Cardoso, J., "Discovering Semantic Web services with and without a Common Ontology Commitment" The 3rd International Workshop on Semantic and Dynamic Web Processes (SDWP 2006), In conjunction with the 2005 IEEE International Conference on Web Services (ICWS 2006), September 18-22, 2006, Chicago, USA. pp. 183-190, IEEE Computer Society. ISBN: 0-7695-2681-0

Discovering Semantic Web services with and without a Common Ontology Commitment

Jorge Cardoso
Department of Mathematics and Engineering
University of Madeira, Portugal
jcardoso@uma.pt

Abstract

This paper presents an algorithm to match a semantic Web service request against semantic Web service advertisements. The algorithm is to be used by systems to discover semantic Web services, such as the UDDI. Matching is based on the assessment of the similarity among semantic Web service properties, such as inputs and outputs. Semantic Web services have their inputs and outputs annotated or described by ontological concepts. The algorithm is able to match a semantic Web service request against advertisements that are annotated with concepts with and without a common ontology commitment. The similarity of inputs and outputs is evaluated based on concepts (classes), their semantic relations, and their common and distinguishing features (properties).

1. Introduction

With the proliferation of Web services and the evolution towards the Semantic Web comes the opportunity to automate various Internet related tasks. Applications should be able to automatically or semi-automatically discover, invoke, compose, and monitor Web services offering particular services and having particular properties [1].

Given the dynamic environment in e-businesses, the power of being able to discover Web services on the fly, to dynamically create business processes is highly desirable. The discovery of Web services has specific requirements and challenges compared to previous work on information retrieval systems and information integration systems. Several issues need to be considered. The discovery has to be based, not only on syntactical information, but also on data, as well as functional and QoS semantics [2].

Discovery is the procedure of finding a set of appropriate Web services that meets user requirements [3]. The discovery of Web services to model Web processes differs from the search of tasks/activities to model traditional processes, such as workflows. One of the main differences is in terms of the number of Web services available to the composition process. In the Web, potentially thousands of Web services are available which make the discovery a difficult procedure. One cannot expect a designer to manually browse through all the Web services available and select the most suitable one. Therefore, one of the problems that needs to be overcome is how to efficiently discover Web services [2].

Currently, the industry standards available for registering and discovering Web services are based on the Universal Description Discovery and Integration (UDDI) specification [4]. Unfortunately, discovering Web services using UDDI is relatively inefficient since the discovery mechanism only takes into account the syntactic aspect of Web services by providing an interface for keyword and taxonomy based searching.

The key to enhance the discovery of Web services is to describe Web services semantically [5] and use semantic matching algorithms (e.g. [2, 6-8]) to find appropriate services. Semantic discovery allows the construction of queries using concepts defined in a specific ontological domain. By having both the advertisement description and request query explicitly declare their semantics, the results of discovery are more accurate and relevant than keyword or attribute-based matching.

The algorithms that enable the discovery of semantic Web services generally use a semantic similarity distance function. Similarity is a judgment process that requires two semantic Web services to be decomposed into aspects in which they are the same and aspects in which they are different. Examples of aspects that can be used to determine if two Web services are similar include their inputs, outputs, and functionality, with and without a common ontology commitment.

This paper describes a semantic matching algorithm based on a feature-based model that determines the matching distance among two semantic Web services using a similarity function in terms of common and different features of the ontological concepts that specify the Web services input and output.

The remainder of this paper is structured as follows: Section 2 gives a brief overview on how Web services can be semantically annotated or described so that they can be considered semantic Web services. In section 3, we present our semantic Web service matching function to discover services. Section 4 describes a ranking algorithm that uses the matching function previously presented and that can be used by discovery mechanisms. Section 5 discusses the related work in this area and section 6 presents our conclusions.

2. Enhancing Web services using semantics

It has been recognized [1] that due to the heterogeneity, autonomy and distribution of Web services and the Web itself, new approaches should be developed to describe and advertise Web services. The most notable approaches rely on the use of semantics to describe Web services. This new breed of Web services, termed semantic Web services, will enable the automatic annotation, advertisement, discovery, selection, composition, and execution of inter-organization business logic, making the Internet become a common global platform where organizations and individuals communicate with each other to carry out various commercial activities and to provide value-added services. Academia has mainly approached this area from the semantic Web side, while industry is beginning to consider its importance from the point of view of Web services [9]. Three main approaches have been developed to bring semantics to Web services: WSDL-S, OWL-S, and WSMO.

WSDL-S. One approach to creating semantic Web services is by mapping concepts in a Web service description (WSDL specification) to ontological concepts. This approach is termed WSDL-S [10]. The idea of establishing mappings between service, task, or activity descriptions and ontological concepts was first proposed in [2]. By this approach, users can explicitly define the semantics of a Web service for a given domain. With the help of ontologies, the semantics or the meaning of service data and functionality can be explained. As a result, integration can be accomplished in an automated way and with a higher degree of success. The WSDL elements that can be marked up with metadata are operations, messages, preconditions and effects, since all the elements are explicitly declared in a WSDL description.

- **Operations**. Each WSDL description may have a number of operations with different functionalities. In order to add semantics, the operations must be mapped to ontological concepts to describe their functionality.
- **Message**. Message parts, which are input and output parameters of operations, are defined in WSDL using the XML Schema. Ontologies – which are more expressive than the XML Schema – can be used to annotate WSDL message parts.
- **Preconditions and effects**. Each WSDL operation may have a number of preconditions and effects. The preconditions are usually logical conditions, which must be evaluated to true in order to execute a specific operation. Effects are changes in the world that occur after the execution of an operation.

OWL-S. OWL-S (formerly DAML-S) is emerging as a description language that semantically describes Web services using OWL ontologies. OWL-S consists of three parts expressed with OWL ontologies: the service profile, the service model, and the service grounding. The profile is used to describe "what a service does", with advertisement and discovery as its objective. The service model describes "how a service works", to enable invocation, enactment, composition, monitoring and recovery. Finally, the grounding maps the constructs of the process model onto detailed specifications of message formats and protocols

WSMO. The third approach, Web Services Modeling Ontology (WSMO), provides ontological specifications for the description of semantic Web services. WSMO has been developed by the Digital Enterprise Research Institute (DERI), a European research organization that targets the integration of the semantic Web with Web services. The WSMO approach is based on the Web Services Modeling Framework (WSMF) [11], a framework that provides the appropriate conceptual model for developing and describing Web services and their composition based on the maximal de-coupling and scalable mediation service principles. The main objective of WSMO is to solve the application integration problem for Web services, Enterprise Application Integration (EAI), and Service-Oriented Architectures (SOA), by providing a conceptual framework and a formal language for semantically describing all relevant aspects of Web services. These technologies will facilitate the automation of discovering, interoperating, composing, and invoking Web services over the Web.

The algorithm presented in this paper can be easily used to discover semantic Web services defined with WSDL-S, OWL-S, and WSMO. For reasons of simplicity we will restrict our focus on semantic Web service input and output parameters. Please note that the algorithm can be easily adapted to match functional and operational semantics [2], and the preconditions and effects [10]of semantic Web services.

3. Matching Algorithm for Semantic Web services

This section presents an algorithm for matching semantic Web services. The algorithm presented computes the degree of match between two output and two input concepts, of a service request and advertisement, represented by an ontology.

We exploit the fact that the input and output concepts which are match may have (in addition to their name) properties (e.g., in the form of attributes) associated with them, and we also take into account the level of generality (or specificity) of each concept within the ontology as well as their relationships with other concepts. Notice that in contrast to semantic-based matching, syntactic-based matching cannot use this information.

Matching input and output concepts differs slightly from calculating their semantic similarity. One difference is that the functions to compute the semantic similarity of ontological concepts are usually symmetric, while matching functions are asymmetric. For example, let us assume that the ontology from Figure 1 is used to semantically annotate or describe a set of Web services.

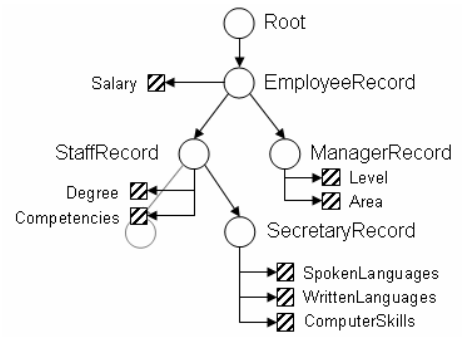

Figure 1. Example of an ontology used to semantically annotate a set of Web services.

Let us assume that we have a semantic Web service request *R* with the input concept *StaffRecord* (c_1) and an advertisement *A* with the input concept *EmployeeRecord* (c_2). In this scenario, request *R* matches advertisement *A* (i.e., $match(c_1, c_2)=true$), since *StaffRecord* is a subclass of *EmployeeRecord*. Our rationale is that if *A* is able to deal with the input *EmployeeRecord* is must also be able to deal with the input *StaffRecord*. We can think that when the Web service is invoked there will be some kind of cast (as in C programming) from *StaffRecord* to *EmployeeRecord*.

Now, let us assume that we have a semantic Web service request *R* with the input concept *EmployeeRecord* (c_2) and an advertisement *A* with the input concept *StaffRecord* (c_1). In this scenario, it is possible that the semantic Web service *A* cannot be invoked with the input *EmployeeRecord* since *A* may need properties that only exist in the class *StaffRecord*. Therefore, $match(c_2, c_1)=false$. As we can see from this two scenarios, the function *match* is asymmetric, since $match(c_1, c_2) \neq match(c_2, c_1)$.

3.1. Formal definition of a semantic Web service

Since we only deal with the input and output parameters of semantic Web services, we define a Web service as a finite sequence of ontological concepts,

$$sws(c_i, c_o).$$

The number of elements can be different than 2 if we consider more or fewer concepts to be used in a match. As we have mentioned before, the functionality and QoS of Web services [2] can also be considered when matching requests with advertisements.

4.2. Comparing semantic Web services with a common ontology commitment

In this scenario, Web service input and output concepts (c_i and c_o) are related to one global and unique ontology providing a common vocabulary for the specification of semantics. Comparing a concept with the ontology is translated into searching for the same or similar concepts within the ontology.

There are several functions that can be adapted and used to compute the degree of match of two input or output concepts belonging to the same ontology. The following four main techniques have been identified [12]:

1. **Ontology based approaches**. These approaches [13-15] use an ontology and evaluate the semantic relations amount concepts. The most basic metric simply computes the distance between two concepts in an ontology.
2. **Corpus based approaches**. These approaches [16-18] use a corpus to establish the statistical co-occurrence of words. The rationale is that if two words constantly appear together we may conclude that some relation exists between them.
3. **Information theoretic approaches**. These approaches [19-22] consider both a corpora and an ontology, and use the notion of information content from the field of information theory. By statistically analyzing corpora, probabilities are associated to concepts based on word occurrences. The information content for each concept is computed in such a way that infrequent words are more informative than frequent ones. Knowing the information content of concepts it is possible to calculate the semantic similarity between two given concepts.

4. **Dictionary based approaches**. These approaches [23, 24] use a machine readable dictionary to discover relations between concepts. For example, one approach determines the sense of a word in a given text by counting the overlaps between dictionary definitions of the various senses.

Most of these approaches are not suitable to compute the degree of matching between input and output concepts of the semantic Web services. All these metrics are symmetric (except [20]). This mean that $f(c_1, c_2) = f(c_2, c_1)$. As explained previously, when matching inputs and outputs the matching function needs to be asymmetric.

Furthermore, ontology-based approaches are rather limited since only the taxonomy of the ontology is used to find similarities between concepts. Corpus and dictionary-based approaches require associating a probability with each concept and finding a specific meaning of a word according to the context it is found in a dictionary, respectively. These approaches are not simple to implement for Web services. Questions raised include which corpus and dictionaries to use and how to deal with the heterogeneity of domains of discourse of Web services.

In our opinion, Tversky's model [20] is the most suitable approach to match semantic Web services. This model has been considered one of the most powerful similarity models to date [25]. It is also known as a feature-counting metric or feature-contrast model. This model is based on the idea that common features tend to increase the perceived similarity of two concepts, while feature differences tend to diminish perceived similarity. The model takes into account the features that are common to two concepts and also the differentiating features specific to each. More specifically, the similarity of a concept c_1 to a concept c_2 is a function of the features common to c_1 and c_2, those in c_1 but not in c_2 and those in c2 but not in c_1. For instance, a SUV (Sport Utility Vehicle) and a sedan are similar by virtue of their common features, such as wheels, engine, steering wheel, and gears, and are dissimilar by virtue of their differences, namely height and the size of the tires.

Based on Tversky's model, we introduce the matching functions $S_i^{=}(c_R, c_A)$ and $S_o^{=}(c_R, c_A)$ which analyze the number of properties shared among two input or output concepts c_R and c_A (R stands for a Web service request, A stands for a Web service advertisement, i stands for input, and o stands for output) conceptualized within the same ontology. In our functions $S^{=}$, the function $p(c)$ retrieves all the properties associated with a concept c and function $|s|$ corresponds to the number of elements in set s.

$$S_i^{=}(c_R, c_A) = \begin{cases} 1, & c_R = c_A \\ 1, & c_R > c_A \\ \dfrac{|p(c_R)|}{|p(c_A)|}, & c_R < c_A \\ \dfrac{|p(c_R) \cap p(c_A)|}{|p(c_A)|}, & c_R \neq c_A \end{cases}$$

$$S_o^{=}(c_R, c_A) = \begin{cases} 1, & c_R = c_A \\ \dfrac{|p(c_A)|}{|p(c_R)|}, & c_R > c_A \\ 1, & c_R < c_A \\ \dfrac{|p(c_R) \cap p(c_A)|}{|p(c_R)|}, & c_R \neq c_A \end{cases}$$

Since functions $S_i^{=}(c_R, c_A)$ and $S^{=}(c_R, c_A)$ are very similar we will only describe function $S_i^{=}$. Four distinct cases can occur:

Case 1: In the first case, since the two input concepts are equal (c_R=c_A) their similarity is maximal and therefore the degree of match is one.

Case 2: In the second case, the concept c_R is a specialization of concept c_A (c_R>c_A). As a result, a Web service with input concept c_A, is able to process concept c_R. For example, let us consider the ontology from Figure 1. If a Web service request specifies concept *StaffRecord* as input and an advertisement specifies concept *EmployeeRecord* as input then the advertised service is able to process the input concept *StaffRecord.* This is because the concept c_R is a subclass of concept c_A and it has at least the same set of properties as c_A. In this case, the similarity is also one.

Case 3: In the third case, if the request concept c_R is a generalization of advertisement concept c_A (c_R<c_A), then c_A has probably some properties that do not exist in c_R. Therefore, it is possible that a Web service advertisement with input c_A is not able to process the input concept c_R due possibly to missing properties. For example, if a Web service request R specifies concept *EmployeeRecord* as input and an advertisement A specifies concept *StaffRecord* as input then Web service A may not be able to process the input concept *EmployeeRecord.* This is because A may need the property *Degree* and *Competencies* of the input concept to work properly.

Case 4: In the last case, the concepts c_R and c_A are not equal and do not subsume each other in any way ($c_R \neq c_A$). In this scenario, we evaluate the matching by analyzing how many common properties exist between the two concepts and how many properties are different. Also, we analyze the percentage of input advertisement properties that were satisfied.

As an example, let us illustrate the use of function $S_i^{=}(c_R, c_A)$ for the four cases – 1), 2), 3) and 4) – that can occur when matching a request c_R with an advertisement c_A. In our example, the Web services' input is annotated with concepts from the ontology illustrated in Figure 1. The four cases that may occur are listed in Table 1 and are evaluated as follows:

- In case 1), both c_R and c_A are associated with the same concept (*StaffRecord*). Since the request matches the advertisement perfectly. The result is 1.
- In case 2), the request c_R is associated with the concept *StaffRecord* and the advertisement c_A is associated with the concept *EmployeeRecord*. Since the concept *EmployeeRecord* is a generalization of concept *StaffRecord*, the properties of the concept *StaffRecord* (the set {*Salary, Degree, Competencies*}) is a superset of the properties of the concept *EmployeeRecord* (the set {*Salary*}). All the properties of c_A exist in c_R. As a result, the similarity is evaluated to 1.
- In case 3), the request c_R is associated with the concept *StaffRecord* and the advertisement c_A is associated with the concept *SecretaryRecord*. Since the concept *StaffRecord* is a subclass of concept *SecretaryRecord*, the properties of the concept *StaffRecord* (the set {*Salary, Degree, Competencies*}) is a subset of the properties of the concept *SecretaryRecord* (the set {*Salary, Degree, Competencies, SpokenLanguage, WrittenLanguage, ComputerSkills*}). In this case, when the request c_R matches the advertisement c_A some properties of c_A are left unfulfilled (the properties *SpokenLanguage, WrittenLanguage,* and *ComputerSkills*). To indicate this mismatch the matching is set to the ratio of the number of properties of c_R and the number of properties of c_A, which in this case is $|p(c_R)|/|p(c_A)| = 3/6 = 0.5$.
- In the last case (4), the request c_R is associated with the concept *StaffRecord* and the advertisement c_A is associated with the concept *ManagerRecord*. The concept *StaffRecord* has the set of properties {*Salary, Degree, Competencies*} and the concept *ManagerRecord* has the set of properties {*Salary, Level, Area*}. Since the concepts do not have a parent/children relationship, we compute the percentage of the advertisement's properties that are fulfilled with a property from c_R. The similarity is evaluated as follows:

$$S_i^{=}(c_R, c_A) = \frac{|\, p(c_R) \cap p(c_A)\,|}{|\, p(c_A)\,|} = \frac{1}{3}$$

The result of evaluating the function indicates a low degree of matching between the concepts *StaffRecord* and *ManagerRecord*. Only one of the three advertisement properties are satisfied by request properties. The following table shows the results for the four cases presented.

Request c_R	**Advertisement c_A**	$S_i^{=}(c_R, c_A)$
StaffRecord	StaffRecord	1
StaffRecord	EmployeeRecord	1
StaffRecord	SecretaryRecord	0.5
StaffRecord	ManagerRecord	1/3

Table 1. An example of matching inputs with a common ontology commitment.

As we can see the concept *SecretaryRecord* is closer to the concept *StaffRecord* than the concept *ManagerRecord*. This result corroborates our intuition and visual analysis of the ontology and its concepts.

3.3. Comparing semantic Web services with no common ontology commitment

In this scenario, different Web services are described by different ontologies. Since there is no common ontology commitment, there is no common vocabulary which makes the comparison of different concepts a more complicated task.

Web service parameters (such as inputs and outputs) are identified by words (classes) and there are two major linguistic concepts that need to be considered: synonymy and polysemy. Polysemy arises when a word has more than one meaning (i.e., multiple senses). Synonymy corresponds to the case when two different words have the same meaning. To tackle the existence of these linguistic concepts we will use a feature-based similarity measure that compares concepts based on their common and distinguishing features (properties).

The problem of determining the similarity of concepts defined in different ontologies is related to the work on multi-ontology information system integration. Most of the similarity measures previously presented [13-19, 21-24] cannot be directly used to match Web services since they are symmetric, and more importantly, they can only be used when the concepts to compare are defined in the same ontology.

Nonetheless, the Tversky's feature-based similarity model [20] is interesting since it takes into account the features or properties of concepts and not the taxonomy that defines the hierarchy of concepts. We believe that when matching inputs and outputs, the features of concepts tell more than the taxonomy.

Based on Tversky's model, we introduce matching functions $S_i^{\neq}(c_R, c_A)$ and $S_o^{\neq}(c_R, c_A)$ for semantic Web services with no common ontology commitment based on the number of properties shared among two input or output concepts c_R and c_A conceptualized within the same ontology. The function computes the geometric distance between the similarity of the domains of concept c_R and concept c_A and the ratio of matched input properties from the concept c_A. Our similarity functions are defined as follows,

$$S_i^{\neq}(c_R, c_A) = \sqrt{\frac{\Pi(p(c_R), p(c_A))}{|p(c_R) \cup p(c_A)| - \Pi(p(c_R), p(c_A))} * \frac{\Pi(p(c_R), p(c_A))}{|p(c_A)|}}$$

$$S_o^{\neq}(c_R, c_A) = \sqrt{\frac{\Pi(p(c_R), p(c_A))}{|p(c_R) \cup p(c_A)| - \Pi(p(c_R), p(c_A))} * \frac{\Pi(p(c_R), p(c_A))}{|p(c_R)|}}$$

Function *Π* establishes a mapping between the properties of two concept classes. Figure 2 illustrates two ontologies involved in a mapping.

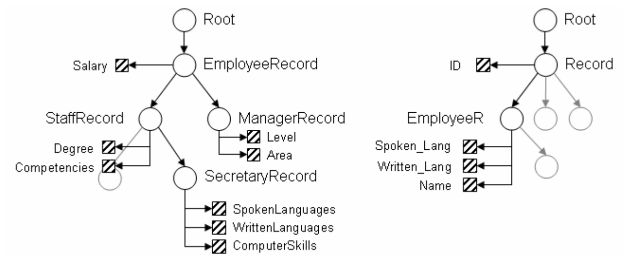

Figure 2. Two ontologies involved in a mapping

For example, when matching the class concepts *SecretaryRecord* and *EmployeeR* we need to establish a mapping between the properties of the two classes. The mapping is computed with function *Π*(*p*(*SecretaryRecord*), *p*(*EmployeeR*)), which is equivalent to *Π*({*Salary, Degree, Competencies, SpokenLanguages, WrittenLanguages, ComputerSkills*}, {*ID, Spoken_Lang, Written_Lang, Name*}). Possible mappings that can be established are the following:

Π$_{i,1}$: (SpokenLanguages , Spoken_Lang)
Π$_{i,2}$: (WrittenLanguages , Written_Lang)
Π$_{i,3}$: (Name, ComputerSkills)

Function *Π* establishes the best mapping between two sets of properties and it is defined as follows:

$$\Pi(pl_1, pl_2) = \begin{cases} Max(\Pi(pl_1 - p_1, pl_2 - p_2) + ss(p_1, p_2)), & ss(p_1, p_2) = 1, \\ & pl_1 \neq \varnothing \wedge pl_2 \neq \varnothing \\ \Pi(pl_1 - p_1, pl_2 - p_2), & ss(p_1, p_2) = 0, \\ & pl_1 \neq \varnothing \wedge pl_2 \neq \varnothing \\ 0, & pl_1 = \varnothing \vee pl_2 = \varnothing \end{cases}$$

Function *ss*(*p$_1$*, *p$_2$*) determines if two properties are considered to be equal using function *g*. If two properties match syntactically then function *ss* returns 1, otherwise it returns 0. Properties match syntactically only if function *g* determines that the syntactic similarly it greater that a constant *β*.

$$ss(p_1, p_2) = \begin{cases} 1, & g(p_1, p_2) \geq \beta \\ 0, & g(p_1, p_2) < \beta \end{cases}$$

Function *g*(*p$_1$*, *p$_2$*) is a function that computes the syntactic similarity of two words. In our approach, we use "string-matching" as a way to calculate similarity. Function *g* can be implemented using several existing methods such as equality of name, canonical name representations after stemming and other preprocessing, *q-grams*, synonyms, similarity based on common sub-strings, pronunciation, soundex, abbreviation expansion, stemming, tokenization, etc. Other techniques borrowed from the information retrieval area may also be considered. A very good source of information retrieval techniques can be found in Belew [26].

For example, let us consider the request query *sws$_R$*(*"SecretaryRecord", c$_{Ro}$*) and the advertisement *sws$_A$*(*"EmployeeR", c$_{Ao}$*). When computing *Π*(*p*(*"SecretaryRecord"*), *p*(*"EmployeeR"*)) of the inputs we obtain two mappings *Π$_{i,1}$* and *Π$_{i,2}$*. The mapping *Π$_{i,1}$* is found since the results of *ss*(*"SpokenLanguages", "Spoken_Lang"*), using the *q-grams* methodology [27] as an implementation of *g* with *β* = 0.5, is 0.53 (i.e., *g*(*"SpokenLanguages", "Spoken_Lang"*)=0.53). As a result, *ss* is evaluated to 1. Mapping *Π$_{i,2}$* yields because *ss*("*WrittenLanguages" ," Written_Lang"*) is 1. The mapping *Π$_{i,3}$* is not part of *Π* since *ss*("*Name", "ComputerSkills"*) is evaluated to zero. All the other mappings are not part of *Π*. For example, if we compute *ss*(*"SpokenLanguages"*, "*Written_Lang*") we obtain a result of 0 (function *g* has a value of 0.13), which means that we do not consider the properties to be syntactically equal. The result of computing $S_i^{\neq}(c_R, c_A)$ is:

$$\sqrt{\frac{2}{6-2} * \frac{2}{4}} = \frac{2}{4} = 0.5$$

This result corroborates our intuition since only two of the four properties of the concept *EmployeeR* are satisfied by the properties of concept *SecretaryRecord.*

4. Ranking algorithm

In this section we present the actual algorithm for ranking Web service advertisements, following the functions presented previously.

```
REQ($c_i$, $c_o$) = Web service request
ADV$_j$ ($c_{ji}$, $c_{jo}$) = List of advertisement

For all j get ADV$_j$($c_{ji}$, $c_{jo}$)
   If same_ontology($c_i$ , $c_{ji}$) i = $S_i^{=}(c_i, c_{ji})$
   else i = $S_i^{\neq}(c_i, c_{ji})$

   If same_ontology($c_o$ , $c_{jo}$) o = $S_o^{=}(c_o, c_{jo})$
   else o = $S_o^{\neq}(c_o, c_{jo})$

   match[j] = (i+o)/2;

Forall
Sort match[j]
```

The algorithm uses the function *same_ontology* that determines if two concepts are defined in the same ontology. Once the matching degree of the input and output between a Web service request and a Web service advertisement is calculated, we define the overall degree of the match as the arithmetic mean of the input match degree and output match degree.

5. Related Work

The OWL-S/UDDI Matchmaker [28] introduces semantic search into the UDDI directory by embedding an OWL-S Profile in a UDDI data structure, and augmenting the UDDI registry with an OWL-S matchmaking component. The matching algorithm recognizes four degrees of match between two concepts defined in the same ontology: (1) *exact*, (2) *plug in*, (3) *subsume*, and (4) *fail*. The function used by the algorithm is asymmetric and is based on the existence of relationships between concepts. When no direct relationship exists among two concepts the algorithm simple return *fail*. Unlike the algorithm presented in this paper, the OWL-S/UDDI Matchmaker searches for services based on inputs and outputs within the IOPEs of the profile which must belong to the same ontology. Our approach allows evaluating the similarities of IOPE that are annotated with concepts from distinct ontologies.

The METEOR-S [10] Web Service Annotation Framework (WSAF) allows semi-automatically matching WSDL concepts (such as inputs and outputs) to DAML and RDF ontologies using text-based information retrieval techniques (for example, synonyms, n-grams and abbreviation). The strength of matches (SM) is calculated using a scoring formula which involved element (ElemMatch) and structure level schema (SchemaMatch) matching. The ElemMatch function performs the element level matching based on the linguistic similarity of the names of the two concepts. The SchemaMatch function examines the structural similarity between two concepts. A concept in an ontology is usually defined by its properties, superclasses and subclasses. Since concept labels are somewhat arbitrary, examining the structure of a concept description can provide more insight into its semantics. In WSAF, the XML representation of WSDL is matched against the concepts of a given ontology. The best match between WSDL concepts and ontological concepts are returned to users as a suggestion of potential mappings. In our work we match ontological concepts with ontological concepts. It should be noticed that the work presented in [10] cannot be easily adapted to our problem. There are several reasons. First, the weight values for calculating the MS function were set without empirical testing and validation. Also, the weights are not defined for a set of ElemMatch and SchemaMatch values. For example, if $0.5<\text{ElemMatch}<0.65$ then no weights are suggested. Furthermore, the function that computes the ElemMatch of a WSDL concept and an ontological concept is not defined when the MatchScore is different than zero, but less than one, using the NGram or Synonym matching algorithms.

6. Conclusions

In this paper we have described a semantic matching algorithm to be used by UDDI registries enhanced with semantics. Our algorithm can work with Web services described with WSMO and OWL-S, or annotated with WSDL-S. Compared to previous work [28], we do not limit the classification of the accuracy of matching a request with an advertisement using a four value schema (i.e. *exact*, *plug in*, *subsume*, and *fail*). The accuracy of matching if assessed with a continue function with the range [0..1]. Furthermore, compared to [28], we allow the matching of semantic Web services with and without a common ontology commitment. This aspect is important since it is not realistic to assume that Web services will always be defined by the same ontology. In some case, similar services may be defined by different ontologies.

Our algorithm relies on the Tversky's feature-based similarity model to match requests with advertisement. This model takes into account the features or properties of ontological concepts and not the taxonomy that defines the hierarchy of concepts. We believe that when matching inputs and outputs, the features of concepts tell more than the taxonomy. The matching process that we are using so far is restricted to the inputs and outputs of Web services. Nevertheless, it can be easily extended to include functional and non-functional capabilities of services.

8. References

[1]. Cardoso, J. and A.P. Sheth, *Introduction to Semantic Web Services and Web Process Composition*, in *Semantic Web Process: powering next generation of processes with Semantics and Web services*, J. Cardoso and A.P. Sheth, Editors. 2005, Springer-Verlag: Heidelberg, Germany. p. 1-13.

[2]. Cardoso, J. and A. Sheth, *Semantic e-Workflow Composition.* Journal of Intelligent Information Systems (JIIS). 2003. **21**(3): p. 191-225.

[3]. Verma, K., et al., *METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services.* Journal of Information Technology and Management (in print), 2004.

[4]. UDDI, *Universal Description, Discovery, and Integration*. 2002.

[5]. Sheth, A. and R. Meersman, *Amicalola Report: Database and Information Systems Research Challenges and Opportunities in Semantic Web and Enterprises.* SIGMOD Record, 2002. **31**(4): p. pp. 98-106.

[6]. Rodríguez, A. and M. Egenhofer, *Determining Semantic Similarity Among Entity Classes from Different Ontologies.* IEEE Transactions on Knowledge and Data Engineering (in press). 2002.

[7]. Smeaton, A. and I. Quigley. *Experiment on Using Semantic Distance Between Words in Image Caption Retrieval*. in *19th Intemational Conference on Research and Development in Information Retrifval SIGIR'96*. 1996. Zurich, Switzerland.

[8]. Klein, M. and A. Bernstein. *Searching for Services on the Semantic Web Using Process Ontologies*. in *International Semantic Web Working Symposium (SWWS)*. 2001. Stanford University, California, USA.

[9]. Cardoso, J., et al., *Academic and Industrial Research: Do their Approaches Differ in Adding Semantics to Web Services*, in *Semantic Web Process: powering next generation of processes with Semantics and Web services*, J. Cardoso and S. A., Editors. 2005, Springer-Verlag: Heidelberg, Germany. p. 14-21.

[10]. Patil, A., et al. *MWSAF - METEOR-S Web Service Annotation Framework*. in *13th Conference on World Wide Web*. 2004. New York City, USA.

[11]. Fensel, D. and C. Bussler, *The Web Service Modeling Framework WSMF.* Electronic Commerce Research and Applications, 2002. **1**(2): p. 113-137.

[12]. Zavaracky, A., *Glossary-Based Semantic Similarity in the WordNet Ontology*, in *Department of Computer Science*. 2003, University College Dublin: Dublin.

[13]. Wu, Z. and M. Palmer. *Verb Semantics and Lexical Selection*. in *32nd Annual Meeting of the Associations for Computational Linguistics (ACL'94).* 1994. Las Cruces, New Mexico.

[14]. Rada, R., et al., *Development and Application of a Metric on Semantic Nets.* IEEE Transactions on Systems, Man, and Cybernetics, 1989. **19**(1): p. 17-30.

[15]. Leacock, C. and M. Chodorow, *Combining local context and WordNet similarity for word sense identification*, in *WordNet: An Electronic Lexical Database*, C. Fellbaum, Editor. 1998, MIT Press. p. 265-283.

[16]. Turney, P.D. *Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL.* in *12th European Conference on Machine Learning*. 2001: Springer-Verlag.

[17]. Keller, F. and M. Lapata, *Using the Web to Obtain Frequencies for Unseen Bigrams.* Computational Linguistics, 2003.

[18]. Church, K.W. and P. Hanks. *Word association norms, mutual information, and Lexicography*. in *27th. Annual Meeting of the Association for Computational Linguistics*. 1989. Vancouver, B.C.: Association for Computational Linguistics.

[19]. Lin, D. *An information-theoretic definition of similarity*. in *15th International Conf. on Machine Learning*. 1989. San Francisco, CA: Morgan Kaufmann.

[20]. Tversky, A., *Features of Similarity.* Psychological Review, 1977. **84**(4): p. 327-352.

[21]. Resnik, P. *Using Information Content to Evaluate Semantic Similarity in a Taxonomy*. in *14th International Joint Conference on Artificial Intelligence*. 1995.

[22]. Jiang, J. and D. Conrath. *Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy.* in *Intemational Conference on Computational Linguistics (ROCLINGX)*. 1997. Taiwan.

[23]. Lesk, M. *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone*. in *5th annual international conference on Systems documentation*. 1986: ACM Press.

[24]. Banerjee, S. and T. Pedersen. *Gloss Overlaps as a Measure of Semantic Relatedness*. in *Eighteenth International Joint Conference on Artificial Intelligence*. 2003. Acapulco, Mexico.

[25]. Richardson, R. and A. Smeaton, *Using WordNet in a Knowledge-Based Approach to Information Retrieval*. 1995, Dublin City University, School of Computer Applications: Dublin, Ireland.

[26]. Belew, R.K., *Finding Out About : A Cognitive Perspective on Search Engine Technology and the WWW.* 2000, Cambridge, U.K: Cambridge University Press. 356.

[27]. Salton, G., *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer.* 1988, Massachusetts: Addison-Wesley.

[28]. Srinivasan, N., M. Paolucci, and K. Sycara, *An efficient algorithm for OWL-S based semantic search in UDDI*, J. Cardoso and A. Sheth, Editors. 2005, Lecture Notes in Computer Science, Springer.