Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, Hong Kong, 2006.

A Similarity Measure for OWL-S Annotated Web Services

YASSER GANJISAFFAR, HASSAN ABOLHASSANI and MAHMOOD NESHATI
Semantic Web Research Laboratory, Computer Engineering Department
Sharif University of Technology, Tehran, Iran
ganji@ce.sharif.edu, abolhassani@sharif.edu, neshati@ce.sharif.edu

Abstract

Measuring semantic similarity of web services has several benefits and most of the proposed service discovery algorithms are based on measuring the similarity of the requested service with each of the advertised services. In this paper, we propose a method for measuring the similarity of web services which are annotated with OWL-S ontology. First, a semantic similarity measure for determining the similarity of OWL concepts is discussed and then based on this measure, the functional similarity of services is defined. Then it is showed that the precision of algorithms that only take into account the functional properties of services for measuring their similarity are low. Therefore the textual descriptions of web services are also taken into account and the textual similarity of services is also calculated. Then it is showed how Neural Networks can be used for combining these two measures for a better compound measure. The proposed technique is applied to a sample test collection and experimental results are presented which demonstrate the effectiveness of the idea.

I. Introduction

Web Services are loosely coupled software components that are published, located and invoked across the web. Recently, web services have gained an increasing popularity. Today, not only web services are used in dynamic business-to-business interactions, but they are also used in the area of business-to-consumer or even peer-to-peer interactions.

The Semantic Web approaches to web services give us the ability to describe the semantics of web services and their capabilities in a formal and machine-processable manner. The Semantic Web should enable greater access not only to content but also to services on the web. Users and software agents should be able to discover, invoke, compose, and monitor web resources offering particular services and having particular properties, and should be able to do so with a high degree of automation if desired. The focus of this paper is on using the semantic meta data which is published for web services for measuring the similarity of web services. The ability of measuring the similarity of web services has several use cases. The following is a brief description of some of these use cases:

- *Service Discovery*: Semantic web service discovery or matchmaking is an emerging research area that exploits the service semantic metadata to locate services that can perform tasks of given description with the best overall degree of satisfaction. Most of the proposed service discovery algorithms are based on measuring the similarity of the requested service with each of the advertised services. Therefore having a similarity measure for web services can be quite helpful in service discovery.
- *Service Recommendation*: It is often happens that a client spent some time for manually finding a service that can fulfil his requirements by browsing categories in a registry and then try to invoke the service but the service is unavailable. In this situation, similar services can be suggested to the client.
- *Service Composition*: Currently, many services are offered on the web and their number would increase exponentially in the future. But these atomic services may not be able to satisfy specific requests individually. So there must be brokers between service providers and service requesters which integrate the atomic services of service providers and create new value added services. The act of taking several services and bundling them together to meet the needs of a given customer is called service composition.

  A composite web service contains a set of services and a data flow between them. In the execution time, if any of these services fail to execute correctly then the execution of the composite web service fails. Being able to find similar services in run-time can help the execution engine to replace failed services with similar services and recover the execution of the composed service.

OWL-S [1] is the current standard for describing semantic metadata about web services, which is based on the OWL [2] ontology language. The OWL-S ontology is organized in three modules: the *Service Profile* module describes the functionality of the service; the *Service Model* module describes how it does it; and the *Service Grounding* module describes how to access the service. The semantics of inputs and outputs of a service are specified in the service profile section by mapping each input/output parameter to a concept which is defined in an OWL ontology. Therefore, in order to calculate the semantic similarity between inputs and outputs of two services the semantic similarity of their corresponding OWL concepts can be measured. In section II, we describe a method for calculating the degree of semantic similarity between two OWL concepts. This method will then be used in section III for measuring the degree of functional similarity between two services. Section

IV shows how the textual descriptions that are provided in service descriptions can be extracted and be used as an alternative measure of similarity of services. In section V, we describe how the functional and textual similarity measures can be combined to provide a compound measure for semantic similarity of web services. In section VI, the experimental results of the proposed measure are studied on a sample test collection. The related works on this topic are explored in section VII and finally a conclusion is given in the last section.

II. Concept Similarity

In this section, we describe a method for measuring the degree of similarity of two OWL concepts. This measure will then be used in next section for determining the degree of functional similarity of two services.

Definition 1 (Concept Similarity): A similarity $\sigma : \mathcal{C} \times \mathcal{C} \rightarrow [0, 1]$ is a function from a pair of concepts to a real number between zero and one expressing the degree of similarity between two concepts such that:

1) $\forall x \in \mathcal{C},\ \sigma(x, x) = 1$
2) $\forall x, y \in \mathcal{C},\ \sigma(x, y) = \sigma(y, x)$

The following two subsections describe two different methods for calculating the semantic similarity of OWL concepts:

A. Taxonomy based Similarity

Ontologies define properties of concepts and relationships between them. One of such relationships is the *IS–A* relationship. We use this relationship for extracting concept taxonomies from ontologies. If $\mathcal{C}$ is the set of concepts defined in an ontology then the concept taxonomy $\mathcal{T}$ is defined as $\mathcal{T}\langle\mathcal{C}, \preceq\rangle$ such that $c \preceq c'$ means that c *IS–A* c' (concept c is subsumed by concept c').

In this section, we suppose that the two concepts that we are interested in measuring their similarity are defined in the same ontology and therefore are located in the same taxonomy. Whenever the two concepts that are to be compared are defined in different ontologies, these two ontologies must be merged with each other to form a single ontology which covers both of them (Fig 1). Then the taxonomy of concepts can be extracted from this ontology by considering the IS–A relations that exist between concepts. A lot of methods have been suggested in the literature for merging ontologies and discussing them is beyond the scope of this paper. [3], [4] explains about such methods.

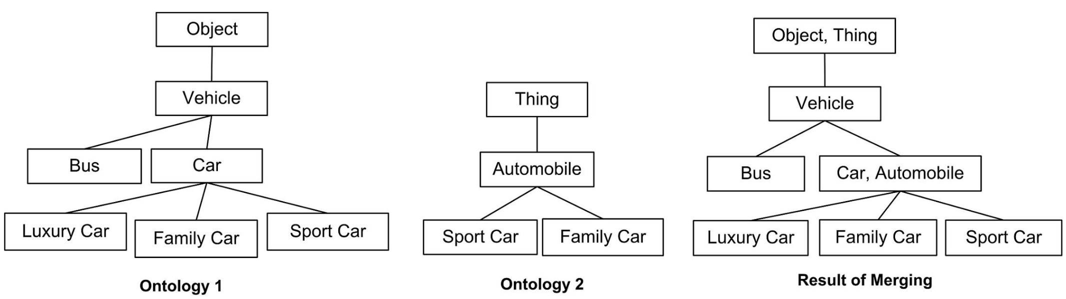

Fig. 1. Merging two ontologies into a single ontology

For calculating the degree of similarity of two concepts, we use a slightly modified version of the algorithm which is introduced in [5]. The concept similarity is defined as: $\sigma(c, c') = 1 - \delta(c, c')$, where $\delta(c, c')$ denotes the weighted distance of these two concepts in a taxonomy. A weight value $\omega(c)$ is assigned to each concept c in the concept taxonomy. Since the distance between two given concepts represents the path over their closest common parent, $ccp(c_1, c_2)$, it is calculated as the sum of the their distances to their closest common parent:

$$\delta(c_1, c_2) = [\omega(ccp(c_1, c_2)) - \omega(c_1)] + [\omega(ccp(c_1, c_2)) - \omega(c_2)]$$

The weight values of concepts in the taxonomy are calculated with the following formula:

$$\omega(n) = \frac{1}{k^{l(n)+1}}$$

where $l(n)$ is the length of the longest path from the root concept to node n in the taxonomy and k is a predefined factor larger than 1 indicating the rate at which the weight values decrease along the taxonomy (currently we set k to 2). This formula has two desirable properties: the semantic differences between upper level concepts are higher than those between lower level concepts (in other words: two general concepts are less similar than two specialized ones) and that the distance between sibling

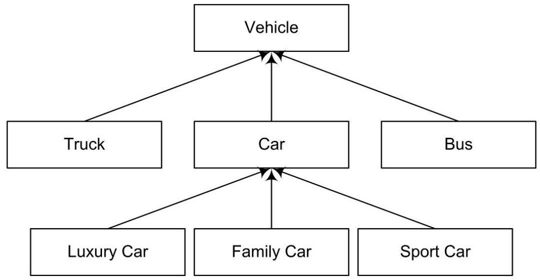

Fig. 2. An example concept taxonomy

concepts is greater than the distance between parent and child concepts. For example, in Fig. 2, similarity degree between *Truck* and *Sport Car* can be calculated as follows:

$$\begin{aligned}
\omega(Vehicle) &= 0.5 \\
\omega(Truck) &= 0.25 \\
\omega(SportCar) &= 0.125 \\
\delta(Truck, Vehicle) &= 0.5 - 0.25 = 0.25 \\
\delta(SportCar, Vehicle) &= 0.5 - 0.125 = 0.375 \\
\delta(Truck, SportCar) &= 0.25 + 0.375 = 0.625 \\
\sigma(Truck, SportCar) &= 1 - 0.625 = 0.375
\end{aligned}$$

For calculating the closest common parent of two concepts, first their common parents are determined as follows:

$$PS(c, T) = \{c' \in T; c \preceq c'\}$$

$$CP(c_1, c_2) = PS(c_1, T) \cap PS(c_2, T)$$

In the above formulae, $PS(c, T)$ is the set of parents of concept c derived from taxonomy T, and $CP(c_1, c_2)$ is the set of the common parents of concepts c_1 and c_2. The closest common parent of these two concepts is the concept in $CP(c_1, c_2)$ with the smallest ω value (this is because that the concept with the smallest ω value has the longest path to the root of the taxonomy and hence it is the most specific concept in the set):

$$x = ccp(c_1, c_2) \Leftrightarrow x \in CP(c_1, c_2) \wedge \forall y \in CP(c_1, c_2); \omega(y) \geq \omega(x)$$

B. Feature based Similarity

In the previous method, for calculating semantic similarity of concepts, only the taxonomy structure is taked into account and other features of the concepts are ignored. However, the features of a concept contain valuable information about that concept and therefore a good measure for semantic similarity of OWL concepts can be defined to measure the degree of similarity of their features. Each OWL concept can have a number of *datatype properties* and a number of *object properties*. These properties somehow determine the characteristics of the concept. Therefore we call these properties as the features of the concept.

For example, a concept `Person` can have a data type property `hasName` and another concept `Father` can have a `subClassOf` relation with this concept. `Father` can also have a `hasChild` relation with `Person`. The feature sets of these two concepts would be as follows:
$F(Person) = \{(type, Class), (hasName, String)\}$
$F(Father) = \{(type, Class), (hasName, String), (subClassOf, Person), (hasChild, Person)\}$
Note that, because of being a sub class of `Person`, concept `Father` has inherited the `hasName` property from `Person`.

The more *common features* two concepts have and the less *non-common features* they have, the more similar they are. The following is a semantic similarity measure based on the measure defined in [6] which takes into account the feature sets of concepts:

$$\sigma(c_1, c_2) = \frac{2 \times |F(c_1) \cap F(c_2)|}{|F(c_1) \cap F(c_2)| + |F(c_1) \cup F(c_2)|}$$

The success of this measure depends on the degree to which the features of concepts are specified in ontologies. In most of the current available ontologies only the subsumption relation between concepts are specified in the ontology and the other relations are neglected. In these kinds of ontologies, the feature based similarity measure can not be a useful measure and the results are often not good enough.

III. Functional Similarity

A good measure for calculating the degree of similarity of two services is the degree of similarity between their functional properties such as their inputs and outputs. In this section, we describe a method for calculating the degree of input/output similarity of services based on the semantic similarity measure that we introduced in the previous section.

Each service has a set of inputs and a set of outputs and each input/output parameter is semantically annotated with mapping it to an OWL concept. Therefore, the problem here is to calculate the similarity of two sets of concepts. Using one of the taxonomy or feature based similarity measures of the previous section (or a weighted combination of them), we are able to calculate the similarity of each pair of concepts in these two sets . Then the best mapping between the elements of these two sets of concept must be found. This can be shown by a bipartite graph in which each part shows one of the sets of concepts and edges show the similarity of concepts that are connected to them. Fig. 3 shows one such mapping.

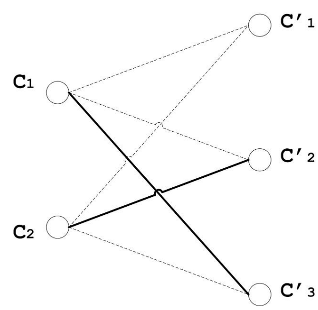

Fig. 3. A Mapping between inputs of two services

Function $\Pi(C_1, C_2)$ which is defined below computes the best mapping that can be obtained between concepts of sets C_1 and C_2 (the mapping that maximizes the overall degree of similarity of pairs):

$$\Pi(C_1, C_2) = \begin{cases} \frac{\max_{c_1 \in C_1, c_2 \in C_2} \{\Pi(C_1 - c_1, C_2 - c_2) + \sigma(c_1, c_2)\}}{\min\{|C_1|, |C_2|\}}, & C_1 \neq \phi, C_2 \neq \phi \\ 0, & C_1 = \phi \vee C_2 = \phi \end{cases}$$

By using this recursive function, we can determine the similarity of input sets of two services, $\Pi(I_1, I_2)$, and also the similarity of output sets of these two services, $\Pi(O_1, O_2)$. Then the functional similarity of the two services can be defined as the average of these values:

$$FunctionalSim(S_1, S_2) = \frac{\Pi(I_1, I_2) + \Pi(O_1, O_2)}{2}$$

IV. Text based Similarity

Sometimes two similar services have inputs and outputs that are mapped to different concepts in the reference ontology. In these situations, using the pure functional similarity measure of the previous section would fail and these two services would not be detected as similar. Therefore, it is needed to combine another similarity measure with the functional similarity measure for better results.

Service descriptions often contain parts which include textual information and the similarity of terms which are in these parts of the service descriptions can be used as an alternative measure for similarity of services. Information retrieval methods for finding similar documents can be used here. Vector model [7] is a well known method in this area. Using this model, services can be represented as vectors $\overrightarrow{s_k} = (w_{1k}, w_{2k}, \ldots, w_{nk})$ where w_{ik} is a weight for term i in the description of service s_k. Then similarity of services would be the cosine of the angle between their corresponding vectors. Terms are extracted from textual parts of the service description after removing common words and extracting the base form[1] of them using WordNet [8]. Then the *tf–idf* [7] weighting method is used for determining weight of each term. According to this weighting method, the importance of a term i in document j is:

$$w_{ij} = tf_{ij} \times idf_i = tf_{ij} \times \log(\frac{N}{n_i})$$

[1]For example *Reservation* would be converted to *Reserve*

where tf_{ij} is the frequency of term i in document j; N is the number of documents in the collection; and n_k is the number of the documents in the collection that contain term i. Textual similarity of services would be calculated as follows:

$$TextSim(S_1, S_2) = \frac{\overrightarrow{S_1} \cdot \overrightarrow{S_2}}{|\overrightarrow{S_1}||\overrightarrow{S_2}|}$$

V. Compound Similarity

Our experimental results which are presented in the next section, show that neither the functional similarity nor the textual similarity measure individually can have a good precision and it is better to combine these two measures for reaching to a better similarity measure. We used a linear combination of these two measures as follows:

$$Sim(S_1, S_2) = \omega_f.FunctionalSim(S_1, S_2) + \omega_t.TextSim(S_1, S_2)$$

$$\omega_f + \omega_t = 1$$

The question here is how to find the best values of ω_f and ω_t. We used the Neural Networks [9] model for this purpose. Since neural networks produce continuous outputs, they may quite naturally be used for estimation and prediction. A Neural Network consists of a layered, feed forward, completely connected network of artificial neurons, or nodes. The neural network is composed of two or more layers, although most networks consist of three layers: an input layer, a hidden layer, and an output layer. There may be more than one hidden layer, although most networks contain only one, which is sufficient for most purposes. Fig. 4 shows a Neural Network with three layers. Each connection between nodes has a weight (e.g., W_{1A}) associated with it. At initialization, the weights are randomly assigned to values between 0 and 1.

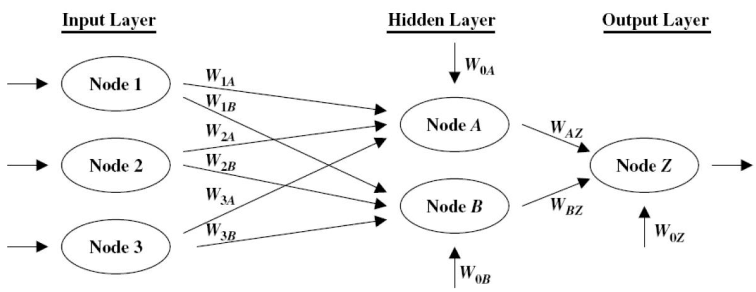

Fig. 4. Neural Networks Model

Neural Network works as follows. First, a combination function (usually summation, $\sum$) produces a linear combination of the node inputs and the connection weights into a single scalar value. Thus, for a given node j:

$$net_j = \sum_{i=0}^{I} W_{ij} x_{ij}$$

where x_{ij} represents the ith input to node j and W_{ij} represents the weight associated with this input and there are $I+1$ inputs to this node. Note that x_1, x_2, $\ldots$,x_I represent inputs from upstream nodes, while x_0 represents a constant input, analogous to the constant factor in regression models, which by convention uniquely takes the value $x_{0j} = 1$.

This combination function is then used as an input to an activation function. In biological neurons, signals are sent between neurons when the combination of inputs to a particular neuron cross a certain threshold, and the neuron fires. This is nonlinear behavior, since the firing response is not necessarily linearly related to the increment in input stimulation. Artificial neural networks model this behavior through a nonlinear activation function.

After creating the neural network with a data mining tool and training it with the test collection which is discussed in the next section, we found that in this test collection the best values for ω_f and ω_t are $\omega_f = 0.603$ and $\omega_t = 0.397$ (the estimated accuracy of this result was 99.29). The neural network that was constructed for our training set had two hidden layers with 13 and 8 neurons respectively.

VI. Experimental Evaluation

For evaluating the measures which are proposed in previous sections, we semi-automatically built a test collection of 241 OWL-S service descriptions which use a reference ontology with 494 concepts[2]. In order to have better results, most of the

[2]This test collection is publicly available at http://projects.semwebcentral.org/projects/sws-tc

Service 1	Service 2	Calculated Similarity	Really similar?	Recall	Precision
SSN Finder	Social Security Number Finder	0.98093	Yes	0.1	1.0
Get Book ISBN	ISBN Finder	0.93985	Yes	0.3	1.0
Address Finder	Address Book	0.78592	Yes	0.4	1.0
Who wrote this book	Book Author Finder	0.78278	Yes	0.5	1.0
Get Altitude	Altitude Finder	0.70785	Yes	0.6	1.0
City to Country	Get Country Capital	0.68613	No	0.6	$\max\{1.0, 0.83\} = 1.0$
Get Author By Publication	Who wrote this book	0.68522	Yes	0.7	0.86
Get Domain NameServer	Get Domain Whois	0.67012	No	0.7	$\max\{0.86, 0.75\} = 0.86$
Hotel Website	Personal Website Finder	0.65375	No	0.7	$\max\{0.86, 0.67\} = 0.86$
Email Address Validator	Email Validator	0.64162	Yes	0.8	0.70
Book Price	Get book price	0.63586	Yes	0.9	0.73
Strong password generator service	Password Generator	0.62618	Yes	1.0	0.75

TABLE I

A SAMPLE CALCULATION OF PRECISION AND RECALL VALUES

service descriptions are extracted from real word services that are publicly available on the web. Most of the structure of the taxonomy of the referenced ontology is also based on the hierarchy of the words in WordNet.

We then used *Recall* and *Precision* metrics for evaluating the results. *Recall* is the extent to which the method retrieves *all* of the similar services (i.e. avoiding false negatives) while *Precision* is the extent to which the method retrieves *only* the real similarities (i.e. avoiding false positives). For evaluating the proposed method, we found the similarity of each pair of services with the proposed method (i.e. a total of $241 \times 240 = 57840$ similarity values). Then we sorted the values in the descending order. Table I shows a sample of such ordering. In this table, 10 of the pairs are really similar and three of them are mistakenly detected as similar. In the first row of the table when we detect one of the 10 real similarities, the recall value would be 0.1 and the precision value is 1.0 (because no false similarity is reported up to that time). In the 7th row of the table, a false similarity is reported and because before that row we have seen 6 of the 10 real similarities the recall value would be 0.6. On the other hand, because 6 of the 7 reported similarities are real similarities the precision would be 0.83. But because we have another precision value for the 0.6 recall level the maximum of the precision values would be considered for this level. The other recall and precision values are calculated as well.

Fig. 5 shows the precision and recall values of the proposed similarity measure on our test collection for different values for ω_f and ω_t. As can be seen in this figure, using only one of the functionality or textual similarity measures has lower precision than using a combination of them.

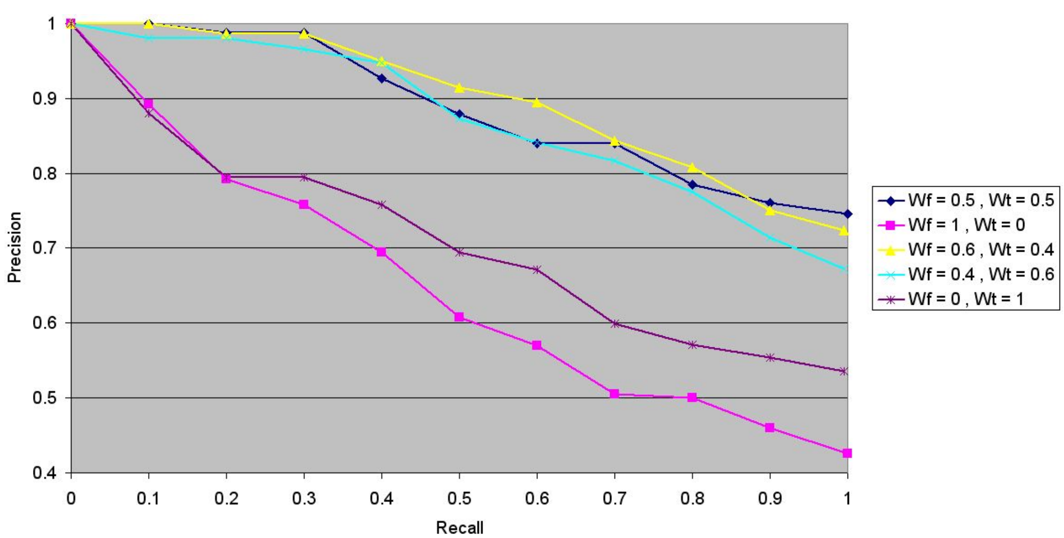

Fig. 5. Recall and Precision diagram of the proposed algorithm

VII. RELATED WORKS

In [10], authors proposed a semantic similarity measure for semantic web services. Their similarity measure is based on the principle that the more shared information is hold by two concepts, the more similar they are. By quantitizing the individual information two OWL concepts contain and also the shared information between them, the similarity between two concepts is measured. Consequently, the similarity of two service ontologies can be measured. The authors use OWL Lite, a restricted version of OWL, as a descriptive language and give examples of the applicability on OWL-S descriptions. In fact, this paper mainly focuses on introducing a concept similarity measure and it is not clear how this similarity measure can be exploited in calculating the similarity of web services and because of the lack of experimental results in that paper it is hard to evaluate this work.

In [11] a similarity-based approach is used for searching Web Services described in WSDL. Authors have built a web service search engine, Woogle, that supports searching for web service operations similar to a given one. The tool also supports searching for web service operations composable with a given one. The tool only uses information available in WSDL files, but clusters it, based on the names of the fields, in an effort to extract semantically meaningful concepts.

To support programmatic service discovery, the authors of [12] have developed a suite of methods that assess the similarity between two WSDL specifications based on the structure of their data types and operations and the semantics of their natural language descriptions and identifiers. Given only a textual description of the desired service, a semantic information-retrieval method can be used to identify and order the most relevant WSDL specifications based on the similarity of the element descriptions of the available specifications with the query. If a (potentially partial) specification of the desired service behavior is also available, this set of likely candidates can be further refined by a semantic structure-matching step, assessing the structural similarity of the desired vs. the retrieved services and the semantic similarity of their identifiers.

[13] presents a conceptual model which classifies properties of web services into four categories: Lexical similarity; Attribute similarity; Interface similarity; and QoS similarity. For each category, a similarity assessment method has been given. This work is also based on WSDL descriptions of web services and therefore can not precisely measure semantic similarity of services.

Semantic similarity of concepts and terms is also widely studied in many research areas. [14] calculates the probability of encountering an instance of a concept. Intuitively, as this probability increases the informativeness of the concept decreases. The similarity of two concepts is then defined as the informativeness of their most specific subsumer concept in the is-a hierarchy. Other similarity measures based on the information content of concepts are also proposed in [15], [16]. Many other papers such as [17], [18] propose methods that use the distance between concepts in the is-a hierarchy as a measure of their similarity.

VIII. Conclusions and Future Work

In this paper, we introduced a semantic similarity measure for determining the similarity of OWL-S annotated web services. The similarity measure is a linear combination of the functional similarity of services and their textual similarities. The functional similarity of services is determined by measuring semantic similarity which exists between their sets of inputs and outputs. In order to be able to measure the input/output similarity of services we introduced techniques for finding the similarity of OWL concepts which are used for annotating the inputs and outputs of services. The experimental results of evaluating the proposed similarity measure are promising.

We then showed how Neural Networks can be used for determining the importance of the two introduced similarity measures, functional similarity and textual similarity. The calculated importance of these two measures can be used for combining them.

Our future work is to use the proposed similarity measure for clustering web services, so that similar services be in the same clusters. Then service taxonomies can be built based on these clusters. A service taxonomy can facilitate the discovery process of web services. It can also be used in the recommendation of similar services to clients.

IX. Acknowledgements

The authors thank Hadi Saboohi for his efforts in preparation of the test collection.

References

[1] D. Martin and et al. OWL-S: Semantic markup for web services. [Online]. Available: http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/
[2] S. B. et al. OWL Web Ontology Language Reference, W3C recommendation 10 february 2004. [Online]. Available: http://www.w3.org/TR/2004/REC-owl-ref-20040210/
[3] G. Stumme and A. Maedche, "Ontology merging for federated ontologies on the semantic web," in *Proceedings of the International Workshop for Foundations of Models for Information Integration (FMII-2001)*, Sept. 2001.
[4] S. Melnik, H. Garcia-Molina, and E. Rahm, "Similarity flooding: A versatile graph matching algorithm and its applications to schema matching," in *Proceedings of the 18th International Conference on Data Engineering*. San Jose, CA: IEEE Computer Society, Feb. 2002.
[5] J. Zhong, H. Zhu, J. Li, and Y. Yu, "Conceptual graph matching for semantic search," in *The 2002 International Conference on Computational Science(ICCS2002)*, Amsterdam, Apr. 2002.
[6] A. Tversky, "Features of similarity," *Psycological Review*, vol. 84, no. 4, pp. 327–352, 1977.
[7] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison-Wesley, 1999.
[8] C. Fellbaum, *An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.
[9] D. T. Laros, *Discovering Knowledge in Data*. New Jersey, USA: John Wiley and Sons, 2005.
[10] J. Hau, W. Lee, and J. Darlington, "A semantic similarity measure for semantic web services," in *Workshop of WWW2005, Web Service Semantics: Towards Dynamic Business Integration*, Chiba, Japan, May 2005.

[11] X. Dong, A. Halevy, J. Madhavan, E. Nemes, and J. Zhang, “Similarity search for web services,” in *Proceedings of the 30th VLDB Conference*, Toronto, Canada, pp. 372–383.

[12] E. Stroulia and Y. Wang, “Structural and semantic matching for accessing web service similarity,” *International Journal of Cooperative Information Systems*, vol. 14, no. 4, pp. 407–437, 2005.

[13] J. Wu and Z. Wu, “Similarity-based web service matchmaking,” in *Proceedings of the 2005 IEEE International Conference on Services Computing (SCC05)*. Los Alamitos, CA, USA: IEEE Computer Society, 2005.

[14] P. Resnik, “Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language,” *Journal of Artificial Intelligence Research*, vol. 11, pp. 95–130, 1999.

[15] J. Jiang and D. Conrath, “Semantic similarity based on corpus statistics and lexical taxonomy,” in *Proceedings of the International Conference on Research in Computational Linguistic*, Taiwan, 1998.

[16] P. Lord, R. Stevens, A. Brass, and C. Goble, “Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation,” *Bioinformatics*, vol. 19, no. 10, pp. 1275–1283, 2003.

[17] R. Rada, H. Mili, E. Bicknell, and M. Blettner, “Development and application of a metric on semantic nets,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 1, pp. 17–30, 1989.

[18] Y. Li, Z. A. Bandar, and D. McLean, “An approach for measuring semantic similarity between words using multiple information sources,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 871–882, 2003.

[19] P. Raftopoulou and E. Petrakis, “Semantic similarity measures: A comparison study,” Technical University of Crete, Department of Electronic and Computer Engineering, Tech. Rep. TR-TUC-ISL-04-2005, Jan. 2005.

[20] D. Lin, “An information-theoretic definition of similarity,” in *Proceedings of International Conference on Machine Learning*, 1998.