

Evaluation of Web Service Similarity Assessment Methods

Yousub Hwang

Department of Business Administration, College
of Business & Economics, University of Seoul
(yousub@uos.ac.kr)

.....

The World Wide Web is transitioning from being a mere collection of documents that contain useful information toward providing a collection of services that perform useful tasks. The emerging Web service technology has been envisioned as the next technological wave and is expected to play an important role in this recent transformation of the Web. By providing interoperable interface standards for application-to-application communication, Web services can be combined with component based software development to promote application interaction and integration both within and across enterprises. To make Web services for service-oriented computing operational, it is important that Web service repositories not only be well-structured but also provide efficient tools for developers to find reusable Web service components that meet their needs. As the potential of Web services for service-oriented computing is being widely recognized, the demand for effective Web service discovery mechanisms is concomitantly growing. A number of techniques for Web service discovery have been proposed, but the discovery challenge has not been satisfactorily addressed. Unfortunately, most existing solutions are either too rudimentary to be useful or too domain dependent to be generalizable.

In this paper, we propose a Web service organizing framework that combines clustering techniques with string matching and leverages the semantics of the XML-based service specification in WSDL documents. We believe that this is one of the first attempts at applying data mining techniques in the Web service discovery domain. Our proposed approach has several appealing features : (1) It minimizes the requirement of prior knowledge from both service consumers and publishers; (2) It avoids exploiting domain dependent ontologies; and (3) It is able to visualize the semantic relationships among Web services. We have developed a prototype system based on the proposed framework using an unsupervised artificial neural network and empirically evaluated the proposed approach and tool using real Web service descriptions drawn from operational Web service registries. We report on some preliminary results demonstrating the efficacy of the proposed approach.

.....

Received : November 11, 2009 Accepted : November 30, 2009 Corresponding author : You-sub Hwang

1. Introduction

One of the industry standards for Web serv-

ice repositories is Universal Description, Discovery, and Integration (UDDI). The search mechanisms provided by most UDDI registries are

based on either keyword search or browsing through predefined hierarchical business categories. Keyword search is known to have the drawback that users may be confronted with many irrelevant search results due to such issues as synonyms and word ambiguity. Consequently, users may have to spend a lot of time browsing through the results to identify the Web services that most closely meet their needs.

If the Web services are categorized well, browsing through predefined categories may be a better alternative for service discovery. However, it requires both service providers and consumers to have prior knowledge of the service categorization schemes, such as the North American Industry Classification System (NAICS) (<http://www.census.gov/epcd/www/naics.html>) and the United Nations Standard Products and Services Code (UNSPSC)(<http://www.unspsc.org/>). In other words, the service providers must publish their Web services in the appropriate UDDI business categories, and the service consumers must browse the 'right' business categories to find the potentially relevant Web services. Thus, although UDDI provides a standard interface for simple keyword based search and predefined business category based browsing, more effective mechanisms are still needed to improve the current search mechanisms for discovering semantically relevant Web services.

The main objective of this research is to develop a more effective mechanism for Web

service organizing framework. We propose a Web service organizing framework that combines clustering techniques with string matching and leverages the semantics of the XML-based service specification in WSDL (Web Services Description Language) documents. We have developed a software tool for Web service organization based on the proposed approach using an unsupervised artificial neural network and empirically evaluated the proposed approach and tool using real Web service descriptions drawn from operational Web service registries. We report on some preliminary results demonstrating the efficacy of the proposed approach. The remainder of this paper is organized as follows. We begin in Section 2 with a literature review on Web service discovery and clustering methods. In Section 3, we discuss the clustering methods implemented in the prototype. We then report on our empirical evaluation in Section 4. Finally, we conclude the paper with a discussion of future research in Section 5.

2. Related Work

2.1 Web Service Discovery

A large number of solutions for service discovery have been proposed, however, no single methodology satisfactorily resolves the problems people face in searching for and retrieving semantically relevant services from a Web services repository. Previous research on Web serv-

* This work was supported by the Foundation of Industry-Academic Cooperation in the University of Seoul.

ice discovery techniques can be categorized into three broad areas: denotational semantic method, information retrieval, and descriptive method.

2.1.1 Denotational Semantic Method

The denotational semantic method considers a service to be a function that requires inputs and generates outputs. Each service can be described by signature and specification pair; a signature represents the structure of a component's input/output parameters, whereas a specification describes a component's dynamic behavior, such as a pre-/post condition. Purtilo and Altee propose a signature matching method for the interface adaptation of software components by specifying their model's parameters(Purtilo, J.M et al., 1991). Zaremski and Wing formally define a set of concepts related to signature- and specification matching for retrieving components from a software library(Zaremski, 1995; Zaremski et al., 1997).

The limitation of UDDI is its lack of an explicit representation of the capability of Web services. The result is that UDDI supports the location of essential information about a particular Web service, once the Web service is known to exist, but it is impossible to locate a Web service based only on what it does. Thus, there is a great need for standardized vocabulary to represent service specifications. The W3C Web Ontology Working Group defined the Web Ontology Language for Services (OWL-S), which is aimed at being a standardized and broadly accepted ontology language to represent service spe-

cifications(W3C Web Ontology Working Group). To leverage on OWL-S, Paolucci et al., propose a translation function from OWL-S profiles to UDDI records(Paolucci et al., 2002). They adopt OWL- S as the service description language, and then discuss a matching algorithm between advertisements and requests described in OWL-S that recognizes various degrees of matching (e.g., exact, plug-in, and subsume)(Paolucci et al., 2002).

Gonzalez-Castillo et al., focus on defining the matching process based on the Description Logics (DL) subsumption relationship between the advertisement and the request(Gonzalez-Castillo et al., 2002). The authors assume an extensive representation of the types of Web services to specify the type of service. Similar to the OWL- S Matchmaker, they define a number of degrees of matching (e.g., exact, sub-concepts of, and subsume).

Li and Horrocks assume an intensive representation of Web services capabilities that is equivalent to the OWL-S service profile(L et al., 2004). A matching process utilizes a modified version of the OWL-S profile to facilitate the subsumption process, and it assumes multiple degrees of matching. The only difference is the use of intersection as an additional degree of match.

Gao et al., propose a new lightweight capability description language (SCDL) to describe, advertise, request, and match Web services capabilities precisely(Gao et al., 2002). A Web service is defined by the following: name, ontologi-

cal description, type, input/output parameters, and pre- and post-conditions. SCDL defines the four types of atomic Web service capability matches as follows: exact match, plug-in match, relaxed match, and not relevant. The pre- and post-conditions need some elaboration.

Benatallah et al., developed an interesting matching algorithm that utilizes both signature matching and specification matching(Benatallah, 2005; Benatallah et al., 2003). Their hybrid algorithm has a distinct feature in that it tries to retrieve a service or a collection of services that provides as many of the outputs of a service request as possible and requires few inputs, which are not provided in the service request. However, while the extension with specification matching allows for a comparison between various behavioral descriptions of Web services, there is no guarantee that the behavioral specifications provided by service providers accurately reflect the components' capabilities. Consequently, the search results may contain too many semantically irrelevant Web services

Gannod and Bhatia propose a toolset that exploits a signature matching method as a means for facilitating Web service discovery(Gannod et al., 2004). In their toolset, a service consumer generates a signature for a Web service query request by specifying a structure of input/output parameters. The toolset then compares the signature of the request with the signatures of the services published in a repository. Since signature matching is based solely on structures (i.e., data type and number of input/output pa-

rameters), it is less helpful when searching for Web services on the basis of what they do (i.e., the semantics of the Web services).

Cardoso and Sheth propose a novel Web service discovery method where WSDL documents are annotated according to shared process ontologies(Gardoso et al., 2003). The annotation-based approach minimizes the expensive migration process from WSDL to OWL-S and allows for semantics- based Web service discovery. The authors introduce a similarity function to identify similar entity classes by using a matching process over synonym sets, semantic neighborhoods, and distinguishing features that are classified according to parts, functions, and attributes. However, the shared process ontologies can be used effectively only in limited domains, and they cannot scale up to the whole Web.

2.1.2 Information Retrieval Method

Traditional information retrieval methods represent each document (and user query) written in a natural language as a set of keywords called "index terms" and use the index terms to compute the degree of similarity between a document and a user query. Hence, the most important of the tools for information retrieval is the index-a collection of terms with pointers to places where information about documents can be found(Dong et al., 2004). In an effort to increase the precision of service discovery without involving an additional level of semantic markup, several approaches based on machine-learning techniques are proposed (e.g., (Sabou, 2007;

Manber, 1999; Kokash et al., 2006). All of them report enhancements in precision of automated service matchmaking.

Wang and Stroulia propose a Web service discovery method that combines information retrieval techniques with a WSDL structure matching algorithm(Wang, 2003; Wang, 2003; Stroulia et al., 2005). To measure similarities between Web services, the WordNet lexicon(WordNet) was employed. According to the experimental results, the methods are neither precise nor robust. The main drawback, in our opinion, is that the methods use poor, unnormalized heuristics (e.g., matching scores of 5 or 10) in assigning weights for term similarity. The actual contents of WSDL files tend to be highly varied given the use of synonyms, hyponyms, and different naming rules. They might not even be composed of proper English words (i.e., abbreviations). Therefore, applying lexical references, such as WordNet, is not feasible. Furthermore, WordNet tends to generate an excessive number of synonyms, and thus, there were many false correlations that might affect the relatively low precision rate. The authors ignore the standard stemming process, which improves recall by reducing all forms of a term to single stemmed form. This may explain why the authors obtained relatively low recall rates in their experiments.

In(Wu et al., 2005), Web service similarity is defined as combining a WordNet-based lexical similarity and structural similarity. The authors aim at grounding the service matchmaking process on a lightweight semantic comparison of

signature specifications in a WSDL file. The authors add Quality of Service (QoS) to the properties of the Web service conceptual model. To measure QoS, the authors introduce four concrete parameters, such as time to process, time to delay, time to repair, and time to failure. In order to measure service similarities between Web services, the authors apply WordNet-based lexical reference. Thus, the authors' approach has the same flaw as that of Wang and Stroulia.

The vector space model (VSM), recognized as one of the most popular information retrieval techniques, was proposed by Salton et al(Salton et al., 1975). In the VSM of information retrieval, queries and documents are represented as vectors in a high dimensional space where each dimension represents a word. Platzer and Dustdar implemented a VSM-based search engine for Web services(56 Platzer et al., 2005). The authors extracted keywords from a proportion of WSDL files, such as Endpoint URL, Message, and textual descriptions. The search engine has the capability of handling natural language based keyword and can return a list of search results with similarity rating scores. The authors do not clearly address why other contents in WSDL files (e.g., operation names and parameter information) are not considered as input features. According to other research studies[45, 50, 53], both the operation and types elements in WSDL files are important sources for inferring the capabilities of Web services. Therefore, the authors may need to expand their keyword extraction policy in WSDL files.

Recently, machine-learning techniques have been applied to Web service matching and classification at either the whole Web service level (Heß, 2004; Fan et al., 2005), or at the operation level (Dong et al., 2004). One interesting solution when using machine-learning techniques is the Woogle engine (Dong et al., 2004). The Dong et al. conduct cluster analysis with parameter names appearing in the WSDL files by using a hierarchical clustering method and compute the similarities between the operations and input/output parameters. These computed similarity scores are used for service similarity assessment. The main intuition behind this approach is that when all the parameters inside a set frequently occur simultaneously, they may correspond to similar functionalities. The Woogle engine provides a simple keyword search and template search. The template search goes beyond keywords by allowing a keyword search that is specifically limited in names of input and output parameters. In addition, the Woogle engine allows users to combine searching and browsing at the same time to improve the precision of the search results. Our work differs in that we are employing an artificial neural network-based approach, rather than a hierarchical clustering method. In addition, Dong et al. ignore the enumerated data values of parameters as an input feature for their service similarity assessment.

Fan and Kambhampati (Fan et al., 1992) provide a survey of publicly available Web services in real-world public service repositories by utilizing cluster analysis. In (Fan, 1992), the hier-

archical clustering method was used for cluster analysis.

The authors compute the similarities between services by applying VSM-based term frequency analysis based on the textual description and documentation fields of the WSDL files. After Fan and Kambhampati gathered WSDL files from the public Web service registries, they removed the duplicates by using a combination of service name and provider name as the key and checking the duplicates based on the keys.

2.1.3 Descriptive Method

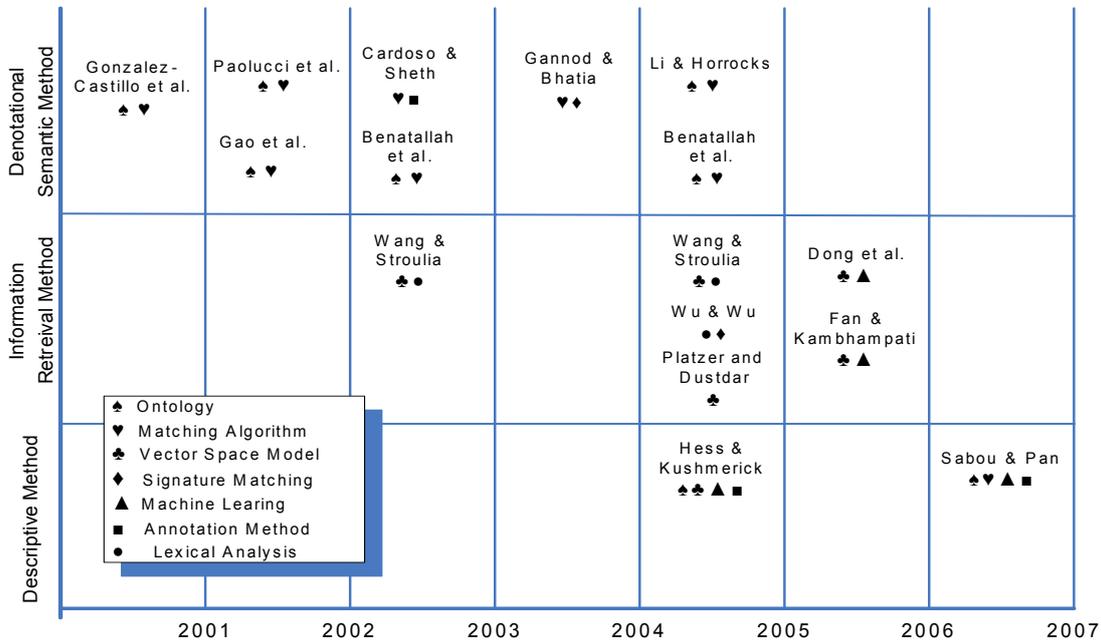
Although reported to be very effective in retrieving software components for reuse, the descriptive methods are labor-intensive for constructing predefined taxonomies (Hendler et al., 1992). To address such an issue, Heß and Kushmerick apply machine-learning techniques to generate predefined taxonomies for Web services (Heß, 2004; Heß et al., 2004). The authors aim at providing a semi-automated approach that makes use of the supervised classification and hierarchical clustering methods to suggest OWL-S-based Web service taxonomies. Our work differs in that we are performing an unsupervised artificial neural network-based service similarity assessment, rather than a supervised classification. The authors assume three levels of OWL-S-based taxonomies: a general taxonomy representing the nature of the services, a domain taxonomy representing a collection of functionalities, and a data-type taxonomy representing a collection of semantic data categories. Because they rely on the domain-spe-

cific ontologies for Web service taxonomies, the authors' method has the same flaw as the denotational semantic methods do: the cost of managing multiple domain-dependent ontologies is too high.

Sabou and Pan propose an ontology learning-based approach for (semi-)automatic concept identification, which will extend the predefined Web service taxonomy, thus ensuring that it reflects the contents of the underlying Web services (Sabou et al., 2007). The newly extracted concepts can be utilized as a set of pre-enumerated keywords for service discovery. The authors observed that most of the noun phrases in the textual descriptions denoted the service parameters while the verbs indicated the functionality of the service. The ultimate objective of

Sabou and Pan's research is generating domain-specific ontologies in a bottom-up fashion (learned from available sources) rather than being imposed in a top-down fashion (requiring costly manual generation of ontologies).

We have presented a summary of the literature on service discoveries classified according to three major methods focused on facilitating service discovery. The service discovery proposals we reviewed are plotted on a timeline (see <Figure 1>). Our method attempts to facilitate service discovery and publishing by leveraging the advantages of all service discovery strategies discussed in this chapter: the denotational semantic methods, the information retrieval methods, and the descriptive method.



<Figure 1> Summary of Web Service Discovery Research

2.2 Clustering Methods

Clustering has been widely studied in several disciplines since the early 70's (Hartigan, 1975; Jain et al., 1999). Clustering is a technique for distinguishing a large number of instances of a given problem domain into a finite number of groups based on a similarity or dissimilarity. Similarity is often derived from the inner product between vector representations, a popular way to quantify document similarity. There are three types of clustering algorithms: non-hierarchical, hierarchical, and artificial neural network-based methods.

2.2.1 Non Hierarchical Clustering Methods

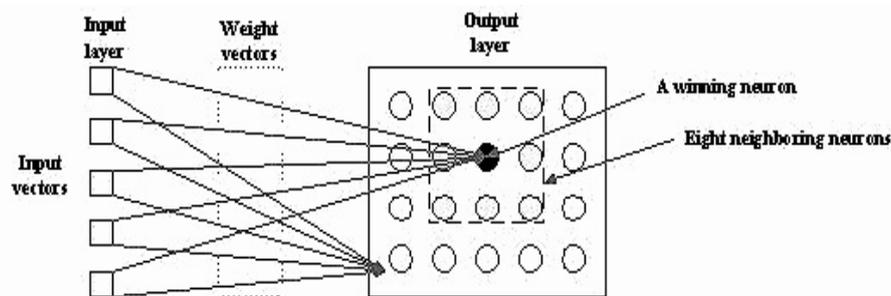
The most popular non-hierarchical clustering methods are k -means and k -medoids. These methods require the user to specify initial seeds of clustering, k , prior to a cluster analysis. All the n objects are then compared with each seed by means of the Euclidean distance and assigned to the closest cluster seed. The procedure is then repeated over and over again. In each stage the seed of each cluster is recalculated by using the average vector of the objects assigned to the cluster. The algorithm stops when the changes in the cluster seeds from one stage to the next are close to zero or smaller than a pre-specified value. Every object is assigned to only one cluster.

When means are not meaningful (e.g., certain binary features), the k -medoids algorithm might be the method of choice. It uses medoids

(representative sample objects) instead of means. However, the extension of median to multivariate data is usually realized using a randomized approach. A random object is selected and the distance function is evaluated assuming that the selected object is one of the k -medoids. If the distance decreases, a swap is performed and the search is iterated until no changes are needed for all medoids. However, these methods are applicable only when mean or medoid is defined; if the clustering criteria contain categorical variables, dummy variables need to be generated and the performance of these methods may degenerate. Furthermore, the accuracy of these methods is very dependent upon the choice of initial seeds (Milligan et al., 1980).

2.2.2 Hierarchical Clustering Methods

Hierarchical clustering methods use distance matrices as clustering criteria. These methods can be agglomerative or divisive. Agglomerative methods proceed by a series of successive fusions of the individuals into groups, whereas divisive methods separate the individuals successively into finer groupings (Everitt et al., 2001). Both *complete linkage* and *single linkage* are agglomerative methods. They basically work in the following way: in the first stage each of the n objects to be clustered is considered as a unique cluster. The objects are then compared among themselves by using a measure of distance such as Euclidean, for example. The two clusters with smaller distance are joined. The



<Figure 2> Architecture of SOM Neural Network from(Kohonen, 2001; Gower et al., 1998)

same procedure is repeated over and over again until the desirable number of clusters is achieved. Only two clusters can be joined in each stage and they cannot be separated after they are joined.

The difference between *single linkage* and *complete linkage* is the definition of cluster distance. *Single linkage*, also called nearest neighbor, defines the distance between two clusters as the minimum distance between their individuals. *Complete linkage*, also known as furthest neighbor, defines the distance between two clusters as the maximum distance between their individual.

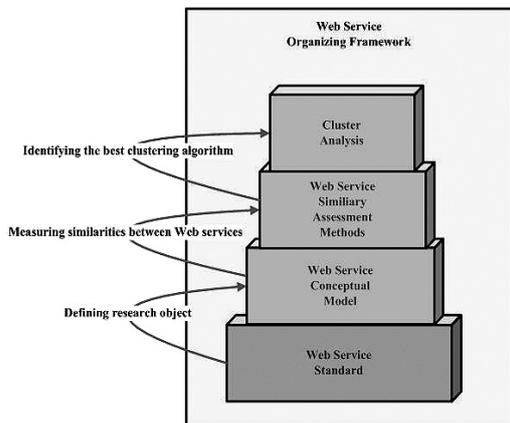
2.2.3 Artificial neural network-based methods

In artificial neural networks [88], the input passes through a connected network of simple processing units, called neurons, to the output. One of the most popular networks for clustering is topologically organized networks. The SOM fits a predefined network structure to the data in a topology preserving fashion by updating the winner and its adjacent neighbors. The prede-

finied network is inspired by the structure and function of the human brain, which is composed of millions of biological neurons working together. Similarly, an artificial neural network consists of a large number of artificial neurons. These neurons are simple and highly interconnected processing units operating in a parallel manner.

The architecture of the SOM is shown in <Figure 2>. An SOM network is typically a two-layer neural network consisting of an input layer and an output layer. The input layer is composed of a set of n -dimensional input vectors $x = [x_1, x_2, \dots, x_n]$, where n indicates the number of features that each input vector contains. The output layer is an m -dimensional (usually two-dimensional) map consisting of a set of neurons, each associated with an n -dimensional weight vector $w = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ (with the same dimension as that of the input vector). The weight vector expresses the relative importance of each input to a neuron in the grid. The layout of the neurons can be rectangular or hexagonal.

SOM takes a set of input vectors and maps them onto the neurons of a two-dimensional map.



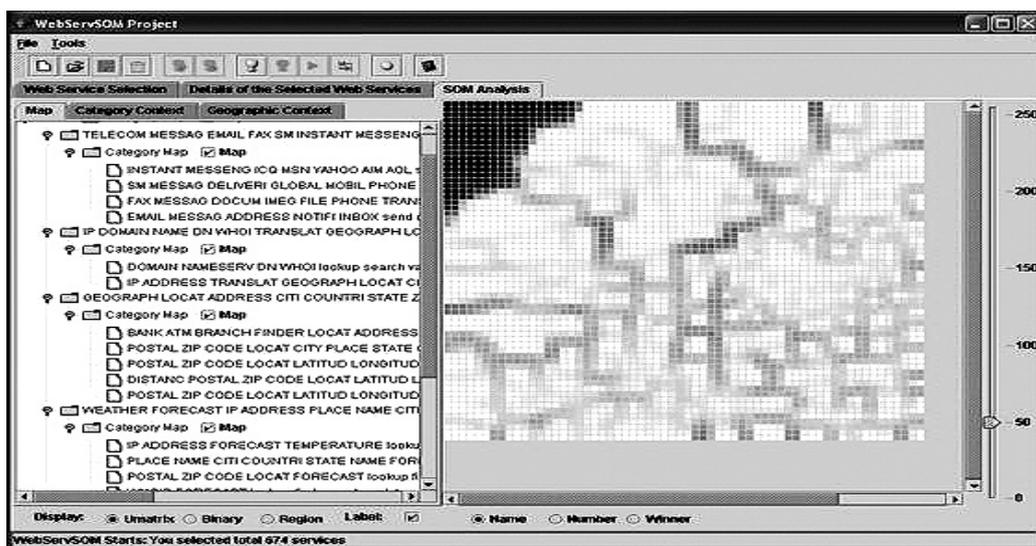
<Figure 3> Overview of Web Service Organizing Framework

Since SOM is an unsupervised learning algorithm, there is no target output available for the input. Hence, the SOM network learns only from its input through repetitive adjustments of the weights of the neurons. The weight vectors are randomly initialized at the first stage.

3. Web Service Organizing Framework

A prerequisite for Web service discovery is the ability to organize Web services into semantically related groups. We propose a comprehensive framework for organizing Web services into semantically related groups relying on existing Web service standards such as WSDL. Three critical issues must be addressed to accomplish this : (1) defining the research object based on the existing Web service standard, (2) measuring similarities between Web services, (3) identifying the best clustering algorithm in the context of Web services. By investigating these three critical issues sequentially, we develop the Web service organizing framework based on the results or findings from each critical issue (See <Figure 3>).

First, a formal Web service conceptual mo-



<Figure 4> Clustering Result from Our Prototype System

del is developed to infer the semantics of Web services based on the existing Web service standard without requiring an additional level of semantic markup that involves the time- and cost-intensive migration process. We formally define the Web service conceptual model based on the WSDL specification standard. Although WSDL files do not explicitly provide support for semantic specifications, they do contain information that can be used to infer the capabilities of the specified Web service. By extensively observing the contents of the WSDL files, we are able to identify the two categories of semantics that can be inferred from the WSDL file : (1) domains of activity (what a service deals with) and (2) types of functionality (what a service does). An extensive literature review reveals that various conceptual models have been defined in Web service discovery during the last decade. However, no attempts have been made to utilize the full contents of the WSDL specification standard. This framework is then used as a basis for measuring similarities between Web services as shown in <Figure 3>.

Having defined the Web service conceptual model, we developed a set of service similarity assessment methods: a term frequency analysis-based method and a string pattern comparison-based method. Final component of the Web service organizing framework is cluster analysis. Based on the Web service similarity, artificial neural network based clustering method is implemented for organizing Web services into semantically related groups (See <Figure 4>).

4. Experiments

Because the accuracy of the final partition depends upon the clustering algorithm used to organize the objects, several studies have been conducted to evaluate the performance of the clustering algorithms (Milligan, 1980; Gower et al., 1967). Zhao and Karypis provide a recent survey on clustering evaluation in (Zhao et al., 2004). Milligan and Cooper compared non-hierarchical clustering algorithm (k -means) and fourteen hierarchical clustering algorithms by simulating with error perturbation (i.e., inclusion of outliers). The complete linkage algorithm was very affected by the inclusion of outliers but the k -means and single linkage algorithm were very robust against this type of error. Mangiameli et al. compared the SOM artificial network with seven hierarchical clustering algorithms and found that SOM is superior to all of the hierarchical clustering methods (Mangiameli et al., 1996). However, Balakrishnan et al., experimentally compared the SOM neural network to the k -means and found that SOM did not have a good performance (Balakrishnan et al., 1994). Since there is no silver-bullet clustering algorithm for every problem, we need to identify which clustering algorithm is suited for organizing Web services. In order to identify the best clustering method for Web service organization, we have conducted a set of experiments and report on the results in this section.

4.1 Experiment Design

We now describe a set of experiments that

evaluate the quality of results delivered by five clustering algorithms (i.e., k -means and k -medoids, single linkage, complete linkage, and SOM neural network). Eisen et al. implemented an integrated clustering tool, which supports both non-hierarchical and hierarchical cluster analysis (Eisen et al., 1998). In our experiments, we utilized both Eisen et al.'s clustering tool and our SOM neural network tool.

We extracted some sample data from several public Web service repositories, including *XMethod.com* and *StrikeIron.com*. In order to collect an operational Web service, we checked whether the WSDL document for the service was available. If the URL of a service was a broken link, we removed the service from our sample data set. Finally, we obtained a total of 310 operational services.

Among the collected operational services, we identified thirty three semantically related services from six different categories: currency rate converter (three services), weather information finder (five services), DNA information searcher (five services), and SMS message sender (five services). Therefore, 11% of the collected services are semantically associated with others.

We conducted experiments to compare the performance of the five clustering methods. We ran each clustering method 100 times; each time, input data were randomly re-sampled from the collected operational services. Each input data set consisted of 40 services; 20 services from the non-semantically related portion and the other 20 from the semantically related portion. We

then used the analysis of variance (ANOVA) and the Tukey post hoc test to test whether the performance measures of different clustering analysis methods were significantly different.

To cluster Web services, we created and used various kinds of features, including: (1) degree of similarity between service names, (2) degree of similarity between a pair of operation names, and (3) degree of similarity between a pair of input/output parameter names. We are in the process of implementing an information retrieval based method for comparing service descriptions and will include it in subsequent experiments.

The current version of the Web Services Description Language for Java Toolkit (WSDL4J) does not support querying the `<wsdl : types>` element, which contains detailed input/output parameter information, such as parameter name, data type, minimum/maximum occurrence, and so on. Thus, we implemented the WSDL parser in our prototype system by utilizing generic XML parser. The WSDL parser extracts operation names and input/output parameter names.

Our prototype system computes the string similarity based on the q -grams method (Marchionini et al., 1988). The q -grams method, an approximate pattern matching technique, is one of the most frequently used methods for measuring the degree of similarity between a pair of strings. In the string similarity calculation process, a set of stop words (e.g., 'GET', 'SET', 'PUT', 'POST', etc.) is removed from the original string to improve the accuracy of the similarity score.

<Table 1> Contingency table model for evaluation

System Answer	Human Expert Answer	
	Yes	No
Yes	<i>A</i>	<i>B</i>
No	<i>C</i>	<i>D</i>

4.2 Evaluation Measures

To evaluate the performance of each clustering method, we adapt some ideas from the contingency table model (Marchionini et al., 1987). The contingency table model provides a systematic way of comparing the system generated clustering outputs with a model solution for the problem designed by human experts. The contingency table model requires explicit construction of a model from human experts and places the burden of comparison between the model and the system's output on the system instead of the judge (Manber et al., 1997). To score our results, both the system generated clustering output and the human expert generated clustering output were converted into two lists of yes-no answers to the co-occurrence question, "Does

the pair of services belong to the same cluster?" for each pair of services. We then built a contingency table (as in Table 1) by counting the number of observed associations in the co-occurrence answer lists.

Four human experts were recruited to generate a Web service category model for each input data set. The recruited human experts were business students (two graduates and two undergraduates) who are majoring in Management Information Systems (MIS). All human experts also had prior application development experience and understanding of Web service technology. They were asked to categorize each input data set into small groups and told that categories should not overlap, but they could select any number of categories. Each human expert referred to service name, service provider information, service descriptive documentation, and WSDL documents. In our experiments, the category model for each input data set generated by human experts was considered the decision base to evaluate the clustering results generated

<Table 2> ANOVA results of three measurements between different clustering methods.

Measurements		SS	df	MSS	F-ratio	p-value (p)
Accuracy	Between groups	0.420	4	0.1051	590.140	0.000
	Within groups	0.088	495	0.000		
False Positive	Between groups	13.650	4	3.412	653.340	0.000
	Within groups	2.585	495	0.005		
False Negative	Between groups	1.210	4	0.302	71.797	0.000
	Within groups	2.867	495	.0042		

<Table 3> Tukey post hoc tests

	Clustering Algorithm	N	Homogenous subset for alpha = .05			
			1	2	3	4
Accuracy	SOM	100	0.977			
	k-means	100	0.975			
	k-medoids	100		0.964		
	Single linkage	100			0.951	
	Complete linkage	100				0.893

<Table 4> Tukey's Post Hoc Test Result with False Positive Rate

	Clustering Algorithm	N	Homogenous subset for alpha = .05			
			1	2	3	4
False Positive (<i>fp</i>)	SOM	100	0.044			
	k-means	100		0.091		
	Single linkage	100		0.099		
	k-medoids	100			0.273	
	Complete linkage	100				0.475

using different experimental settings.

Based on <Table 1>, the following performance measures, which have been frequently used as performance measures in the literature, are derived:

$$\text{Accuracy} = \frac{A+D}{A+B+C+D}$$

$$\text{False Positive} = \frac{B}{A+B}$$

$$\text{False Negative} = \frac{C}{A+C}$$

4.3 Experiment Results

<Table 2> summarizes the ANOVA re-

sult, showing significant differences in all three measures across clustering methods. For example, the F-ratio of accuracy in the ANOVA test with five different clustering algorithms is 590.14 and the p-value is 0.000. This means that the difference between the accuracy of the five different clustering algorithms is significant at the level of 0.000. Since there is a significant difference among the different clustering algorithms, we conducted Tukey's post hoc test to determine exactly which means are significantly different from which other ones.

<Table 3> list Tukey's post hoc analysis results based on the average accuracy rates of five different clustering algorithms. In terms of accuracy, the Tukey's post hoc test grouped the

<Table 5> Tukey's Post Hoc Test Result with False Negative Rate

	Clustering Algorithm	N	Homogenous subset for alpha = .05				
			1	2	3	4	5
False Negative (<i>fn</i>)	<i>k</i> -means	100	0.077				
	Complete linkage	100		0.110			
	<i>k</i> -medoids	100			0.145		
	SOM	100				0.180	
	Single linkage	100					0.219

five clustering algorithms into four homogeneous subsets in the following order : SOM, *k*-means > *k*-medoids > Single linkage > Complete linkage. By analyzing details of the Tukey's post hoc test, we found the following:

- SOM and *k*-means outperform any other clustering algorithms in terms of accuracy.
- Non-hierarchical clustering algorithms outperform hierarchical clustering algorithms in terms of accuracy

<Table 4> lists Tukey's post hoc test results based on the average false positive rates of the five different clustering algorithms. From the Tukey's post hoc analysis results, we identify the followings:

- SOM outperforms any other clustering algorithms in terms of purity of clustering results within clusters.
- *k*-means outperforms *k*-medoids algorithm in terms of purity of clustering results within clusters.

Single linkage outperforms complete linkage in terms of purity of clustering results within clusters

<Table 5> summarizes Tukey's post hoc test results based on the average false negative rates of the five different clustering algorithms. The Tukey's post hoc analysis indicates that *k*-means outperforms any other clustering algorithms in terms of purity of clustering results across clusters. Although *k*-means had good accuracy and false negative rate, it requires that the number of clusters be specified a priori. In many fields, this is problematic because cluster analyses are often exploratory.

5. Conclusion

In this paper, we have presented a novel clustering based Web service organizing framework, which can be used in conjunction with the existing Web service technology, such as WSDL, to support a more automated service discovery process. We have also reported on some preliminary evaluation results. In terms of both accuracy and false positive rate, SOM outperformed

any other clustering algorithms. While k-means resulted in a lower false negative rate (purity of clustering results across clusters) in our experiments, SOM has the additional advantage of visualizing the inter-relationship among clusters by locating one cluster to other clusters (i.e., adjacent clusters are more similar than nonadjacent clusters). In the contingency table model, a member of each cluster is compared. Thus, the visualization effect of the SOM neural network has been ignored in the performance evaluation. We believe that the visualization of the SOM neural network can cancel out the relatively poor false negative rate and therefore SOM neural network is a promising tool for organizing Web services into semantically related groups.

Furthermore, the SOM neural network has been studied for the purpose of browsing. A two-dimensional SOM has been applied to produce a map of Usenet posting in the WEBSOM project(Honkela et al., 1997). The emphasis in WEBSOM is not to maximize cluster quality but to produce a humanly interpretable two-dimensional spatial map of known categories (e.g., newsgroups). Based on the findings from our experiments, we conclude that SOM is a promising clustering algorithm for organizing Web services.

Our work opens up several avenues for future research. While we have used only the q-grams string matching method for creating input features for cluster analysis, it will be interesting to explore other string/document matching methods (e.g., vector-space model). We have used a

single clustering method for each experiment. Future research may evaluate the effects of combining multiple clustering methods. SOM uses fixed network architecture in terms of the number and arrangement of neurons, which has to be defined prior to training. Obviously, in case of largely unknown input data characteristics, it is far from trivial for users to determine the network architecture that provides adequate clustering results. Hierarchical SOM, which allows users to gradually zoom in and out, may be more useful than a simple SOM. Finally, our proposed clustering base approach can be integrated with keyword search and predefined categorization based browsing, so that users can combine multiple search strategies in a flexible manner.

References

- <http://www.census.gov/epcd/www/naics.html>.
- <http://www.unspsc.org/>.
- Sabou, M. and J. Pan, "Towards Improving Web Service Repositories through Semantic Web Techniques", *Web Semantics : Science, Services and Agents on the World Wide Web*, Vol.5, No.2(2007), 142~152.
- Marchionini, G. and B. Shneiderman, "Finding facts vs. browsing knowledge in hypertext systems", *IEEE Computer*, Vol.21, No.3, (1988), 70~79.
- Marchionini, G., "An invitation to browse : Designing full text systems for novice users", *Canadian Journal of Information Science*, Vol.12, No.3(1987), 69~79.
- Manber, U., M. Smith, and B. Gopal, "WebGlimpse

- Combining Browsing and Searching”, in *Proceedings of the USENIX 1997 Annual Technical Conference*, Anaheim, CA. 1997.
- Wu, J. and Z. Wu. “Similarity-based Web Service Matchmaking”, in *Proceedings of the 2005 IEEE International Conference on Services Computing (SCC’05)*, 2005.
- Purtilo, J. M. and J. M. Atlee, “Module reuse by interface adaptation”, *Software Practice and Experience*, Vol.21, No.6(1991), 539~556.
- Zaremski, A. M. and W. J. M., “Signature Matching: a Tool for Using Software Libraries”, *ACM Transactions on Software Engineering and Methodology*, Vol.4, No.2(1995), 146~170.
- Zaremski, A. M. and J. M. Wing, “Specification matching of software components”, *ACM Transactions on Software Engineering and Methodology*, Vol.6, No.4(1997), 333~369.
- W3C Web Ontology Working Group, OWL Web Ontology Language Overview, <http://www.w3.org/TR/owl-features/>.
- Paolucci, M., T. Kawamura, T. Payne, and K. Sycara, “Importing the Semantic Web in UDDI”, in *Proceedings of the First International Semantic Web Conference (ICWC 2002)*, 2002.
- Paolucci, M., T. Kawamura, T. Payne, and K. Sycara, “Semantic Matching of Web Services Capabilities”, in *Proceedings of the First International Semantic Web Conference (ISWC2002)*, 2002.
- Gonzalez-Castillo, J., D. Trastour, and C. Bartolini. “Description Logics for Matchmaking of Services”, in *Proceedings of the Workshop on Applications of Description Logics*, Vienna, Austria, 2001.
- Li, L. and I. Horrocks, “A Software Framework for Matchmaking Based on Semantic Web Technology”, *International Journal of Electronic Commerce*, Vol.8, No.4(2004), 331~339.
- Gao, X., J. Yang, and M. P. Papazoglou, “The Capability Matching of Web Services”, in *Proceedings of IEEE Fourth International Symposium on Multimedia Software Engineering (MSE’02)*, 2002.
- Benatallah, B., M. Hacid, A. Leger, C. Rey, and F. Toumani, “On Automating Web Services Discovery”, *Journal on Very Large Data Bases*, Vol.14, No.1(2005), 84~96.
- Benatallah, B., M.-S. Hacid, and C. Rey, “Semantic reasoning for Web Services discovery”, in *Proceedings of the Workshop on E-Service and the Semantic Web*, 2003.
- Gannod, G. C. and S. Bhatia, “Facilitating Automated Search for Web Services”, in *Proceedings of the IEEE International Conference on Web Services*, San Diego, California, USA, 2004.
- Cardoso, J. and A. Sheth, “Semantic E-Workflow Composition”, *Journal of Intelligent Information Systems*, Vol.21, No.3(2003), 191~225.
- Manber, U., Foreword. In *Modern Information Retrieval*, ed. R. Baeza-Yates and B. Ribeiro-Neto. Reading, M A: Addison-Wesley, 5~8, 1999.
- Dong, X., J. Madhava, and A. Halevy, “Similarity Search for Web Services”, in *Proceedings of VLDB Conference*, Toronto, Canada, 2004.
- Dong, X., J. Madhavan, and A. Halevy, “Mining Structures for Semantics”, *ACM SIGKDD Explorations Newsletter*, Vol.6, No.2(2004), 53~60.
- Heß, A., E. Jonston, and N. Kushmerick, “Semi-Automatically Annotating Semantic Web Services (Extended Abstract)”, in *Proceedings of*

- Semantic Web Conference*, 2004.
- Heß, A. and N. Kushmerick, "Machine Learning for Annotating Semantic Web Services", in *Proceedings of AAAI Spring Symposium on Semantic Web Services*, 2004.
- Bruno, M., G. Canfora, M. D. Penta, and R. Scogamiglio, "An Approach to support Web Service Classification and Annotation", in *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service(EEE'05)*, 2005.
- Kokash, N., W.-J. Heuvel, and V. D'Andrea, "Leveraging and Web Services Discovery with Customizable Hybrid Matching", in Technical Report DIT-06~042, University of Trento, 2006.
- Wang, Y. and E. Stroulia, "Flexible Interface Matching for Web-Service Discovery", in *Proceedings of the Fourth International Conference on Service Oriented Computing (WISE '03)*, Rome, Italy : IEEE Computer Society Press, 2003.
- Wang, Y. and E. Stroulia, "Semantic Structure Matching for Accessing Web-Service Similarity", in *Proceedings of the First International Conference on Service Oriented Computing*, Trento, Italy, 2003.
- Stroulia, E. and Y. Wang, "Structural and Semantic Matching for Assessing Web-service Similarity", *International Journal of Cooperative Information Systems*, Vol.14, No.4 (2005) 407~436.
- WordNet. <http://wordnet.princeton.edu/>.
- Salton, G., A. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing", *Communications of the ACM*, Vol.18, No.11 (1975), 613~620.
- 56 Platzter, C. and S. Dustdar, "A Vector Space Search Engine for Web Services", in *Proceedings of the Third European Conference on Web Services(ECOWS'05)*, 2005.
- Fan, J. and S. Kambhampati, "A Snapshot of Public Web Services", *SIGMOD Record*, Vol.34, No.1(2005), 24~32.
- Hendler, J., R. P. Diaz, and C. Braun, "Computing Similarity in a Reuse Library System: An AI-Based Approach", *ACM Transactions on Software Engineering and Methodology*, Vol.1, No.3(1992), 205~228.
- Kohonen, T., *Self Organizing Maps*, Third ed. Berlin : Springer, 2001.
- Everitt, B. S., S. Landau, and M. Leese, *Cluster Analysis*, Arnold, A member of the Hodder Headline Group, 2001.
- Hartigan, J. A., *Clustering Algorithms*, New York: Wiley, 1975.
- Jain, A. K., M. N. Murty, and P. J. Flynn, "Data clustering: a review", *ACM Computing Surveys*, Vol.31, No.3(1999), 264~323.
- Milligan, G. W. and M. C. Cooper, "An examination of the effect of six types of error perturbation on fifteen clustering algorithms", *Psychometrika*, Vol.45, No.3(1980), 159~179.
- Deboeck, G. and T. Kohonen, *Visual Explorations in Finance with Self-Organizing Maps*, Berlin, New York : Springer, 1998.
- Gower, J. C., "A comparison of some methods of cluster analysis", *Biometrics*, Vol.23, No.4 (1967), 623~638.
- Zhao, Y. and G. Karypis, "Criterion functions for document clustering: Experiments and Analysis", *Machine Learning*, Vol.33, pp. 322~331, 2004.
- Mangiameli, P., S. K. Chen, and D. West, "A

- comparison of SOM neural network and hierarchical clustering methods”, *European Journal of Operational Research*, Vol.93, No.2(1996), 402~417.
- Balakrishnan, P. V., M. C. Cooper, V. S. Jacob, and P. A. Lewis, “A study of the classification of neural networks using unsupervised learning : A comparison with K-means clustering”, *Psychometrika*, Vol.59, No.4(1994), 509~525.
- Eisen, M. B., S. Landau, and M. Leese, “Cluster analysis and display of genome-wide expression patterns”, in *Proceedings of the National Academy of Science of the United States of America*, Vol.95, No.25(1998), 14863~14868.
- Honkela, T., S. Kaski, K. Lagus, and T. Kohonen, “WEBSOM-self-organizing maps of document collections”, in *Proceedings of WSOM'97, Workshop on Self-Organizing Maps*, Espoo, Finland, 1997.
- Affi, A. A. and V. Clark, *Computer-Aided Multivariate Analysis*, 3rd ed. Chapman and Hall, 1996.

Abstract

웹서비스 유사성 평가 방법들의 실험적 평가

항유섭*

월드와이드웹(WWW)은 유용한 정보를 포함하는 자료들의 집합에서 유용한 작업을 수행할 수 있는 서비스들의 집합으로 변화하고 있다. 새롭게 등장하고 있는 웹서비스 기술은 향후 웹의 기술적 변화를 추구하고 최근의 웹의 변화에 중요한 역할을 수행할 것으로 기대된다. 웹서비스는 어플리케이션 간의 통신을 위한 호환성 표준을 제시하며 기업 내/외를 아우를 수 있는 어플리케이션 상호작용 및 통합을 촉진한다. 웹서비스를 서비스 중심 컴퓨팅환경으로서 운용하기 위해서는 웹서비스 저장소는 조직화되어 있어야 할 뿐 아니라, 사용자들의 요구에 맞는 웹서비스 컴포넌트를 찾을 수 있는 효율적인 도구들을 제공하여야 한다. 서비스 중심 컴퓨팅을 위한 웹서비스의 중요성이 증대됨에 따라 웹서비스 발견을 효율적으로 제공할 수 있는 기법의 수요 또한 증대된다. 웹서비스 발견을 위한 많은 기법들이 제안되어 왔지만, 대부분의 선행연구들은 활용하기에는 제대로 발달하지 못하였거나 특정 도메인에 너무 치중하여 일반화하기 어려웠다.

이 논문에서는 군집화기법과 XML기반의 서비스 기술표준인 WSDL의 의미적 가치를 활용하여 다수의 웹서비스를 군집화하는 프레임워크를 제안한다. 웹서비스 발견이라는 연구영역에 최초로 데이터마이닝 기법을 적용한 연구이다. 본 논문에서 제안하는 방식은 여러 흥미로운 요소들이 있다: (1) 서비스 사용자와 제공자들의 사전지식 요구를 최소화한다 (2) 특정 도메인에 과도하게 치중한 온톨로지를 피한다 (3) 웹서비스들 간의 의미론적 관계를 시각화할 수 있다. 이 논문에서 인공신경정신망 네트워크를 기반으로 하여 프로토타입 시스템을 개발하였으며, 실제 운용되고 있는 웹서비스 저장소로부터 획득한 실제 웹서비스들을 사용하여 제안하는 웹서비스 조직화 프레임워크를 실증적으로 평가하였으며 제안하는 방식의 효용성을 보여주는 실험결과를 보고한다.

Keywords : 웹서비스 발견, 데이터 마이닝, 인공신경정신망, 군집화, 의미론적 유사도, 평가

* 서울시립대학교 경영대학 교수

저자 소개



황유섭

현재 서울시립대학교 경영대학 교수로 재직 중이다. The University of Arizona에서 경영정보시스템을 전공하여 경영학사, 석사, 그리고 경영학 박사학위를 취득하였다. 미국 NASA와 Raytheon의 Hydrology Resource Management Project에 참여하였으며 Photogrammetric Engineering and Remote Sensing과 ER 학회지, Information Systems Review, 지능정보연구 등에 논문을 게재하였다. 주요 관심분야는 service-oriented computing, forecasting, artificial neural network의 활용 방안 연구, IT strategy 등이다.