An Edge-Based Approach for Semantic Matchmaking of Service Capabilities

Adala Asma, Tabbane Nabil and Tabbane Sami
Multimedia Mobile Radio Networks Research Unit (MEDIATRON), Higher School of Communication of Tunis (Sup’Com), University of Carthage, Ariana, Tunisia

Received: May 09, 2011 / Accepted: June 02, 2011 / Published: August 25, 2011.

Abstract: Semantic Web Services is an emerging technology that promises to enable dynamic, execution-time discovery, composition, and invocation of Web Services. Semantic matchmaking plays a vital role in the automated and dynamic discovery process of Semantic Web Services and consists in measuring the semantic distance between a requested service and an advertised one. In this paper, an innovative approach to effectively compute the semantic distance between Ontology Web Language for Services (OWL-S) annotated services is proposed. First, an edge-based method for measuring the semantic distance between Web Ontology Language (OWL) concepts is presented. Then, a comparison of the proposed measure and the one presented in a recent related work is made in order to show that our method is more efficient and fine-grained. Finally, some equations to compute semantic matchmaking of service capabilities, which are expressed in terms of inputs and outputs, are presented.

Key words: Semantic web, semantic web service, semantic distance, matchmaking, service discovery, OWL-S (ontology web language for services), ontology.

1. Introduction

Web Services represent the cornerstone of service-oriented architectures today. They are loosely coupled software components that are published, invoked and executed across the Web over standard Internet protocols, which facilitates business applications interaction and integration. Therefore, Web Services are recognized and widely accepted from experts on industry and academia.

As the number of available Web Services on the Web is increasing, support for service discovery mechanisms becomes essential. A discovery process aims to find the appropriate Web Services, among those previously published, according to user’s requirements or goals. This process is still a challenging task, essentially because existing technologies which currently allow Web Service discovery like Universal Description Discovery and Integration (UDDI) [1] are based on keywords matching with service description attributes like service or provider name. In fact, syntactic matching does not allow retrieving services with similar functionalities. For example, two Web Service Description Language (WSDL) [2] descriptions can be used to describe the same service but with different words such as “vehicle” and “car”. With a syntactic search, both of these Web services will not be returned when only one of the terms is used as keyword in Web service discovery. Moreover, syntactic matching is not suited for automatic processing so it still requires human interaction. Thus, a more reliable and effective Web Service discovery approaches is needed.

With the emergence of Semantic Web [3] concept, researchers have focused on enriching service descriptions with machine-readable semantics to obtain Semantic Web Services (SWSs). SWSs pledge
the automation of core tasks related to Web Services such as discovery.

The vision of Semantic Web is to minimize the manual discovery and usage of Web resources, such as Web Services, by allowing software agents to automatically identify, integrate and execute these Web resources to achieve the user objectives [4]. Semantic matching plays a vital role in discovery process. It can be performed by exploiting the semantic representation of concepts and their relations.

The first step towards SWS is to define an ontology description language for Web Services. An ontology description language is designed for use by applications that need to process the content of information instead of just presenting information to humans. An ontology may include descriptions of classes (or concepts), along with their related properties and instances in a specific domain. Currently, there is no standard language for SWS description. But, several standards have been proposed: WSDL-S [5] is a lightweight approach for adding semantics to WSDL Web service description. In WSDL-S, the semantic models are maintained outside of WSDL documents and are referenced from the WSDL document via WSDL extensibility elements. Web Service Modeling Ontology (WSMO) [6] provides a conceptual framework and a formal language to describe all relevant aspects of Web services to facilitate the automation of service discovery using semantics. Ontology Web Language for Services (OWL-S) [7] is a Web Ontology Language (OWL)-based [8] Web service ontology, which supplies Web service providers with a core set of markup language constructs for describing the properties and capabilities of their Web services in unambiguous and computer-interpretable form. An OWL-S description is composed of three parts which are Service Profile, Service Model and Service Grounding. The Service profile describes service capabilities and it is the part used in the discovery process. The Service Model describes how the service works (internal processes), and the Service Grounding specifies the details of how the service can be accessed. Semantic descriptions are added to functional parameters like inputs and outputs in Service Profile by mapping each of them to a concept which is defined in an OWL ontology. Therefore, to accomplish semantic matching of Web Service capabilities, it is necessary to calculate the semantic distance between the input/output concepts of queried service and the input/output concepts of advertised service.

The outline of this paper is as follows: In section 2, we discuss some related work in the area of semantic matching for Web Service discovery. In section 3, we present our approach to calculate semantic distance between two concepts defined in an ontology. We also present some functions for semantic matching of service capabilities in section 4. The last section concludes the paper and gives an overview of our future work.

2. Related Work and Research

Many approaches have been proposed to determine the degree of match or the semantic similarity between Web Services in order to allow automatic and dynamic discovery of services among the Web.

One approach is based on the subsumption relations between concepts in the taxonomy tree deduced from the domain ontology description. This approach was presented in [9-13]. In Ref. [9], M. Paolucci, et al., propose a matchmaking algorithm having two input/output concepts as inputs and one of four degrees of match as output, that are Exact, Plug-in, Subsumes and Fail. They propose also a ranking algorithm to sort matchmaking results. An extension to this algorithm is presented in Ref. [10] by adding the intersection relation between concepts. It presents also a prototype implementation for a matchmaking agent based on Racer [14] reasoner and Jena API [15]. In Ref. [11], a two step matchmaking
algorithm is presented. In the first step, the degree of match between concepts is calculated based on the algorithm proposed by Paoulucci. Then, in the second step, a usability function which expresses relation between the number of required inputs/outputs and the number of advertised inputs/outputs is calculated. Usability consideration aims to define the client convenience for executing the services. Some limitations in Paoulucci algorithm are identified in Ref. [12] and a more exhaustive matching algorithm is proposed. It models semantic matchmaking as a bipartite graph and then it computes the optimal matching using the Hungarian algorithm. The approach presented in Ref. [13] proposes to filter from all the advertised services those that belong to the same category as those requested by the client, before applying matching algorithm and shows that it enhances response time. The matching algorithm detailed in Ref. [13] extends the one proposed in Ref. [9] by considering Sibling relation between concepts.

In Ref. [16], another approach which extends simple subsumption matching to semantic distance calculation is presented. The proposed semantic distance function is the sum of the distance between two concepts that are defined in an ontology description language. As presented in previous section, the majority of related works focus on the subsumption relations between concepts in the taxonomy tree (ontology). The most important disadvantage of this approach is that it leads to coarse-grained results limited to few degrees of match. Giving thousands of advertised Web Services in a registry, the matching algorithm will return a great number of candidate services with the same degree of match with the queried one. This will affect discovery efficiency and complicate the selection process for client application which is in contrast with the SWS vision.

In this section, a novel edge-based approach to measure the semantic distance between two concepts expressed in OWL language is detailed. We take benefit from researches [19-22] done in Natural Language Processing (NLP) which is a field of computer science and linguistics concerned with the interactions between computers and human languages. It studies the semantic characteristics and relations (e.g. polysemy, synonymy) that exist between words of natural human language and has many applications such as machine translation, multilingual or cross-language information retrieval and speech recognition.

In this work, the two concepts that will be matched together are supposed to be defined in the same ontology. According to the ontology description, a
taxonomy tree is constructed, similar to Fig. 1, representing the subsumption relations (IS-A relations) between concepts in a specific domain. The concept semantic distance function is defined as follows:

Definition 1 (Concept Semantic Distance):
A concept semantic distance function $\sigma$ is a function from a pair of concepts to a real positive number,

$$ \sigma : C \times C \rightarrow [0, \infty) $$

such that:

1. $\forall c_0 \in C, \sigma(c_0, c_0) = 0$
2. $\forall c_1, c_2 \in C, \sigma(c_1, c_2) = \sigma(c_2, c_1)$

To measure the semantic distance between concepts, a weight value $wt$ is assigned for each edge in the taxonomy. $p(c)$ is the parent concept (super concept) of $c$ in the taxonomy. We define the semantic distance function as the sum of all edges weights along the shortest path $\text{path}(c_1,c_2)$ between the two nodes (concepts) $c_1$ and $c_2$ in the hierarchy (taxonomy):

$$ \sigma(c_1, c_2) = \sum_{c \in \text{path}(c_1, c_2)} wt(c, p(c)) \quad (1) $$

To consider all the edges between the two concepts, it is important to determine the lowest super concept $LS(c_1, c_2)$ in common between $c_1$ and $c_2$.

3.1 Lowest Super Concept Definition

Let consider $T$ as the taxonomy in which all concepts are represented, the relation $c \ll c'$ means that $c$ is subsumed by $c'$ (IS-A $c'$) or $c$ is the same as $c'$. Thus, we first define the set of super concepts of a concept $c$ in a taxonomy $SC(c, T)$ as follows:

$$ SC(c, T) = \{ c' \in T; c \ll c' \} $$

Thus, the set of common super concepts between $c_1$ and $c_2$ is:

$$ CSC(c_1, c_2) = SC(c_1, T) \cap SC(c_2, T) $$

Finally, the lowest super concept of $c_1$ and $c_2$ $LS(c_1, c_2)$ is the common super concept between $c_1$ and $c_2$ which is subsumed by all other common super concepts:

$$ c_0 = LS(c_1, c_2) \Leftrightarrow c_0 \in CSC \land \forall c' \in CSC; c_0 \ll c' $$

3.2 Edge Weight Calculation

An edge in the taxonomy corresponds to an IS-A relation between a concept $c$ and its parent $p(c)$. Our edge-based approach estimates the distance (edge length) between the two concepts. The shorter the path from one node (concept) to another, the more similar they are.

Fig. 1  Example concept taxonomy (fragment from OWL travel ontology).
For an IS-A semantic hierarchy (taxonomy), the simplest way to calculate the distance between concepts is the shortest path that links the two concepts. In reality, distances between any two adjacent nodes are not necessarily equal. For example in Fig. 1, we can easily realize as humans that “Town” and “City” are more similar than “Urban Area” and “Rural Area” although in both case, they are sibling concepts. Therefore, edges should be weighted.

Many aspects have been considered to calculate the weights of edges in NLP researches. We use a slightly modified and a more adapted version of the equation presented in Ref. [20]. The weight value of each edge in the taxonomy is calculated with the following formula:

\[
\text{wt}(c,p) = \left( \beta + (1 - \beta) \frac{E(p)}{d(p)} \right)^{\alpha} \left( \frac{d(p) + 1}{d(p)} \right)
\]

Our proposed edge weight value depends on two parameters that are the depth of the parent node (super concept) in the hierarchy \(d(p)\) and the local density of the parent node \(E(p)\). Since the edge weight formula depends only on the characteristics of the parent node, the weights of all edges between sibling concepts and their super concept are equals. Consequently, semantic distance between a super concept and each of its child concepts is the same. Moreover, semantic distance between a concept and each of its sibling concepts is also the same.

3.2.1 Depth of Node

The depth of a node \(d(p)\) is the distance between the concept \(p\) and the root node in the taxonomy. \(d(p)\) is always superior or equal to 1 \((d(p) \geq 1)\). It is equal to 1 if the node \(p\) is the root node in the taxonomy \(T\). As one descends the hierarchy, the distance between concepts decreases because differentiation is based on finer and finer details in the OWL description. Thus, in the formula when \(d(p)\) increases, edge weight \(wt\) decreases. In the Eq. (2), \(\alpha\) is a positive or nulle parameter \((\alpha \geq 0)\) that controls the degree of how much the node depth contributes to the edge weighting computation.

3.2.2 Local Density of Node

If we observe the ontology density, we can notice that the densities in different parts of the hierarchy are higher than others. The local density effect \([21]\) would suggest that the greater the density, the closer the distance between the nodes (i.e., parent/child nodes or sibling nodes). The local density \(E(p)\) is defined as the number of edges in the child links of the node \(p\) in the taxonomy. In order to evaluate the density of a node, we need to compare it with the average density \(E\) on the whole ontology which is defined as follows:

\[
E = \frac{\sum_{c \in T} E(c)}{\text{number of concepts}}
\]

The semantic distance between the two concepts \(City\) and \(Rural Area\), represented in Fig. 1, is calculated as follows (currently we set \(\alpha = 1\) and \(\beta = 0\)):

- \(LS(City, Rural Area) = Destination\)
- \(E = 0.928\)
- \(wt(Rural Area, Destination) = 0.232\)
- \(wt(Urban Area, Destination) = 0.232\)
- \(wt(City, Urban Area) = 0.174\)
- \(\sigma(City, Rural Area) = 0.812\)

In the same way, the semantic distances between some pairs of concepts represented in Fig. 1 are calculated. A comparison of the matching results of our approach and the one represented in Ref. [13] is shown in Table 1.

From Table 1, we can notice that our approach gives more precise matching results since in our approach the matching degree (semantic distance) could be any positive real number in \([0, \infty]\) although in Ref. [13] it is limited to seven values. Our proposed concept semantic distance equation offers a fine-grained method to compare similarity between the requested concept and the advertised one. For example, in Ref. [13] every matching between any child/parent concepts will return \(CSubclassP\) and also every matching between any sibling concepts will return \(CSiblingP\). However, in our approach we obtain different results each time according to the specific child/parent or sibling concepts properties and also according to the related ontology which will necessarily enhance the efficiency and accuracy of the discovery process.
Table 1 Comparison of our approach and the one in Ref. [13].

<table>
<thead>
<tr>
<th>Approach</th>
<th>Concepts</th>
<th>City, RuralArea</th>
<th>UrbanArea, Destination</th>
<th>City, UrbanArea</th>
<th>Town, Capital, Destination</th>
<th>Town, City, FarmLand, NationalPark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our approach</td>
<td></td>
<td>0.812</td>
<td>0.232</td>
<td>0.174</td>
<td>0.406</td>
<td>2.049</td>
</tr>
<tr>
<td>Approach in Ref. [13]</td>
<td></td>
<td>Fail</td>
<td>CSUBCLASSP</td>
<td>PSUBSUMECSUBCLASSP</td>
<td>PSUBSUMESCBSIBLINGP</td>
<td>CSIBLINGP</td>
</tr>
</tbody>
</table>

3.3 Case Study and Analysis

In this section, an example of how to calculate the semantic distance between two concepts in an ontology by means of the defined equations is given. Fig. 1 shows a fragment of an OWL travel ontology visualized using Protégé-4.1 [23] with an OWL plug-in.

4. Semantic Matchmaking of Service Capabilities

The service capabilities, which are described in the Service Profile part of an OWL-S description, are divided in three logical parts [7]; the first is Actor which records information about service provider. The second is Functional Attributes which describes some service parameters such as Quality Rating and Geographic Radius. The last part represents the Functional Description of the service. It describes services capabilities in terms of inputs, outputs, preconditions and effects. In this paper, we consider the functional semantic distance between two services i.e., the semantic distance between their input/output concepts.

Eq. (1) is applied to calculate the matching degree between two sets of concepts, expressed in OWL, as follows:

\[
MatchD(S_1, S_2) = \begin{cases} 
\max_{c_1 \in S_1} \min_{c_2 \in S_2} (\sigma(c_1, c_2)) & \forall S_1 \neq S_2 \\
0 & \forall S_1 = S_2 
\end{cases}
\]

In an OWL-S description, a service could have a set of input concepts and a set of output concepts. Thus, our proposed equation for semantic matching of two SWS capabilities consists first in calculating the match degree (using previous equation) between the set of requested input concepts \(Query_{in}\) and the set of advertised input concepts \(Adv_{in}\), then between the set of requested output concepts \(Query_{out}\) and the set of advertised output concept \(Adv_{out}\). Second, the average of the two matching degrees is calculated. This is shown in the following equation:

\[
\text{SemD}(\text{SWS}_1, \text{SWS}_2) = \frac{\text{MatchD}(\text{Query}_{in}, \text{Adv}_{in}) + \text{MatchD}(\text{Query}_{out}, \text{Adv}_{out})}{2}
\]

5. Conclusions and Future Work

In this paper, we presented a novel method to calculate the semantic distance between two OWL concepts in a given taxonomy. Our proposed approach relies on the calculation of the edges’ weights along the path between concepts in the related ontology. In order to determine these weights, we consider some interesting properties such as the local density and the depth of a node in the hierarchy. The first parameter reflects the density effect on the degree of similarity between adjacent concepts like parent/child concepts and sibling concepts. In fact, the greater the density the more similar are the adjacent concepts. The second parameter which is the depth of concept (or node), expresses that the deeper are the adjacent concepts in the ontology the more similar they are because differentiation in description between them becomes finer.

We then compared our proposition with the one presented in Ref. [13] and showed that we obtain better results in terms of granularity and efficiency. In order to be able to compute the semantic distance between two OWL-S annotated services, we also proposed a matchmaking equation of services capabilities based on inputs/outputs concepts of the requested and advertised services.

Our future work is to implement a framework for SWS discovery based on our presented semantic
matchmaking approach and to evaluate other experimental measures like precision, recall and response time. We also intend to consider precondition and effect attributes in the matchmaking equation to enhance the accuracy of the discovery process.

References