A Fuzzy VSM-Based Approach for Semantic Service Retrieval

Supannada Chotipant[1], Farookh Khadeer Hussain[1], Hai Dong[2], and Omar Khadeer Hussain[3]

[1]Decision Support and e-Service Intelligence Lab (DeSI)
Quantum Computation and Intelligent Systems (QCIS) and School of Software
University of Technology Sydney (UTS), Sydney, NSW, Australia
Supannada.Chotipant@student.uts.edu.au,
Farookh.Hussain@uts.edu.au
[2] School of Computer Science and Information Technology, RMIT University, Australia
Hai.Dong@rmit.edu.au
[3]School of Business
University of New South Wales, ADFA, Canberra BC
ACT, 2601, Australia
O.Hussain@adfa.edu.au

Abstract. A vast number of business services have been published on the Web in an attempt to achieve cost reductions and satisfy user demand. Service retrieval consequently plays an important role, but unfortunately existing research focuses on crisp service retrieval techniques which are unsuitable for vague real world information. In this paper, we propose a new fuzzy service retrieval approach which consists of two modules: service annotation and service retrieval. Related service concepts for a given query are semantically retrieved, following which services that are annotated to those concepts are retrieved. The degree of retrieval of the retrieval module and the similarity between a service, a concept, and a query are fuzzy. Our experiment shows that the proposed approach performs better than a non-fuzzy approach on Recall measure.

Keywords: Fuzzy Service Retrieval, Semantic Service Annotation, Semantic Service Retrieval.

1 Introduction

Nowadays, a huge number of business services have been published on the Internet because companies desire to reduce costs and easily access their customers. Annotating services semantically enables machines to understand the purpose of services and can further assist in intelligent and precise service retrieval, selection and composition. Nowadays, meanings are manually ascribed to most services by service providers. Although this makes the results more acceptable, it is time-consuming, and there are difficulties in dealing with online tasks. The automation of service annotation is therefore desirable, but unfortunately no service annotation technique for service technology exists. Research about Web service annotation [1,2,3,4,5,6] concerns

C.K. Loo et al. (Eds.): ICONIP 2014, Part III, LNCS 8836, pp. 682–689, 2014.

semi-automated systems which suggest annotations to service providers. While [1], [6] are term-based annotation approaches, the others [2,3,4,5] are ontology-based annotation approaches. However, these approaches focus on only crisp annotation. To link the parameters of service descriptions to concepts of ontology is fuzzy. One parameter can be linked to several concepts with different degrees of relevance. Moreover, those works focus only on Web services. In this paper, we attempt to annotate business services to service concepts by using fuzzy variables.

Understanding the semantics of a query is the other significant factor for improving service retrieval performance. Zhai et al [7] focus on semantic query expansion by using ontology. In contrast, in this paper, we apply the ECBR algorithm [8] to semantically retrieve relevant services by using the synonyms of a query. Moreover, the intention of our approach is to overcome the issue of vague information, such as the similarity between a service and a concept, the similarity between a query and a concept, and the retrieval degree of a service. Therefore, we apply Fuzzy logic to the semantic service retrieval, which has not been done previously in the existing literature. Although there are some papers relating to fuzzy service retrieval [7], [9,10,11], their methods are based on fuzzy ontology, instead of crisp ontology which is used in this paper. Moreover, the existing work [9], [11] focus on retrieving Web services, while we focus on retrieving business service information on the Web.

The rest of this paper is organized as follows. In Section 2, we introduce the design of our semantic service retrieval system. The experiments and results are provided in Section 3, and the work is concluded in Section 4.

2 System Design

In this paper, we propose a Fuzzy VSM-based approach for a semantic service retrieval system, to enable users to search services based on service concepts.

The overall system architecture of the semantic service retrieval system is shown in Fig.1. The system consists of three main components, namely the service knowledge base, service annotation module and service retrieval module. The service knowledge base stores information related to services. It contains service ontology and service description entity (SDE) metadata. The service annotation module automatically annotates a SDE to relevant service concepts in the domain specific service ontology. The service retrieval module retrieves SDEs which are relevant to a user's query.

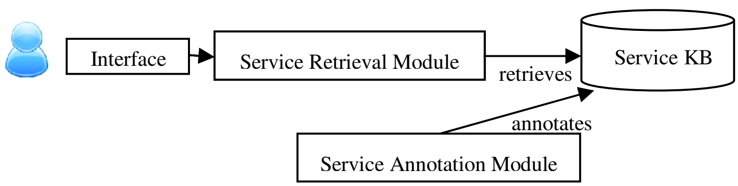

Fig. 1. Overall system architecture of the semantic service retrieval system

2.1 Service Knowledge Base

As mentioned above, the service knowledge base stores domain specific service ontology and SDE metadata. In this paper, we apply the service knowledge base from the work by Dong, Hussain and Chang [8]. The SDEs are easily linked to service concepts according to the categories of those services. In this paper, we propose a new service annotation method, which is presented in Section 2.2.

Service Ontology

The service ontology is a conceptualization of the services offered. Its structure is separated into four layers, namely the abstract concept layer, the service subdomain concept layer, the abstract service concept layer, and the actual service layer. Each service concept consists of a service concept name and service concept description. Only service concepts in the actual service layer are linked to SDE metadata.

Service Description Entity (SDE)

The SDE metadata is the information about the actual business services which may be relevant to more than one service concept. In this paper, the SDE metadata consists of five properties, namely linked concepts, service provider name, provider address, provider contact details, and SDE description. For example, the SDE metadata "Airline Agents Bookings" is assigned to "Virgin Airlines", "131 Fortitude Valley QLD, 4006 Australia", "Phone : 13 6789", and "Low Fares, Great Service." as service provider, provider address, provider contact details, and SDE description respectively.

2.2 Service Annotation Module

The service annotation module is a pre-processing element of the service retrieval system. The main purpose of this module is to automatically link SDE metadata to related service concepts. For example, the SDE called "Airline Agents Bookings" is connected to the service concept called "Airline_Booking". This enables the system to semantically retrieve the services associated with the relevant service concepts.

We present two approaches to SDE annotation, namely non-fuzzy and fuzzy service annotation respectively. The workflow of both approaches is presented in Fig. 2. The input data are SDE metadata and service ontology, while the output data are links between SDE metadata and service concepts. SDE metadata and service concepts are represented by a vector space model (VSM). While the VSM of a SDE is generated from the SDE name and description, the VSM of a service concept is created from service concept descriptions. The similarity between a SDE and a service concept is then calculated in the Matching Module using the cosine similarity between vectors. The similarity values range from 0 to 1. The higher the similarity value is, the more closely a SDE relates to a service concept. A value of 0 means that a SDE and a service concept are totally different. On the other hand, a value of 1 means that a SDE and a service concept are the same. The SDE-Service Concept Linking Module will subsequently link a SDE to a service concept if they are related. There is a difference in this step between a non-fuzzy system and a fuzzy system. If the SDE-Concept similarity value is greater than the link threshold (LT) in a non-fuzzy system, a SDE will be linked to a concept. In contrast, a SDE in a fuzzy system will be linked only if the similarity is greater than 0.

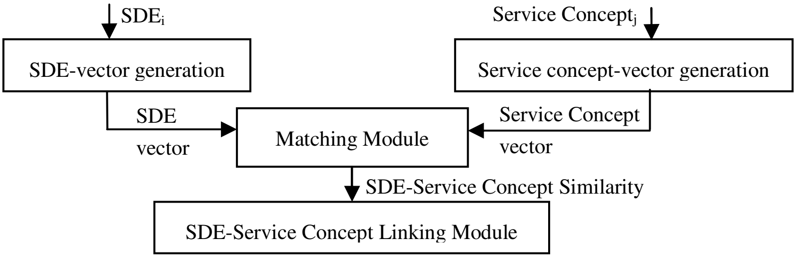

Fig. 2. System architecture of the service annotation module

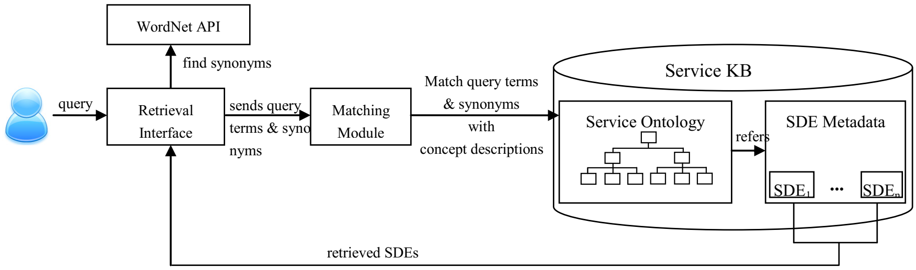

Fig. 3. System architecture of the service retrieval module

2.3 Service Retrieval Module

The service retrieval module fetches SDEs which are relevant to a user's query. For example, when a user sends the query "Flight booking service" to the system, the module retrieves related SDEs, such as "Airline Agents Bookings", which is provided by Air Niugini. The architecture of the service retrieval module is shown in Fig. 3.

First, the module receives a query via the retrieval interface. It extracts a set of separating terms from the query and removes the stop words. The retrieval interface then sends each query term to the WordNet API which returns the synonyms of the received term. The retrieval interface sends a set of query terms and their synonyms to the matching module, and we add those synonyms to the query terms. The matching module computes the similarity value between the set of query terms and each service concept in the service ontology. The processes of this module for non-fuzzy and fuzzy service retrieval systems are different. For the non-fuzzy system, arbitrary service concepts whose similarity values are greater than the retrieval threshold (RT) are selected, and the module refers to SDEs that are linked to the selected service concepts in the previous step. For the fuzzy system, the matching module applies Fuzzy logic to retrieve the relevant SDEs. The fuzzy rules and fuzzy membership functions are defined in Fig.4 and Fig.5 respectively. The fuzzy rules in this paper are quite simple. For example, if the similarity value between a query and a service concept is high, and the value between a SDE and a service concept is high, then the degree of retrieving a SDE is high. The variable $query\_concept\_sim_{(Q,C)}$ is the similarity value between a query and a service concept. The variable $sde\_concept\_sim_{(SDE,C)}$ is the value between a SDE and a concept, which is calculated in the service annotation process. The variable $retrieve_{(SDE)}$ presents the degree of retrieving a SDE. The values of these fuzzy variables are divided into three levels: high, medium, and low.

IF $query\_concept\_sim_{(Q,C)}$ is High and $sde\_concept\_sim_{(SDE,C)}$ is High
THEN $retrieve_{(SDE)}$ is High.
IF $query\_concept\_sim_{(Q,C)}$ is Medium and $sde\_concept\_sim_{(SDE,C)}$ is Medium
THEN $retrieve_{(SDE)}$ is Medium.
IF $query\_concept\_sim_{(Q,C)}$ is Low and $sde\_concept\_sim_{(SDE,C)}$ is Low
THEN $retrieve_{(SDE)}$ is Low.

Fig. 4. Fuzzy rules for service retrieval system

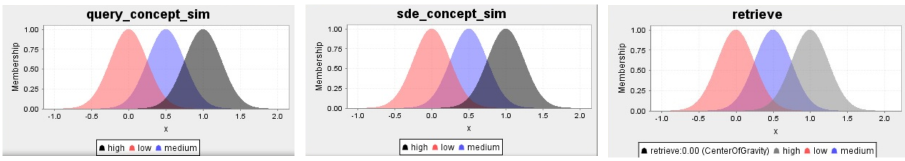

Fig. 5. Membership functions for service retrieval system

Query-Concept Similarity

We apply an extended case-based reasoning (ECBR) algorithm [8] to compute the similarity value between a set of query terms and a service concept. The main concept of the ECBR algorithm is to compare a set of query terms with the descriptions of a service concept. As previously mentioned, a service concept may contain many service concept descriptions. Therefore, the module will first compute the similarity values between query terms and each service concept description. Then, their maximum similarity values will be returned.

The similarity value for a service concept description is the summation of the matching value between each term and that concept description. We normalize the summation with the length of that concept description. If a query term from a user appears in the description, a value of 1 will be added to the matching value. On the other hand, if a synonym of a query term appears, a value of 0.5 will be added instead.

3 Experiment and Result

In this section, we compare the performance of the service retrieval system based on the non-fuzzy VSM model with the system based on the fuzzy VSM model in a Transportation domain ontology (TO) defined in [8]. We apply four performance measures from the area of information retrieval [12], namely Precision, Recall, Harmonic Mean, and Fallout Rate.

3.1 Experiment and Results

To evaluate the non-fuzzy system, we test the performance by setting the retrieval threshold (RT) for the ECBR algorithm as 0.8 and setting the link threshold (LT) in the service annotation module from 0.1 to 0.9 with an increment of 0.1. The performance of non-fuzzy service annotation and retrieval are shown in Table 1 and Table 2 respectively. For the fuzzy system, we set the RT as 0.8, and the retrieval degree (RD), the result of firing fuzzy rules, from 0.1 to 0.9 with an increment of 0.1. The performance of the fuzzy service annotation and retrieval are presented in Table 3 and Table 4 respectively. Note that we set the value of the RT as 0.8 because this value gives us the best performance values in term of Precision and Recall.

Comparing the Performance of Non-fuzzy with Fuzzy Service Annotation

The Precision values of non-fuzzy service annotation in Tables 1 and 3 are greater with every LT than those in the fuzzy system. It should be noted that all measure values (Precision, Recall, Harmonic Mean and Fallout) are the same values with different levels of RD, because the system links all SDEs that relate to a concept, even if that relationship is low. The performance of the fuzzy service annotation is equal to the performance of non-fuzzy annotation with LT value 0. Consequently, the Recall value of the fuzzy system is greater than all the Recall values of the non-fuzzy system. However, the non-fuzzy system performs better on Harmonic Mean and Fallout.

Table 1. The Performance of Non-Fuzzy Service Annotation for the TO, RT = 0.8

Link Threshold (LT)	Precision	Recall	Harmonic Mean	Fallout
0.1	17.43%	99.85%	11.01%	5.91%
0.2	20.34%	98.81%	12.86%	4.50%
0.3	27.14%	97.10%	16.20%	2.95%
0.4	34.08%	90.25%	18.01%	2.01%
0.5	42.08%	78.51%	21.13%	1.06%
0.6	53.95%	67.93%	24.18%	0.52%
0.7	59.72%	55.89%	24.48%	0.16%
0.8	56.28%	36.29%	19.34%	0.06%
0.9	54.98%	28.80%	16.71%	0.01%

Table 2. The Performance of Non-Fuzzy Service Retrieval for the TO, RT = 0.8

Link Threshold (LT)	Precision	Recall	Harmonic Mean	Fallout
0.1	67.59%	100.00%	37.67%	3.44%
0.2	70.23%	99.35%	38.91%	2.61%
0.3	78.83%	94.45%	41.49%	1.45%
0.4	87.95%	84.71%	41.77%	0.91%
0.5	82.15%	61.71%	33.04%	0.58%
0.6	69.55%	43.55%	25.94%	0.20%
0.7	75.29%	37.66%	22.91%	0.12%
0.8	60.00%	18.53%	12.44%	0.20%
0.9	62.50%	19.57%	12.29%	0.00%

Table 3. The Performance of Fuzzy Service Annotation for the TO, RT = 0.8

Retrieval Degree (RD)	Precision	Recall	Harmonic Mean	Fallout
All RDs	16.59%	100.00%	10.48%	6.48%

Table 4. The Performance of Fuzzy Service Retrieval for the TO, RT = 0.8

Retrieval Degree (RD)	Precision	Recall	Harmonic Mean	Fallout
0.1	67.38%	100.00%	37.58%	3.53%
0.2	67.38%	100.00%	37.58%	3.53%
0.3	67.38%	100.00%	37.58%	3.53%
0.4	67.38%	100.00%	37.58%	3.53%
0.5	67.56%	100.00%	37.67%	3.41%
0.6	87.34%	87.76%	42.36%	1.03%
0.7	79.30%	69.74%	35.57%	0.77%
0.8	78.17%	53.37%	29.61%	0.39%
0.9	57.84%	18.54%	11.66%	0.18%

Comparing the Performance of Non-fuzzy with Fuzzy Service Retrieval

Because the processes of retrieving the services for non-fuzzy and fuzzy systems are quite different, we will compare them with their maximum Precision values and we will focus on the performance of a non-fuzzy system with the LT value 0.4 and a fuzzy system with the RD value 0.6. Both non-fuzzy and fuzzy service retrieval systems perform well on Precision measure with values of 87.95% and 87.34% respectively. We observe that the Precision value of the fuzzy retrieval system is still high, although the value of the fuzzy annotation system is very low. This means that the Precision values of fuzzy retrieval system do not depend on those of fuzzy annotation system. With the same Precision, it can be seen that the fuzzy system performs slightly better than the non-fuzzy system – around 3% on the Recall measure. This is because fuzzy annotation links all related SDEs to a concept, which non-fuzzy annotation fails to do. As a result, the Fallout Rate of the fuzzy system is a marginally higher because it retrieves more non-relevant services.

4 Conclusion

In this paper, we proposed a Fuzzy VSM-based approach for semantic service retrieval. This approach is divided into two main modules: an annotation module and a retrieval module. The purpose of the annotation module is to prepare the Service Knowledge Base for the retrieval module. In this step, SDEs are automatically linked to relevant service concepts by comparing the similarity between their VSM-based vectors. The retrieval module semantically retrieves relevant service concepts using the ECBR algorithm and then retrieves SDEs that relate to selected concepts using Fuzzy technique. The experiment demonstrates that fuzzy service retrieval performs better on Recall than non-fuzzy service retrieval, but that both non-fuzzy and fuzzy retrieval perform well on Precision and Harmonic Mean.

References

1. Stavropoulos, T.G., Vrakas, D., Vlahavas, I.: Iridescent: A tool for rapid semantic annotation of web service descriptions. In: Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics, Madrid, Spain, pp. 1–9. ACM (2013)
2. Scicluna, J., Blank, C., Steinmetz, N., Simperl, E.: Crowd Sourcing Web Service Annotations. In: 2012 AAAI Spring Symposium Series (2012)
3. Patil, A.A., Oundhakar, S.A., Sheth, A.P., Verma, K.: Meteor-s web service annotation framework. In: Proceedings of the 13th International Conference on World Wide Web, pp. 553–562. ACM, New York (2004)
4. Hui, W., Zhiyong, F., Shizhan, C., Jinna, X., Yang, S.: Constructing Service Network via Classification and Annotation. In: 2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), pp. 69–73 (2010)
5. Bo, J., Zhiyuan, L.: A New Algorithm for Semantic Web Service Matching. Journal of Software (1796217X) 8, 351–356 (2013)
6. Meyer, H., Weske, M.: Light-Weight Semantic Service Annotations Through Tagging. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 465–470. Springer, Heidelberg (2006)
7. Zhai, J., Cao, Y., Chen, Y.: Semantic information retrieval based on fuzzy ontology for intelligent transportation systems. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2008, pp. 2321–2326. IEEE (2008)
8. Dong, H., Hussain, F.K., Chang, E.: A service search engine for the industrial digital ecosystems. IEEE Transactions on Industrial Electronics 58, 2183–2196 (2011)
9. de Castilho, R.E., Gurevych, I.: Semantic Service Retrieval Based on Natural Language Querying and Semantic Similarity. In: ICSC, pp. 173–176 (2011)
10. Madkour, M., Maach, A., Driss, E., Hasbi, A.: Fuzzy-based approach for context-aware service retrieval. In: 2012 Second International Conference on Innovative Computing Technology (INTECH), pp. 396–401 (2012)
11. Zhao, W., Peng, L., Zhang, J., Tian, C.: A fuzzy service matching algorithm based on Bloom filter. In: 2012 IEEE 14th International Conference on Communication Technology (ICCT), pp. 945–950. IEEE (2012)
12. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman, Boston (1999)