mathrm{Sim}_c(D_i, D'_j)} \\ \sqrt{\mathrm{Sim}_P(P_i, P'_j)\,\mathrm{Sim}_d(D_i, D'_j)} \end{cases} \quad (7)$$

where $\mathrm{Sim}_P(P_i, P'_j)$ represents the semantic similarity between two properties P_i and P'_j, which is decided by their semantic relationships defined in the ontology; $\mathrm{Sim}_c(D_i, D'_j)$ represents the semantic similarity between two sets of instances; $\mathrm{Sim}_d(D_i, D'_j)$ represents the similarity between two sets of data values. There are several existing similarity measures for measuring the similarity between instances or data values.

3 Experimental Results and Analysis

In order to show the effectiveness and efficiency of the discovery method proposed in this paper, we compare our method with two methods for SWS discovery:

- OWLS-M0: pure logic based matchmaker in OWLS-MX[9].
- OWLS-M4: the best performing hybrid semantic matchmaker variant of OWLS-MX which computes the syntactic similarity value by the Jensen-Shannon information divergence.

OWLS-M0 can only return the services which match it logically. All the logic matching levels require that the input concepts of the request are subsumed by the input concepts of the service. OWLS-M4 uses the hybrid method to match semantic web services. Our matchmaker, denoted by RM, is a refined concept expression-based matchmaker proposed in section 2, which is implemented in JAVA using Jena. A new dataset has been constructed, which has 586 SWSs by adding 10 new SWSs to OWLS-TC v2(including 576 SWSs). These new SWSs are derived from the book price service book_price_service.owls in OWLS-TC v2, which are also considered as the queries in this evaluation. Two experts judge the relevance set of each query.

The macro averaged recall-precision curves are shown in Fig. 2. RM clearly outperforms OWLS-M4 and OWLS-M0 in terms of precision and recall except at the first recall point(where the precision of OWLS-M4 is higher than that of RM), since the OWLS-MX always ranks the equal web services(exact matching) at the beginning which are definite relevant services, while RM may put the unequal web services at the top of the ranking according to the real value of similarity.

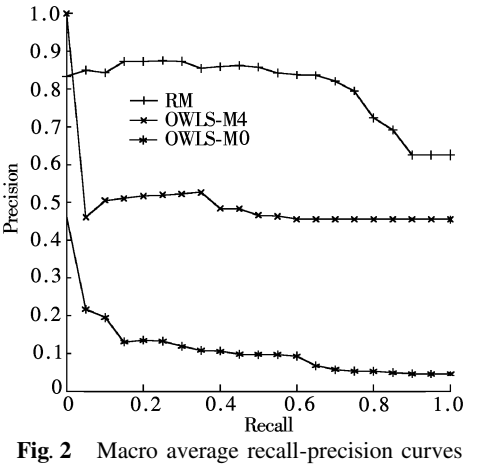

Fig. 2 Macro average recall-precision curves

The experimental evaluation for efficiency is performed on a standard PC machine with a 2 GHz Intel processor and 2 GB RAM. The queries are performed ten times for each matchmaker and the average value is used. Fig. 3 shows the execution time of these ten queries. We can observe that in most cases RM is on average faster than the OWLS-MX engine, since the comparison of refined concept expressions in RM only needs to judge the subsumption relationships between two atomic concepts or two properties, while OWLS-MX computes the semantic similarity between two concept expressions according to their subsumption relationships which causes higher computational complexity. The variant OWLS-M0 of OWLS-MX is on average faster than the OWLS-M4, since OWLS-M4 needs to unfold the concept and compute the syntactic similarity between two unfolded concept expressions. This indicates that RM has low computational complexity and can be applied in real semantic web environments with the increasing number of semantic web services annotated by concept expressions.

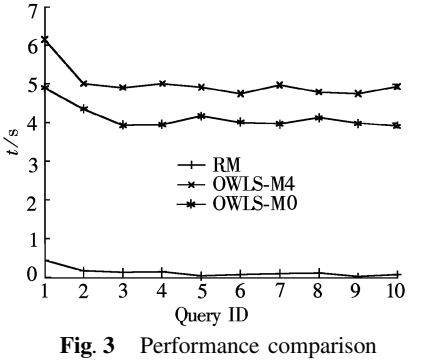

Fig. 3 Performance comparison

4 Conclusion

This paper presents a novel SWS discovery method which can distinguish the SWS through more accurate concept expressions and can dynamically reduce the semantics bias. It includes two main parts. First, we propose a method to reduce the semantic bias of annotations of the SWS by generating refined concept expressions with the help of the annotated context and the restriction template. Secondly, a novel semantic similarity measure for SWS discovery

is proposed, in which the similarity between refined concept expressions is estimated based on the instances of concept expressions. Experimental results show that concept expressions refined by the annotated context can greatly improve the discovery effectiveness, and the semantic similarity measure for concept expression can effectively rank the SWS with low computational cost.

References

[1] Klusch M. Semantic services coordination[C]//*CASCOM*: *Intelligent Service Coordination in the Semantic Web*. Berlin: Birkhäuser Basel, 2008: 59 - 104.

[2] Stollberg M, Keller U, Lausen H, et al. Two-phase web service discovery based on rich functional descriptions[C]// *Proceedings of the Fourth European Semantic Web Conference*. Innsbruck, Austria, 2007: 99 - 113.

[3] Kaufer F, Klusch M. WSMO-MX: a logic programming based hybrid service matchmaker[C]//*Proceedings of the Fourth European Conference on Web Services*. Zurich, Switzerland, 2006: 161 - 170.

[4] Paolucci M, Kawamura T, Payne T R, et al. Semantic matching of web services capabilities[C]//*Proceedings of the First International Semantic Web Conference*. Sardinia, Italy, 2002: 333 - 347.

[5] Srinivasan N, Paolucci M, Sycara K. Semantic web service discovery in the OWL-S IDE[C]//*Proceedings of the* 39*th Hawaii International Conference on System Sciences*. Kauai, HI, USA, 2005: 109.2.

[6] Wei Dengping, Wang Ting, Wang Ji, et al. Extracting semantic constraint from description text for semantic web service discovery[C]//*Proceedings of the* 7*th International Semantic Web Conference*. Karlsruhe, Germany, 2008: 146 - 161.

[7] Keßler C, Raubal M, Janowicz K. The effect of context on semantic similarity measurement[C]//*Proceedings of the International Workshop on Semantic Web and Web Semantics*(*SWWS*). Vilamoura, Portugal, 2007: 1274 - 1284.

[8] Baader F, Calvanese D, McGuinness D, et al. *The description logic handbook*: *theory*, *implementation and applications*[M]. Cambridge: Cambridge University Press, 2003: 43 - 261.

[9] Klusch M, Fries B, Sycara K. OWLS-MX: a hybrid semantic web service matchmaker in OWL-S[J]. *Journal of Web Semantics*, 2009, **7**(2): 121 - 133.

基于减少语义标注偏差的语义 Web 服务发现

魏登萍[1] 王 挺[1] 唐晋韬[1] 王 戟[2]

([1] 国防科学技术大学计算机学院, 长沙 410073)

([2] 国防科技大学并行与分布处理国家重点实验室, 长沙 410073)

摘要: 为了提高语义 Web 服务的发现性能, 从 Web 服务描述上下文中抽取语义标注的约束信息并生成新的更精确的语义标注, 从而减少语义标注与参数之间的语义偏差, 精化 Web 服务的语义描述. 首先,从本体定义中抽取概念的约束模板, 并对 Web 服务的描述文本进行句法分析; 然后,根据约束模板, 从句法分析树中抽取语义标注的约束信息, 并构造新的概念表达式作为对应参数的新的语义标注. 最后, 提出了一种新的语义相似度度量方法以度量概念表达式的相似度. 实验结果表明: 该方法能够提高语义 Web 服务发现的平均准确率, 且计算代价相对较小. 从描述文本中抽取概念的约束信息, 能够减少标注的语义偏差, 更精确地表达语义 Web 服务的语义, 提高 Web 服务的发现性能.

关键词: 语义 Web 服务发现; 语义偏差; 向下文; 约束模板; 相似度度量

中图分类号: TP311