Hierarchical Clustering Based Web Service Discovery

Huiying Gao[1], Susu Wang[1], Lily Sun[2], and Fuxing Nian[1]

[1] School of Management and Economics, Beijing Institute of Technology, Beijing, China
huiying@bit.edu.cn, susu_cheer@sina.com, nfx1228@163.com
[2] School of Systems Engineering, University of Reading, Whiteknights, UK
lily.sun@reading.ac.uk

Abstract. This paper presents a hierarchical clustering method for semantic Web service discovery. This method aims to improve the accuracy and efficiency of the traditional service discovery using vector space model. The Web service is converted into a standard vector format through the Web service description document. With the help of WordNet, a semantic analysis is conducted to reduce the dimension of the term vector and to make semantic expansion to meet the user's service request. The process and algorithm of hierarchical clustering based semantic Web service discovery is discussed. Validation is carried out on the dataset.

Keywords: Web service discovery, semantic analysis, hierarchical clustering, service matching, vector space model, Web service description.

1 Introduction

Web services describe a standardized way of integrating Web-based applications using open standards, such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web service description language (WSDL) and the Universal Description, Discovery, and Integration (UDDI) over an Internet protocol backbone[1]. Popova et al. [2] believe that Web service can enable the applications built on different servers to be accessed more easily, independent of platforms and programming languages. In recent years, especially with the development of cloud computing, different types of Web services are emerging [3]. As the rapid increase in the number of Web services, however, looking for the Web service in the online registry is like looking for a needle in the haystack. A better way for Web services discovery is to classify the Web services into different categories based on semantic analysis when they are published so that the matching algorithm can be done only in the related category with high efficiency and accuracy.

There are many approaches to build automatic taxonomies, such as rule based approach, semantic analysis, cluster analysis and learning algorithms. The trend by more and more taxonomy systems is to combine multiple methods based on various algorithms using statistical method, semantic analysis and clustering technique. This paper aims at a hybrid solution for Web service discovery by combining semantic analysis with the techniques of hierarchical clustering.

K. Liu et al. (Eds.): ICISO 2014, IFIP AICT 426, pp. 281–291, 2014.

The remainder of the paper is organised as follows: Section 2 discusses the state of art in relation to the methods of Web service matching. Clustering analysis is also argued. In Section 3 the semantic expansion of Web service is described and a hierarchical clustering based Web service discovery algorithm is put forwarded to classify and discover the Web services automatically. By using a Web service testing set, Section 4 illustrates the discovery process and validates the feasibility of the proposed method. Finally Section 5 draws the conclusion and flowed by the outlook.

2 Literature Review

2.1 Web Service Matching

A lot of research work has been conducted to solve the problem of Web service discovery. Two important ones are 1) Web service matching algorithm with high recall, precision and efficiency, and 2) the effective services selection and ranking methods in the found preliminary services set. The Web service matching is mainly through three stages [4]: a grammatical matching method based on keyword or vector space model such as UDDI systems of IBM, Microsoft, and SUN; a semantic matching method based on ontology such as OWL-S, METEOR-S and WSMO; a service matching method based on information search technology, for example, Niu et al. [5] have devised a semantic Web service discovery method based on context and action inference.

2.2 Clustering Algorithm

Clustering is a process of dividing data objects into several classes or clusters to make the similarity between different clusters maximal and the similarity between objects within one cluster minimal [6]. Clustering algorithm can generally be divided into five categories [7][8]: hierarchy-based approach, division-based approach, density-based approach, grid-based approach and model-based approach. Each method has its own advantages and disadvantages. As the hierarchy clustering method can show all the whole process of clustering, so we can confirm the number of categories by analyzing the process. It may reduce the time spending on determining the number of categories.

The basic hierarchical clustering algorithm can be divided into agglomerate hierarchy clustering algorithm (AGNES) and divisive hierarchy clustering algorithm (DIANA). The idea of agglomerate algorithm is that treat every object as a separate class, then merge the two closest clusters and update the original distance between clusters and the new cluster, repeat this step until all objects are in one cluster; while the idea of divisive algorithm is contrary [9].

As a technique widely used in the field of information search, clustering method improve the efficiency of information discovery to some extent. There are also some studies on the utilization of service clustering to improve efficiency of Web service discovery. For example, Rajagopal et al. [10] propose a service discovery method which is based on ontology clustering for grid service. Sudha et al. [11] complete the

service clustering based on the WSDL to improve the efficiency of service discovery. Sun et al. [12] cluster the services with high function and process similarities based on Petri Net. Wang et al. [13] propose a service discovery method by combining the ideas of P2P and clustering. Xu et al. [14] propose a discovery method based on graph theory. Yahyaoui et al. [15] introduce a novel matching approach which allows reducing the matching space through fuzzy classification rules. Facing the explosive increase of Web service, combine semantic analysis together with clustering technique to improve the accuracy and efficiency of Web service discovery is a trend.

3 The Method of Hierarchical Clustering Based Web Service Discovery

In our approach, we aim to realize the semantic Web service discovery process based on a hierarchical clustering. The advantage of this approach is that it can generate a hierarchically nested clustering and has high accuracy.

3.1 Web Service Description

In order to conduct similarity calculation on Web services, Web service description documents should be abstracted mathematically based on WSDL [16]. According to the structure of the WSDL document, Web service can be defined as follows.

Definition 1 [4]. Service description is an abstract of functional attributes and process model of Web service which can be expressed as a four-tuples:

$$WS =< sName, sDescription, sInput, sOutput >$$

among which,

$sName$ is the service name
$sDescription$ is the service description
$sInput = \{I_1, I_2, ..., I_m\}$ shows the input set of service
$sOutput = \{O_1, O_2, ...O_m\}$ shows the output set of service

Corresponding to the description of Web service, we can define the functional similarity of two services as follows.

Definition 2 [17]. Define the similarity of service ws_1 and ws_2 in Equation (1).

$$Sim(ws_1, ws_2) = \alpha_1 Sim(Sn_1, Sn_2) + \alpha_2 Sim(Sd_1, Sd_2) + \alpha_3 Sim(Si_1, Si_2) + \alpha_4 Sim(So_1 + So_2) \quad (1)$$

Among which, α_1, α_2, α_3, α_4 are respectively the weight of name similarity, description similarity, input similarity, output similarity of Web service in the whole similarity, and $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1, 0 \le \alpha_1 \le 1,\ 0 \le \alpha_2 \le 1, 0 \le \alpha_3 \le 1, 0 \le \alpha_4 \le 1$.

Service name similarity can be calculated by using:

$$Sim(Sn_1, Sn_2) = \frac{1}{p+q}\sum_{j=1}^{q}\sum_{i=1}^{p} Sim_{concept}(n_i^{(1)}, n_j^{(2)})$$

Service description similarity can be calculated by using:

$$Sim(Sd_1, Sd_2) = \frac{1}{p+q}\sum_{j=1}^{q}\sum_{i=1}^{p} Sim_{concept}(d_i^{(1)}, d_j^{(2)})$$

Service input similarity can be calculated by using:

$$Sim(Si_1, Si_2) = \frac{1}{p+q}\sum_{j=1}^{q}\sum_{i=1}^{p} Sim_{concept}(i_i^{(1)}, i_j^{(2)})$$

Service output similarity can be calculated by using:

$$Sim(So_1, So_2) = \frac{1}{p+q}\sum_{j=1}^{q}\sum_{i=1}^{p} Sim_{concept}(o_i^{(1)}, o_j^{(2)})$$

among which, p , q are respectively the number of concepts in S_n, S_d, S_i, S_o of the two services. If we ignore the difference between the name, description, input and output of service, then $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$.

Similar to the description of Web service, a service request can be defined as follows.

Definition 3 The description information of service request of a Web service requestor can be defined as the four-tuples：

$$RS = < rName, rDescription, rInput, rOutput >$$

in which,

$rName$ represents the service name that the requestor is looking for

$rDescription$ represents the service functional description that requestor needs

$rInput = \{rI_1, rI_2, ..., rI_m\}$ is the input set that service requestor provides

$rOutput = \{rO_1, rO_2, ..., rO_m\}$ is the output set that service requestor provides

Semantics is introduced through the calculation of concept similarity and in generally, the similarity between two concepts can be defined as follows.

Definition 4[18]. In a ontology category, the similarity of two concepts can be defined by Equation (2). The value of Sim is between [0,1].

$$sim(c_1, c_2) = \frac{2 * depth(LCS(c_1, c_2))}{dis(c_1, LCS(c_1, c_2)) + dis(c_2, LCS(c_1, c_2)) + 2 * depth(LCS(c_1, c_2))} \quad (2)$$

in which,

LCS refers to the smallest common ancestor node of two concepts,
depth represents the hierarchy depth of the repository,
dis represents the shortest path length between two concepts.

3.2 Web Service Discovery Processes Based on Clustering

Web service discovery consists mainly of two basic processes: the clustering and matching of Web services. The introduction of semantic technology and clustering method may affect both the two processes and ultimately impact the efficiency and accuracy of Web service discovery.

3.2.1 Process of Web Service Clustering

Web service can be seen as a kind of a short text. It can be described by using the service name, service description, service input and service output at the same time. So the process of text clustering can be equally applicable to Web service clustering as shown in Fig.1[19].

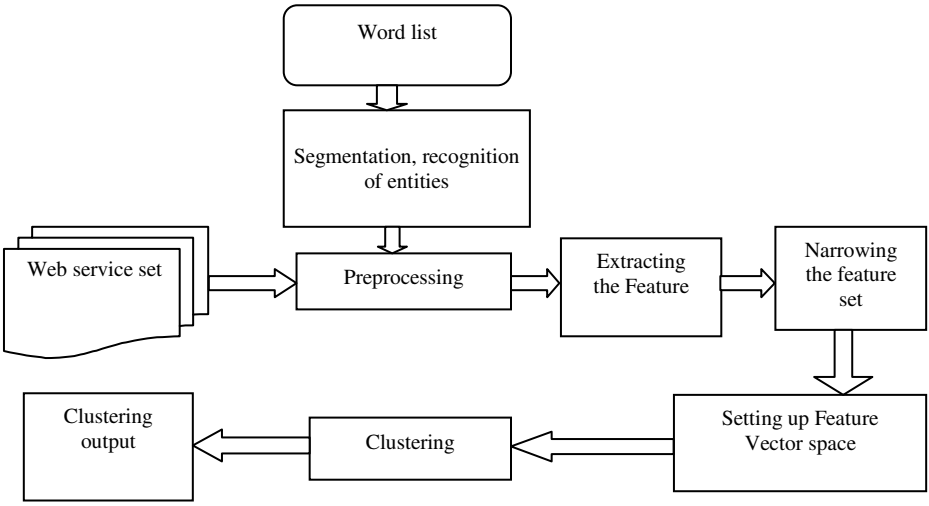

Fig. 1. The clustering process of Web service based on service description documents

The main process of hierarchical clustering are described as follows.

Step 1. Preprocessing of Web services: To preprocess the Web service description text to complete the Web service description segmentation and stemming process.

Step 2. The establishment of Web service feature space: There is a variety of feature representations of Web service information. We adopt the vector space model[20] (VSM) which is one of the methods that have been widely used and got better results in recent years and Term Frequency-Inverse Document Frequency (TF-IDF) which is a weighting technique commonly used for information retrieval and text mining.

Step 3. Reduction of Web service information feature set: Filter the words extracted from the Web service description text, remove prepositions, pronouns etc that have no actual meaning, and then stemming. For the feature vector space of Web service got from Step 2, calculate the similarity of feature words and reduce the dimension of the semantic vector space. According to definition 4, similarities can be calculated.

Step 4. The description of Web services collection: After reducing the feature set, the establishment of feature vector space of Web services is completed. Assuming that $WS = \{ws_1, ws_2, ..., ws_n\}$ represents a Web services set containing n Web services, in which ws_i represents the number i service, then the service set can be represented abstractly by Equation (3).

$$WS = \begin{pmatrix} tfidf_{1,1} & tfidf_{1,2} & ... & tfidf_{1,s} \\ tfidf_{2,1} & tfidf_{2,2} & ... & tfidf_{2,s} \\ ... & ... & tfidf_{i,j} & ... \\ tfidf_{n,1} & tfidf_{n,2} & ... & tfidf_{n,s} \end{pmatrix} \tag{3}$$

in which each row of the matrix represents a service, element $tfidf_{i,j}$ represents the weight of service i on feature word j, s is the dimension of feature word vector.

Step 5. Web services clustering: The prerequisite of clustering is determining the similarity between Web services. As mentioned before, TF-IDF is normally applying together with cosine similarity. Therefore, the cosine coefficients can be calculated by Equation (4) [21].

$$Sim(d_i, d_j) = Cos(d_i, d_j) = \frac{\sum_{k=1}^{|T|} tfidf_{i,k} \times tfidf_{j,k}}{\sqrt{\sum_{k=1}^{|T|} tfidf_{i,k}^2 \times \sum_{k=1}^{|T|} tfidf_{j,k}^2}} \tag{4}$$

in which $tfidf$ represents the TF-IDF weight of two Web service in the feature vector space, d represents the documents and T represents the feature word set.

3.2.2 Web Service Matching Process

Three main steps are involved in the Web service matching process.

Step 1. Service request description: according to definition 3, we can describe the service request as $R_{ws} = (Rt_1, Rt_2, ...Rt_i, ...Rt_m)$, in which Rt_i represents the number i feature word of request entered by users, m is the number of valid service request feature words.

Step 2. The semantic matching of service requests: according to definition 1 and 2, supposing that the feature vector of Web service set is $T = \{t_1, t_2, ..., t_s\}$, then the matching process of traditional method will be: for all feature word in T, judge whether there is R_{ws}. If there is, note the frequency of the word as 1; otherwise as 0; then treat it as a Web service to calculate the TF-IDF value in the space of T,

represented as $R_{tfidf} = (R_{tfidf,1}, R_{tfidf,2}, ..., R_{tfidf,m})$. Therefore, the matching process is namely the process of calculating the similarity between R_{tfidf} and all services in WS. Among them, the conversion from service request vector space to Web service feature word vector space can be achieved by Hungarian algorithm and the concept similarity is calculated based on WordNet.

Step 3. Service request matching based on clustering: After clustering based on Equation (4) the Web services storing in the database, the storage of Web service become different. Each cluster has a representative. If the new service has a higher similarity with a representative, it might mean that the similarity between the new service and all services in the cluster which the representative is in is higher than others. Before calculating the similarity between service request and all services, the similarity between Web service request and each representative should be compared. Select the clusters whose representatives have the top n similarity with the service request, calculate the similarity between the service request and all services in these clusters and finally return them to users by the similarity.

4 Results and Discussion

In our research work we choose OWL-TC 4.0 as the data set. It is the fourth version of OWL-S service test set which provides 1083 semantic Web services from nine different areas. The method mentioned in the paper and the traditional method of Web service matching is realized on the same data set. Comparative analysis is conducted on the recall, precision and time efficiency.

According to the advice of Lehmann that the number of categories should be between $n/30$ and $n/60$ [9] (in which n represents the number of samples), we can implement clustering using SPSS19.0 based on the above results. The cosine similarity can be the measurement of similarity between Web services according to the process of Web service clustering mentioned in Section 3.2.1. Comparing the clustering distribution with 19-36 clusters, some of the results are shown in Fig.2.

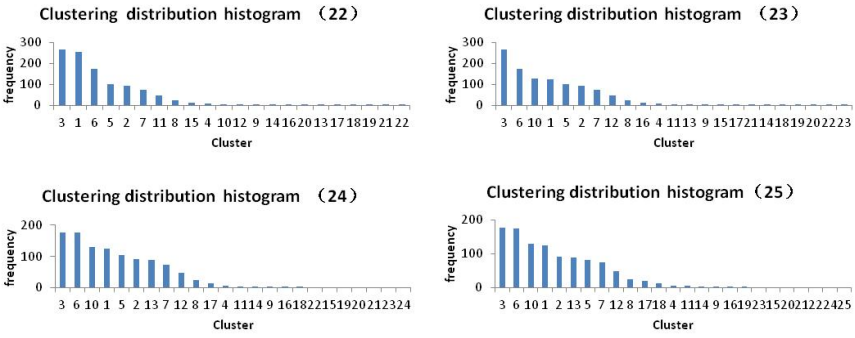

Fig. 2. The clustering distribution histogram with 22-25 clusters

The distribution with 24 clusters is relatively the most average and thereafter every time a cluster is added, the distribution only adds a cluster with the frequency less than 5 while the impaction on the clusters with higher frequency is weak. So considering both average and simplicity of computation, we choose 24 as the number of clusters of experimental data.

According to the Web service matching process of Section 3.2.2, we take "movie price" as a service request and complement the experiment while "movie" does not appear in the feature vector and "price" exists in the feature vector. The gravity of every cluster is taken as representative and cosine similarity between all representatives and service request is calculated by software Matlab to compare the results of matching. Table 1 shows the results of similarity between representatives and service request.

Table 1. The calculating results of similarity between representatives and service request

Cluster NO	Cluster size	Similarity (high to low)
13	88	0.483482
1	126	0.101886
10	129	0.081205
3	177	0.058091
11	5	0.053409
16	3	0.035541
20	1	0.020473
4	7	0.014712
9	3	0.014255
17	13	0.01176
18	3	0.009699
12	48	0.009568
5	103	0.00072
6	176	0.000445
2	92	0
7	74	0
8	24	0
14	4	0
15	1	0
19	1	0
21	1	0
22	2	0
23	1	0
24	1	0

After clustering, the top ten clusters are chosen which have the greatest similarity between representatives and service request and are returned to users in the descending order.

Taking the reality into consideration, users' browse focuses on the results in the top pages, we select the top 100 Web services to calculate the evaluation indicators after the matching results are returned. Let average of ten times calculating results be the indicator value of two methods on time efficiency and let

Recall rate = (number of related service discovered/total number of related)*100%

Precision rate = (number of related service discovered/total number of service discovered)*100%.

The results are shown in Table 2, from which we can see that the recall rate and precision rate in the experiment of the method proposed in this paper are 0,4167 and 0,2 respectively, both are higher than that of the traditional method without hierarchical clustering and semantic analysis. Yet the time consuming of our method is 0.1086 seconds which is less than that of the traditional method of 0.4441seconds.

Table 2. The comparison between Web service discovery method based on traditional keywords and hierarchical clustering (based on the top 100 return results)

	Recall	Precision	Time-consuming(s)
Traditional method	0	0	0.4441
This method	0.4167	0.2000	0.1086

The verification results show the advantages of hierarchical clustering based Web discovery on recall rate, precision rate and time efficiency which indicates the effectiveness of the method.

5 Conclusions

The method of hierarchical clustering based Web services discovery is devised to improve the efficiency and accuracy of Web service discovery. The hybrid solution for building and populating a taxonomy structure of web services are studied by combining semantic analysis with clustering techniques. The verification is carried out by using the dataset of OWL-TC4.0 to demonstrate the acceptance of the hierarchical clustering method in Web service discovery.

The work to convert the Web services into term vectors as well as the semantic expansion of the query of the user is done on WordNet, based on which a service provider and consumer would derive a meaning of the text or key word of the query or the Web service description document in a semantic level. However, a full understanding of web service potentiality can only be reached when the effects of the sign, made by contextual interpretation, is made by the service provider and consumer through consensus (pragmatics) [22]. In order to overcome limitations of this work, the future work will consider introducing domain ontologies in a semiotics way to make the Web service discovery context aware so as to improve the accuracy of

service discovery in further step. Additionally, more properties like the preconditions and results of Web services in the Web service description document should be taken into account when the semantic vector space are set up.

Acknowledgments. This work was supported in part by the Beijing Natural Science Foundation (Grant No. 9133020) and by the National Natural Science Foundation of China under Grant 71102111.

References

1. Thomas, E.: Service-oriented Architecture Concepts, Technology, and Design, pp. 56–80. China Machine Press (2007) (in Chinese)
2. Popova, G., Nedeva, V.: Web Services-an Instrument to Resolve the Problems of Information Systems Integration. Rakia Journal of Sciences, 61–64 (2006)
3. Yue, K., Wang, X., Zhou, A.: The Core Support Technology of Web Service: Review. Journal of Software 15(3), 428–440 (2004) (in Chinese)
4. Liao, Z., Liu, J., Liu, Y., Liu, H.: Review of Web Service Discovery Technology. Journal of The China Society for Scientific and Technical Information 27(2), 186–192 (2008) (in Chinese)
5. Niu, W., Chang, L., Wang, X., Han, X., Shi, Z.: Semantic Web Service Discovery Based on Context and Action Reasoning. Pattern Recognition and Artificial Intelligence 23(1), 65–71 (2010) (in Chinese)
6. Ertel, W.: Introduction to artificial intelligence. Springer (2011)
7. He, L., Wu, L., Cai, Y.: Summary of Clustering Algorithm in Data Mining. Computer Application Research, 10–13 (2007) (in Chinese)
8. Hu, Q., Ye, N., Zhu, M.: Summary of Clustering Algorithm in Data Mining. Computer and Digital Engineering (2007) (in Chinese)
9. Liang, B.: Detection of Top-n Global Outliers in Datasets Based on Hierarchical Clustering. Computer Engineering and Applications 48(9), 101–103 (2012)
10. Rajagopal, S., Selvi, T.: Semantic grad service discovery approach using clustering of service ontologies. In: Proceedings of IEEE TENCON 2006, pp. 1–4 (2006)
11. Sudha, R., Yousub, H., Zhao, H.: A Clustering Based Approach for Facilitating Semantic Web Service Discovery. In: Proceedings of the 15th Annual Workshop on Information Technologies & Systems, Las Vegas, USA (2006)
12. Sun, P., Jiang, C.: Process Model-oriented Semantic Web Service Discovery Using Service Clustering Optimization. Journal of Computers 31(8), 1340–1353 (2008) (in Chinese)
13. Wang, L., Hu, X.: Web Clustering and Composition based on P2P. Computer Engineering 35(17), 7–10 (2009) (in Chinese)
14. Xu, X., Chen, J., Wu, Y.: Web Service Discovery Method based on Clustering Optimization. Computer Engineering 37(9), 68–70 (2011)
15. Yahyaoui, H., Almulla, M., Own, H.: A Novel Non-functional Matchmaking Approach between Fuzzy User Queries and Real World Web Services Based on Rough Sets. Future Generation Computer Systems (2014)
16. Gao, H., Stucky, W., Liu, L.: Web Services Classification Based on Intelligent Clustering Techniques. In: Proceedings of 2009 International Forum on Information Technology and Applications, pp. 242–245 (2009)
17. Ma, Y., Jin, B., Feng, Y.: Semantic Web service Dynamic Discovery based on Evolution Distributed Ontology. Journal of Computers 28(4), 603–614 (2005) (in Chinese)

18. Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics, pp. 133–138 (1994)
19. Qing, Y., Gong, L., Xiang, L.: Text Clustering Algorithm based on Vector Space Model. Computer Engineering 34(18), 39–44 (2008)
20. Hamadi, R., Benatallah, B.: A Petri Net based Model for Web Service Composition. In: Proc. of the 14th Australasian Database Conference on Research and Practice in Information Technology, pp. 19–200 (2003)
21. Jing, P., Dong, Y.: A Text Clustering Algorithm Based on Semantic Inner Product Space Model. Journal of Computer 30(8), 1354–1362 (2007)
22. Liu, K., Benfell, A.: Software and Data Technologies Communications in Computer and Information Science. 50, 18–32 (2011)