International Journal of u-and e-Service, Science and Technology
Vol.7, No.5 (2014), pp. 293-304
http://dx.doi.org/10.14257/ijunnesst.2014.7.5.26

Services Rank by Semantic Similarity

HuiPeng

Education Technology Center, Beijing International Studies University, 100024,Beijing, China
penghui@bisu.edu.cn

Abstract

Services can be described by their semantics to improve the precision of services discovery. OWL-S describes services semantic by their inputs,outputs, precondition and effects. In the frame of OWL-S, domain ontology concepts are used to describe the semantic of inputs,outputs, precondition and effects of a service in this paper. A service rank algorithm based on the semantic similarity of ontology concepts is introduced, and a travel service discovery example shows the algorithm is effective.

Keywords: *semantic web services, services match, similarity, services rank*

1. Introduction

By combining or mashup open service on the network to build the new value-added services has become the focus of academic and industry research[13]. Many companies such as IBM, Microsoft and SAP advertised services on UDDI and provide the function of service registration and discovery. With the proliferation of web service, it is becoming increasingly difficult to find a web service that will satisfy our requirements. While UDDI[18] becomes an appealing registry for web service, its discovery mechanism which based on key word matching does not make use of semantic information of a service yields coarse results.

The semantic web initiative addresses the problem of web data lack of semantic by creating a set of XML based language and ontology. After combining semantic web with web service, researchers have built service ontology to describe capability information about a web service and make use of the description and reason function of the ontology to discovery web service automatically. Ontology Web Language for Service (OWL-S)[21,11] is a prevalent language which formalized by W3C for building service ontology.

OWL-S Profile provides methods to describe service capabilities. The function description in the OWL-S Profile describes the capabilities of the service in terms of inputs, outputs, preconditions and effects (IOPEs). An input is what is required by a service in order to produce a desired output. An output is the result which a service produces. A precondition represents conditions in the world that should be true for the successful execution of the service. And an effect of a service is its influence to its environment.

WSDL-S also provides methods to describe service inputs and outputs in its XML files[19]. It allows annotation service provider to annotation the services' IOPEs with domain ontology in its WSDL file[6].

OWL-S Profile and WSDL-S both provide upper ontology for both service advertisers and service requesters. And the semantic of services' IOPEs can be described by domain ontology concepts, ontology instances and ontology formulas. In this paper, we propose a way to match a request service and an advertised service by their similarities when their semantic is

ISSN: 2005-4246 IJUNESST

described with ontology concepts. After all similarities between advertised services and request service, the advertised services can be ranked by the similarities[2,4].

2. Semantic Similarity Calculation of Service Parameters

Because the semantic of services can be described by ontology concepts, the similarity of IOPEs of two services is concerned with the ontology concepts similarities. In this part, we define similarity of ontology concepts which concerned with service IOPEs.Subsume relation between nodes in an ontology is the most important semantic relation because it decides the classification of a domain[1,12]. So we first consider the subsume relations between nodes to discovery their semantic relations.

The match degree of two services can be judged by the subsume relations between concepts in the ontology. If the inputs of a request include all inputs of an advertised service and the outputs of an advertised service include all outputs of a request, that means the inputs and precondition of the request can satisfy the advertised service and meanwhile the outputs and results of an advertised server can satisfy the request, then we defined the request service and the advertised service are matched semantically. The semantic match of two services can be described like:

For a request service Sr={Ir,Or,Pr,Er} and an advertised service Sa={Ia,Oa,Pa,Ea}, if they satisfy the following condition meanwhile. Then Sa matches Sr semantically.

- $Ir \supseteq Ia$
- $Or \subseteq Oa$
- $Pr \supseteq Pa$
- $Er \subseteq Ea$

Because each Ir,Or,Pr,Er and Ia,Oa,Pa,Ea consists of many parameters, we should judge the match degree between every parameters to calculated the match of services.

2.1 Semantic Distance between Concepts

Inorder to calculate the semantic match degree of two parameters, we define the semantic distance and semantic similarity between two concepts in ontology like this:

If node *vi* is the same node as v_j, which marked as $v_i=v_j$then$distance(v_i,v_j)=0$

Definition 2.1: For any two nodev_i,v_j, If node $v_i$doesn't subsume node v_j, which marked as $v_j \not\subset v_i$ and node v_j doesn't subsume node, which marked as $v_i \not\subset v_j$ then $distance(v_i,v_j)=\infty$ and the $sim(v_i,v_j)=sim(v_j,v_i)$ =0.

This definition means that if there is no subsume relations between two nodes, we think there is no semantic relationship between them and 0 is the minimum value of semantic similarity of any two nodes.

For any two nodesv_i,v_j,if $i=j$, then $distance(v_i,v_j)=0$, $sim(v_i,v_j)=sim(v_j,v_i)$ =1.

That means if the two concepts in ontology are the same one, they are matched complete. 1 is the maximum value of semantic similarity of any two nodes.

Definition 2.2: If the total layer of an ontology tree is *m*, the layer where node $v_i$located is *l*, the distance between node v_i and its direct father is:

$$distance(v_i) = \frac{1}{m} + \frac{m^2-1}{m^2} * \left(\frac{m+1}{m}\right)^{-l} \quad l>=1,\ m>=2 \quad (1)$$

In formula (1), the layer of the root of the tree is 0, when the root is the only node of the tree, then l=0,m=1.From formula(1), we can concludethe distance between $v_i$and its direct father has the following feature.

1．When m is invariable, the greater the l is, the smaller the *distance*(v_l) *is.*

Proof: when $m$is invariable, *distance*(v_l)is diminishing

⇔ *distance*(v_l)> *distance*(v_{l+1}) and v_i is located in the l layer, $v_j$is located in the l+1 layer.

$$\Leftrightarrow \frac{1}{m} + \frac{m^2-1}{m^2} * \left(\frac{m+1}{m}\right)^{-l} > \frac{1}{m} + \frac{m^2-1}{m^2} * \left(\frac{m+1}{m}\right)^{-(l+1)}$$

$$\Leftrightarrow \left(\frac{m+1}{m}\right)^{-l} > \left(\frac{m+1}{m}\right)^{-(l+1)}$$

$$\Leftrightarrow \frac{\left(\frac{m+1}{m}\right)^{-l}}{\left(\frac{m+1}{m}\right)^{-(l+1)}} > 1$$

$$\Leftrightarrow \frac{m+1}{m} > 1$$

2．The value of *distance*(v_l)is between1/m *and* 1. When l=1,*distance*(v_l)=1

Proof:1)*distance*(v_l)>$\frac{1}{m}$

$$\Leftrightarrow \frac{1}{m} + \frac{m^2-1}{m^2} * \left(\frac{m+1}{m}\right)^{-l} > \frac{1}{m}$$

$$\Leftrightarrow \frac{m^2-1}{m^2} * \left(\frac{m+1}{m}\right)^{-l} > 0$$

$$\Leftrightarrow \frac{m^2-1}{m^2} > 0 \text{ and } \left(\frac{m+1}{m}\right)^{-l} > 0 \quad (2)$$

Because $m>=2$,equation (2)is true apparently.

2) *distance*(v_l) ≤1

⇔when l=1,*distance*(v_l)=1 and *distance*(v_l)is diminishing(proved in 1.)

3．The semantic distance between two nodes v_i,v_j, which $v_j \subset v_i$ can be:

distance(v_i,v_j)=*distance*(v_{l+1})+ *distance*(v_{l+2})+…+ *distance*(v_k)

$$=\frac{k-l}{m} + (m+1)^{1-k} * (m-1) * (m)^{l-1} * ((m+1)^{k-l} - m^{k-l}))$$

$<m$-1

(l, k *each is the layer which node* v_i,v_j*located,* and k>l>=1,m>=2) (3)

Proof: *distance*(v_i,v_j)=*distance*(v_{l+1})+ *distance*(v_{l+2})+…+ *distance*(v_k)

$$distance(v_i, v_j) = distance(v_{l+1}) + distance(v_{l+2}) + \ldots + distance(v_k)$$

=〉 $(k-l)/m < distance(v_i,v_j) \leq (k-l)$

=〉 $(k-l)/m < distance(v_i,v_j) < m-1$

From formula(3),we can calculate the distance between nodes in Figure1, the results is shown in Table1

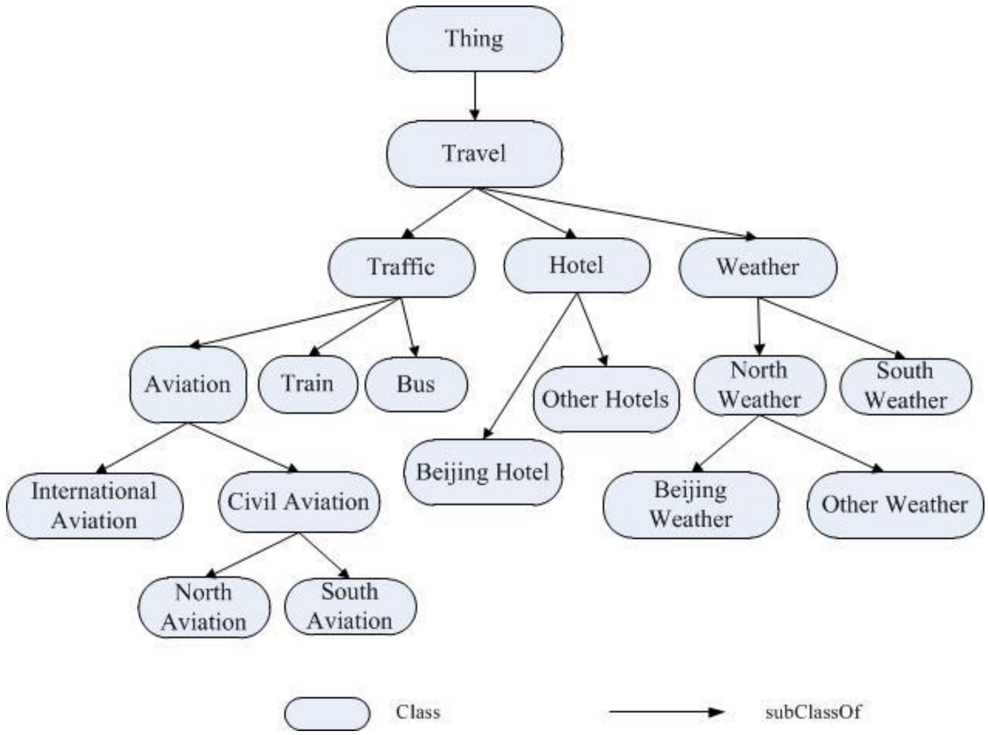

Figure 1. Part of Travel Ontology

Table 1. Some of Distance between Concepts in an Ontology

	(Thing, Travel)	(Travel, Traffic)	(Traffic, Aviation)	(Aviation, Civil Aviation)	(Civil Aviation, South Aviation)
distance	1	0.88	0.78	0.69	0.62

From Table 1, we can conclude that the distance of two nodes which have consumed relationship is decreasing with the increase of depth of the nodes related. That means the semantic difference of the two nodes is decreasing. For example, the similarity of "Thing" and "Traffic" is larger than the similarity of "Civil Aviation" and "South Aviation" apparently.

2.2 Semantic Similarity between Concepts

Definition 2.3:

There are two nodes v_i,$v_j$in an ontology tree,if $v_j \subset v_i$ The similarity between them is defined like the following:

$$sim(v_j,v_i)=\frac{1}{2}+\frac{1}{2}*(1-distance(v_i,v_j)/(m-1)) \text{ (j>i>=1,m>=2)} \quad (4)$$

$$sim(v_i,v_j)=\frac{1}{2}*(1-distance(v_i,v_j)/(m-1)) \text{ (j>i>=1,m>=2)} \quad (5)$$

Because $sim(v_j,v_i)\neq sim(v_i,v_j)$,we call the similarity between nodes is v_i,$v_j$unsymmetrical and marked as$sim<v_j,v_i>$ *and* $sim<v_i,v_j>$

From definition 2.3 we can conclude that:

1． if $v_i \subset v_j$,then $sim<v_j,v_i> \in (1/2,1\ )$ and $sim<v_i,v_j> \in (0,1/2)$,

2． if $v_j \subset v_i$, $sim<v_j,v_i>$ *and* $sim<v_i,v_j>$ is inversely proportional to $distance(v_i, v_j)$.

So for any two nodes v_i,v_j which exist consume relation, the similarity of node v_i to node v_j is not equal to the similarity of node v_j to node v_i. if v_i subsumes v_j, then the range of $sim<v_i,v_j>$ is from 0 to 1/2 and the larger the distance the smaller the similarity.The $sim<v_j,v_i> =$ $sim<v_i,v_j> +1/2$, the range of $sim<v_j,v_i>$ is from 1/2 to 1 and also the larger the distance is the smaller the similarity is.

From definition 2.1,2.2 and 2.3,the similarity of any two nodes in ontology tree is:

$$sim \prec v_j, v_i \succ = \begin{cases} 1 & if\ \ v_i = v_j \\ \frac{1}{2} + \frac{1}{2} * (1 - distance(v_i, v_j) / (m-1)) & if\ \ v_j \subset v_i \text{且} v_i \neq v_j \\ \frac{1}{2} * (1 - distance(v_i, v_j) / (m-1)) & if\ \ v_i \subset v_j \text{且} v_i \neq v_j \\ 0 & if\ \ v_i \not\subset v_j \text{且} v_j \not\subset v_i \end{cases} \quad (6)$$

Table 2 shows some similarities between nodes in Figure1 according formula(6)

Table 2. Examples of Similarity Calculation

	<Civil Aviation, Civil Aviation>	<Civil Aviation, Train>	<Civil Aviation, Aviation>	< Aviation, Civil Aviation>
sim	1	0	0.93	0.43

Table2 shows that $sim<v_j, v_i> \neq sim< v_i, v_j>$ when there is subsume relation between node v_i and v_j.For example,*sim*< Civil Aviation, Aviation＞≠*sim*< Aviation,Civil Aviation＞

We define the similarity of service parameters in part 2.3 after defining the concepts similarity.

2.3 The Similarity of Service Parameters

One of inputs, one of outputs ,one of preconditions and one of results of a request service each is expressed as inR,outR,preR,reR. One of inputs, one of outputs,one of preconditions and one of results of an advertised service each is expressed as inA、outA,preA,reA. The similarity of service parameters is defined as:

degreeOfTwoOutputMatch(outR,outA)=sim<outR,outA>
degreeOfTwoInputMatch(inR,inA)=sim<inA,inR>
degreeOfTwoResultMatch(reR,reA)=sim<reR,reA>
degreeOfTwoPreconditionMatch(preR,preA)=sim<preA,preR>

The similarity of service parameters can be explained as the following:

Take the output parameter as anexample, there are four situation of the similarity of two outputs.

1. If there is no subsume relation between two output concepts, the semantic distance of two outputs is ∞, the similarity is 0. Take Figure1 as an example, if one request output of a request service is “Civil Aviation” and one output of an adverdised service is “Trian”.authough both “Civil Aviation” and “Trian” are children of “Traffic”,but the advertised service cannot satisfy the request apparently. So the two outputs is not match. The similarity of them is 0.

2. When the two output express by the same concept, the distance of them is 0 and the similarity of them is 1. They are matched completely.

3. If the output of an advertised service subsumes the output of a request service, they are “plugIn” match. The range of the similarity of two outputs is (1/2,1), and the nearer the distance the more similar they are. Take request output “Civil Aviation” as an example, the service which a output is “Traffic” can usually satisfy the request output, but the service which a output is “Aviation” can satisfy the request output better than “Traffic”. Both similarities of *sim*<Civil Aviation,Traffic> and *sim*<Civil Aviation, Aviation> are more than 1/2,but the later one is nearer to 1.

4. If the output of a request service subsumes the output of an advertised service, they are “subsumes” match. The range of the similarity of two outputs is (0,1/2), and the nearer the distance the more similar they are. Take request output “Traffic” as an example, the service which a output is “Aviation” can partly satisfy the request output, and the service which a output is “North Aviation” will partly satisfy the request output worse than “Aviation”. Both similarities of *sim*<Traffic, Aviation> and *sim*<Traffic, North Aviation> are less than 1/2,but the frontier one is nearer to 1/2.

From the above analysis, the similarity of service outputs decides the match degree of the two outputs. The larger the similarity is the higher degree they match. Table 3 shows the conclusion.

Table 3. The Relationship between Match Degree and the Value of Similarity

Match degree	Value of similarity
exact	1
plugIn	(1/2,1)
subsumes	(0,1/2)
fail	0

For two inputs of services, the condition is conversed with the outputs of services. When the request input consumes the advertised input, that means the request input will satisfy the advertised input well, the range of the similarity value is (1/2,1). When the advertised input consumes the request input, that means the request input will not satisfy the advertised input well, the range of the similarity value is (0,1/2).

We conclude the match condition for a pair of input and a pair of output. We can conclude the service match degree from the value of all pairs of input and all pair outputs.

3. Service Match based on the Similarity of Service Parameters

Every request service and advertised service includes some inputs and outputs. If the request inputs expressed as *Request(ri1,ri2,…rin)*, and the outputs expressed as

Request(ro1,ro2,…rom). The advertised inputs expressed as *Advertise(ai1,ai2,…aiu)*,and the inputs expressed as *Advertise(ao1,ao2,…aov)*. If *ri1,ri2,…rin* and *ro1,ro2,…rom* corresponding to nodes *vri1,vri2,…vrin* and *vro1,vro2,…vrom* in ontology tree. We can calculate the value of every pair of output or input from matrix A or B.

$$A = \begin{array}{c|cccc|} & vao_1 & vao_2 & \dots & vao_v \\ vro_1 & sim(ro_1,ao_1) & sim(ro_1,ao_2) & \dots & sim(ro_1, ao_v) \\ vro_2 & sim(ro_2,ao_1) & sim(ro_2,ao_2) & \dots & sim(ro_2, ao_v) \\ & \dots & \dots & \dots & \\ vro_m & sim(ro_m,ao_1) & sim(ro_m,ao_2) & \dots & sim(ro_m, ao_v) \end{array}$$

Let $sim(o_1)=max(sim(ro_1,ao_j)),\ j\in[1, v], j\in N;$

$sim(o_2)=max(sim(ro_2,ao_j)),\ j\in[1, v], j\in N;$

.. .. ..

$sim(o_m)=max(sim(ro_m,ao_j)),\ j\in[1, v], j\in N$

$$sim(Output_{Request}, Output_{Advertise})=\sum_{i=1}^{m} w_i * sim(O_i),\ w_i\in[0,1],$$

w_i is the weight of every request output,itreflects the importance of every request output.

$$B = \begin{array}{c|cccc|} & vri_1 & vri_2 & \dots & vri_n \\ vai_1 & sim(ai_1, ri_1) & sim(ai_1, ri_2) & \dots & sim(ai_1, ri_n) \\ vai_2 & sim(ai_2, ri_1) & sim(ai_2, ri_2) & \dots & sim(ai_2, ri_n) \\ \dots & \dots & \dots & & \\ vai_u & sim(ai_u, ri_1) & sim(ai_u, ri_2) & \dots & sim(ai_u, ri_n) \end{array}$$

Note that the sequence of parameters in matrix A is conversed with the sequence of parameters in matrix B.

Let $sim(i_1)=max(sim(ai_j,ri_1)),\ j\in[1, u], j\in N;$

$sim(i_2)=max(sim(ai_j,ri_2)),\ j\in[1, u], j\in N;$

.. .. ..

$sim(i_n)=max(sim(ai_j,ri_n)),\ j\in[1, u], j\in N$

$$sim(Input_{Request}, Input_{Advertise})=\sum_{i=1}^{n} h_i * sim(I_i),\ h_i\in[0,1],$$

h_i is the weight of every request input,it reflects the importance of every request input.

With the same logic of service inputs , we can calculate the similarity of precondition of services $sim(Pre_{Request}, Pre_{Advertise})$.

The match value of services is:

$$sim(Request, Advertise)=k_1 sim(Input_{Request}, Input_{Advertise})+ k_2 sim(Output_{Request}, Output_{Advertise}) + k_3 sim(Pre_{Request}, Pre_{Advertise})$$

k_1,k_2,k_3 is the weights

Algorithm1 describes the procedure of calculating all distance between any two nodes in the tree, and the tree is directed and is scanned once. This algorithm is the most difficult and important in the calculation of service match.

Algorithm 1.The distance of two nodes in a tree

Inputs:
Tree *T //direct tree expressed by adjacency list
int root // root of the tree
int K; //total number of nodes in the tree

```
int m;   /total layer of the tree
outputs:
distance[1..K,1..K] // distance of nodes in  tree
int[][] computeDist(Tree * T, int root, int K){
        for (int i=1;i<=K;i++)
              for (int j=1;j<=K;j++)
              if (i==j)
distance[i][j]=0;  //  the initial value of the distance of a node to  itself is 0.
              else
distance[i][j]=infinite;  //   the initial value of the distance of a node to  another node is
                                                    infinite./
int Pre[] = new int[logK];  //Pre the node sequence from  root to the precursor of node vi/
DFS(Tree *T, int root, Pre[], 0); // scan tree T by the sequence of root, left child and right
                                                              child.
        return distance;
    }

/*the following the algorithm of scanning the tree by root first. v is the node being visited.
Pre is the node sequence from root to v(not including v).Prelength is the number of nodes
from root to v(not including v)*/
void DFS(Tree *T, int v, int Pre[], intPreLength){
        EdgeNode *p;
        for (int i=0;i<PreLength;i++) {
distance[Pre[i]][v]= distance[Pre[i-1]][v]+ 1/m+(m^2-1)/m^2+((m+1)/m)-^i //calculate the distance
                                            from the precursors to v
              }
Pre[PreLength]=v  //  before visiting v, put v into the precursors path array Pre.
    // visit the children of v
    p=T->adjlist[v].firstedge;
while(p)
if (!visited[p->adjvex]) {
DFS(T,p->adjvex, Pre[], PreLength++);
              p=p->next;
          }
  }
```

4. Example of Service Match

The following is the example of services match. Figure 2 is part of the OWL-S file of the advertised service. It describes the semantic of inputs and outputs on traffic and weather query.The service gets “date” from input and outputs the “traffic”and“weather” condition of the “date”. Figure 3 is part of OWL-S file of the request service. It describes the semantic of inputs and outputs on demand. The request service inputs “date” and want to gain the information of “civil aviation”,“hotel” and “Beijing weather”. The ontology is show in Figure1. According to the above algorithm the similarity of the two services is 0.8. The procedure of calculation is:


```
<profile:Profilerdf:ID="Traffic&WeatherService">
<profile:serviceName>Traffic&WeatherService</profile:serviceName>
<profile:providedBy> ... </profile:providedBy>
<input>
<profile:ParameterDescriptionrdf:ID="   Time_Input">
<profile:parameterName>Date</profile:parameterName>
<profile:restrictedTordf:resource="Time.owl#Date"\>
</profile:ParameterDescription>
</input><output>
<profile:ParameterDescriptionrdf:ID="Trffic_Output">
<profile:parameterName>Traffic</profile:parameterName>
<profile:restrictedTordf:resource="Traffic.owl#Traffic"\>
</profile:ParameterDescription>
<profile:ParameterDescriptionrdf:ID="WeathReport_Output">
<profile:parameterName>Weather</profile:parameterName>
<profile:restrictedTordf:resource="Travel.owl#Weather"\>
</profile:ParameterDescription>
</output></profile:Profile>
```

Figure 2. The Aviation and Weather Description of an Advertised Service

```
<profile:Profilerdf:ID="TravelService">
<input>
<profile:ParameterDescriptionrdf:ID="   Time_Input">
<profile:parameterName>Date</profile:parameterName>
<profile:restrictedTordf:resource="Time.owl#Date"\>
</profile:ParameterDescription>
</input><output>
<profile:ParameterDescriptionrdf:ID="Traffic_Output">
<profile:parameterName>Civil Aviation</profile:parameterName>
<profile:restrictedTordf:resource="Traffic.owl# Civil Aviation "\>
</profile:ParameterDescription>
<profile:ParameterDescriptionrdf:ID="Hotel_Output">
<profile:parameterName>Hotel</profile:parameterName>
<profile:restrictedTordf:resource="Travel.owl#Hotel"\>
</profile:ParameterDescription>
<profile:ParameterDescriptionrdf:ID="WeatherReport_Output">
<profile:parameterName>Beijing Weather</profile:parameterName>
<profile:restrictedTordf:resource="Travel.owl#Beijing Weather "\>
</profile:ParameterDescription>
</output></profile:Profile>
```

Figure 3. The Aviation, Hotel and Weather Description of a Request Service

The similarity of inputs of two services is:

sim(Date,Date)=1,

Because there is only one input for request and advertised service, the match degree of inputs is:

degreeOfInputMatch=1.

The similarities of outputs of two services are:

sim(Civil Aviation, Aviatin):

M=6,k=5,l=4,distance(Civil Aviation, Aviatin) =0.69,sim (Civil Aviation, Aviatin)=(1/2)+(1/2)(1-0.69/5))=0.93*

sim(Civil Aviation, Weather)=0,

The maximum value of the output similarities is the match output for request output "*Civil Aviation"*. That is:

sim(Civil Aviation)=max(0.93,0)=0.93

With the same reason,we can get:

sim(Hotel, Aviatin)=0,

sim(Hotel,Weather)=0

sim(Hotel)=max(0,0)=0

sim(Beijign Weather, Aviatin)=0,

distance(Beijign Weather, Weather)=0.69+0.78=1.47,sim(Beijign Weather, Weather)=0.88,

sim(Beijign Weather)=max(0,0.88)=0.88

Let the outputs degree of two services is the weight average of all output parameters, that is:

degreeOfOutputMatch=w1 sim(Civil Aviation)+w2* sim(Hotel) +w3* sim(Beijign Weather)*

And*w1,w2,w3 are weight, letw1=w2=w3=1/3,*then

*degreeOfOutputMatch =1/3*0.93+1/3*0.88=0.6*

Let the services match degree is the weight average of inputs and outputs,then:

degreeOfServiceMatch=u1 degreeOfInputMatch+u2* degreeOfOutputMatch*

Let*u1=u2=0.5,*then

*degreeOfServiceMatch =0.5*1+0.5*0.6=0.8*

The example shows that the two services match well while their service names "TravelService" and "Traffic&WeatherService" are quite different. After we calculate all the similarities between request service and advertised services we can rank the advertised services by similarities.

5. Related Work

In order to improve web services discovery process and results, a lot of research work. have been done. semantic discovery methods for webservices have proposed in work [3,5,14,20, 19]. In work[3], services is matched in four grades:exact, plugIn,subsume and fail. Work [3] also describes the procedure of service discovery in OWL-S. work[14, 20] introduces the wsmo model for semantic web services. Work[19] introduces the wsdl-s model for semantic web services. The advantage of semantic services discovery is its high precision, but the shortcoming of semantic services discovery is its low efficiency. To improve the efficiency of semantic services discovery, we pre-process all nodes in ontology tree and save the similarities of any two nodes. When there is a request, only inputs and outputs parameters matrix are calculated. Another feature of our algorithm is the Arithmetic value of similarity is convenient for rank. Work[15, 16] proposed the distributed service discovery to improve the scalability of services discovery. They are based p2p structure. As more and more services are deployed in a variety of websites directly, work[7, 8, 9, 17, 10] proposed services discovery based on search engine and use search engine to search services on the Web.

Acknowledgements

This work is supported by the scientific plan project of Beijing municipal commission of education(No.SQKM201410031001)and the National Science Foundation of China (No. 61103161).

Reference

[1] P.Ganesan, H.G. Molina and J.Widom,"Exploiting hierarchical domain structure to compute similarity", ACM Transactions on Information Systems, vol. 21, no. 1, **(2003)**, pp. 64-93.
[2] J.M. García, D. Ruiz and A. Ruiz-Cortés, "A Model of User Preferences for Semantic ServicesDiscovery and Ranking", The Semantic Web: Research and Applications, Lecture Notes in ComputerScience, **(2010)**,
[3] N.Srinivasan, M.Paolucci and K.Sycara, "Adding OWL-S to UDDI, implementation and throughput", The First International Workshop on Semantic Web Service, **(2004)**.
[4] A.Gunay and P.Yolum, "Structure and Semantic Similarity Metrics for Web Service Matchmaking", EC-Web,**(2007)**.
[5] R. Aydogan and H.Zirtiloglu, "A Graph-Based Web Service Composition Technique Using OntologicalInformation", Proc. of ICWS,**(2007)**, pp. 1154-1155.
[6] K.Elgazzar, A.E. Hassan and P. Martin, "Clustering WSDL Documents to Bootstrap the Discovery ofWeb Services", In Proc. of ICWS, **(2010)**, pp. 147-154.
[7] H. Song, D. Cheng, A. Messer and S. Kalasapur, "Web Service Discovery Using General-Purpose SearchEngines", In Proc. of ICWS,**(2007)**, pp. 265 – 271.
[8] E. A. Masri and Q. H. Mahmoud, "Investigating Web Services on the World Wide Web", Proc.Of WWW, **(2008)**, pp. 795-804.
[9] N. Steinmetz, H. Lausen and M. Brunner,"Web Service Search on Large Scale", In Proc. of InternationalConference on Service Oriented Computing (ICSOC), **(2009)**.
[10] Y. Li, Y. Liu, L. Zhang, G. Li, B. Xie and J. Sun, "An Exploratory Study of Web Services on the Internet", In Proc. of ICWS, **(2007)**, pp. 380-387.
[11] M. Klusch, B. Fries and K. Sycara, "Automated Semantic Web Service Discovery with OWLS-MX", Proc.of AAMAS, May, **(2006)**; Hakodate, Japan.
[12] Z. Lin, M.R. Lyuand I. King, "MatchSim: A Novel Neighbor-Based Similarity Measure With MaximumNeighborhood Matching", Proceeding of the 18th ACM conference on Information and knowledgemanagement(CIKM'09), ACM Press,**(2009)**.
[13] M. Rambold, H. Kasinger, F. Lautenbacher and B. Bauer, "Towards Autonomic Service Discovery – A Survey and Comparison", Proc. of SCC,**(2009)**, pp. 192-201.
[14] C.Feier, D.Roman, A.Polleres,*et al.*, "Towards Intelligent Web Services: The Web Service Modeling Ontology(WSMO)", In the Proceedings of 2005 International Conference on Intelligent Computing, **(2005)**; Hefei,China.
[15] Q. He, J. Yan, Y. Yang, R. Kowalczyk and H. Jin, "Chord4S: A P2P-based Decentralised Service DiscoveryApproach", Proc. of SCC, **(2008)**, pp. 221-228.
[16] Y. Zhang, L. Liu, D. Li, F. Liu and X. Lu, "DHT-Based Range Query Processing for Web Service Discovery", In Proc. of ICWS, **(2009)**, pp. 477-484.
[17] Y. Zhang, Z. Zheng and M.R.Lyu, "WSExpress: A QoS-aware Search Engine for Web Services, In Proc.of ICWS,**(2010)**, pp. 91-98.
[18] Universal Description, Discovery and Integration(UDDI),http://www.uddi.org/specification.html
[19] R. Akkiraju, J. Farrell, J. Miller. Web Service Semantics-WSDL-S.http://www.w3.org/Submission/WSDL-S/,**(2010)**.
[20] J.D. Bruijn, C. Bussler, J. Domingue, D. Fensel. Web Service Modeling Ontology(WSMO). Available from: http://www.w3.org/Submission/WSMO/, **(2010)**.
[21] The OWL Services Coalition. OWL-S: Semantic Markup for Web Services,http://www.w3.org/Submission/OWL-S/, **(2004)**.

Author

HuiPeng, she received her PhD in Computer Sciences (2009) from Institute of Computer Technology Chinese Academy of Science. Now she is full professor of education technology center, Beijing International Studies University. Since 2012 he is rector of University. Her current research interests include different aspects of Artificial Intelligence and Distributed Systems.