Web Service Categorization Using Normalized Similarity Score

Shalini Batra, Seema Bawa

Abstract—**Service discovery is one of challenging issues in Service-Oriented computing. Currently, most of the existing service discovering and matching approaches are based on keywords-based strategy. However, this method is inefficient and time-consuming. Based on the current dominating mechanisms of discovering and describing Web services with UDDI and WSDL, a novel approach for Web service categorization is proposed, where WSDL's documentation tag is used as only means to describe information pertaining to the entire Web service's functionality which is used in conjunction with the current Web service standards, to automatically categorize a Web service into a one of the pre-defined categories. The words are extracted from WSDL of a Web service and Nearest Similarity Score (NSS), a Measure of Semantic Relatedness (MSR) of each word is calculated with every pre-defined category. Total value of all the words is calculated through the NSS and then Web service is assigned a category based on the sum of MSR of all the words provided in the Web service description tag. This work enables automatic semantic categorization of Web services.**

Index Terms—**Measures of Semantic Relatedness, Nearest Similarity Score, Web services, Web Service Discovery**

I. Introduction

Web Service discovery process mainly involves locating desired services either published in a registry like UDDI or scattered in P2P systems, matching users' requirements to a set of services and returning **relevant** ones to the consumers. With the ever-increasing number of services published in Internet, finding desired services is just similar to looking for a needle in a haystack [1]. Efficiently finding Web services on the Web is a challenging issue in service-oriented computing. Currently, UDDI is a standard for publishing and discovery of Web services, and UDDI registries also provide keyword searches for Web services.

However, the search functionality is very simple and fails to account for relationships between Web services. Firstly, users are overwhelmed by the huge number of irrelevant returned services. Secondly, the intentions of users and the semantics in Web services are ignored. The capability of a Web service is often implicitly indicated through a service's name, a method's name and some descriptions included in the service and this capability can be described as an abstract interface by using standard Web Services Description Language (WSDL) [2].

Current technologies for publishing Web services, for example UDDI, enable providers to manually assign a category to their services from a number of predefined choices such as business, educational, finance, scientific, etc [4]. In the present day scenario service consumer has to manually search published services by category. Automatic mechanisms can help in assisting service publishers in the categorization task, in order to reduce the effort required, and promote globally consistent classification decisions, even when several users are involved [3]. Users will put a query and an automatic classifier will determine the most suitable categories where to look for the needed functionality. As a result, both service providers and consumers will be able to exploit Web service technologies in a better manner. The major features required in the service discovery interaction are that when the service provider advertises services the registry should locate suitable services and best matched services should be returned to the requestor as per his query.

Web services should be semantically annotated to provide the best match to the service requestor as per his requirements. In order to address these problems, a novel approach for efficiently finding Web services on the Web based on their associated semantics is presented in this paper. The main objective is to develop an effective mechanism for Web service discovery. The proposed approach puts a Web service into a specific category after calculating the Nearest Semantic Similarity, a probability based MSR, of a Web Service with a specific Web Service category. Our key contribution is a novel approach for service categorization by extracting N from the WSDL, provided by the service publisher or words extracted from the WSDL after the preprocessing and then calculating Nearest Semantic Similarity of each extracted word.

The rest of this paper is organized as follows. Section II discusses the related work in the area of Web service discovery. Section III provides the overview of the approach and Section IV describes methodology of our approach and its method of implementation. Finally, concluding remarks and future directions of research are considered in Section V.

II. Related Work

There are two major approaches considered in general centralized discovery architectures: the first one proposes the idea of semantically extending UDDI by mapping semantic

Shalini Batra is with Computer Science and Engineering Department, Thapar University, Patiala, Punjab, India (Ph. +91-9876173704) email: sbatra@thapar.edu.

Dr. Seema Bawa is with Computer Science and Engineering Department, Thapar University, Patiala, Punjab, India; email: seema@thapar.edu

annotations (*e.g.* DAML-S service profiles) to standard UDDI entries (*e.g.* tModels) and other where the matching algorithms are published as web services in the UDDI. The first one considers an extension to the UDDI registry to take advantage of the semantic service annotations, while using the popular and standardized UDDI infrastructure. Such translation of the semantic service advertisement or request to a UDDI-compliant representation combines the popularity and support of the UDDI standard and the semantically-grounded descriptions of services. An external matching engine is introduced that performs semantic matching on service capabilities. The second one is a more seamless integration of semantic matching mechanism with UDDI registry exists, where the matching algorithms are published as web services in the UDDI. UDDI detects and selects the most suitable matching service for each service request, and then invoke it for the actual semantic matching. The UDDI registry may offer the possibility of using multiple matchmaking services to fulfill a given service request. There may be matching service providers that offer diverse semantic matching mechanisms (*e.g.* implementing different algorithms or supporting different service annotation ontologies) [5].

Various approaches have been proposed for automatically or semi-automatically classifying Web services and some of these have been discussed in [4], [9]-[11] and [12]. After going through these approaches it has been observed that propose to classify Web services basing on the definitions of operation arguments that belong to a particular category while some of these approaches have low accuracy and some of these methods do not exploit a Web service interface description and its associated textual documentation. The main limitation of these matching approaches is that they do not attempt to reduce the distance between different styles for defining arguments present in standard descriptions.

The problem of identifying data types used by a Web service based on metadata is similar to the problems in Named Entity Reorganization, Information Extraction and Text Classification. Usually, fewer tokens are used in naming a data type compared to those in documents. Even though tokens from corresponding Web service message and operations, the number is very small. The text in a WSDL files are generally ungrammatical, noisy and varied. Such situations require some concrete meta data which can be used to semantically categorize the Web services [6].

III. Overview Of Our The Approach

A. EXTRACTING THE WORDS FROM THE WSDL

The most critical and important requirement to categorize the data semantically is that some standard should be followed in representation of the data in WSDL of every Web service and it is essential that all Web service developers structure the WSDL in a standard method. Once data is represented in the desired format, information extraction process can be followed in any one of the two ways:

From the document part of the WSDL the i.e. document tag </ documentation/> part which includes comments about what the service does. Instead of having useful comments, we propose that it can be made mandatory for every service publisher to give a set of n words, such that

$N = \{n_1, n_2, n_3, \ldots\}$

Where n_1, n_2, n_3, etc. are the words describing the service functionalities or in other words, the set correspond to most probable query words for the published Web service. Given a Web service with most useful words in the document tag the document part of the WSDL of a Web Service is accessed and the words are extracted.

i) Second approach for performing semantic categorization of the Web Services is to preprocessing the data after extracting the words from the WSDL file of the give Web service. The steps involved in prepossessing include detagging, tokenizing, stop word removal and stemming. First the names and comments are extracted and the combined names are split to generate different words. The terms are then filtered to remove the non relevant words, called stop words and stemming is done to reduce terms to their stems. Now the extracted terms of every pre-processed WSDL will be represented as a vector ~v = $(e_0, e_1, \ldots, e_n)$. Each element in the vector represents the importance of a distinct word w for that document If a term is represented two or more times it will be considered only once as it is representing the same concept or the word again and again.

If first approach is followed the heterogeneity in the data representation can be easily removed. This approach is equivalent to annotating a Web Service manually by the service publisher. The developer or publisher of a Web Service will give the most closely related terms in the documentation part of WSDL which best describe the service functionalities and capabilities, and these words can be easily extracted. This group of extracted words will serve as the input dataset for categorizing the similar Web services. Although any of the above approaches can be used the first approach is considered as a better alternate as it will give more meaning full and technical annotations to the Web services published.

B. MEASURING THE SEMANTIC SIMILARITY

Measures of Semantic Relatedness (MSRs) are statistical methods for extracting word associations from text corpora. Two of the varieties of MSRs are vector-based and probability-based. Probability-based MSRs, such as Point wise Mutual Information (PMI) [7] and Normalized Google Distance (NGD) [8], are easily implemented on top of search engines (like Google™ search) and thus have a virtually unlimited vocabulary. Vector-based MSRs, such as LSA [13] and GLSA [14], have the capability to measure relatedness between multi-word terms [15].

We use the probability-based MSR – Normalized Similarity Score (NSS). NSS is an MSR that is derived from NGD. To be more precise, the relatedness between two words *x* and *y* is derived as follows:

$NSS(x, y) = 1 – NGD(x, y$ (1)

where NGD is a formula derived by Cilibrasi, R., & Vitanyi, P. M. B [7]:

$NGD(x, y) = \underline{\max\{\log f(x), \log f(y)\}\ \log f(x, y)}$

$\log M \min\{\log f(x), \log f(y)\}$ (2)

Where M is the total number of web pages searched by Google; f(x) and f(y) are the number of hits for search terms x and y, respectively; and f(x, y) is the number of web pages on which both x and y occur. It is not necessary to use NSS only, as PMI and other similar metrics may be used. We chose NSS because some previous testing has revealed that overall it is a better model of language than PMI [16].

IV. Calculating The Msr Of The Words

The Web Services are initially categorized under seven different categories: *Zip Code, Country Information Stock Market, Temperature, Weather, Fax* and *Currency*. The extracted words of a particular Web Service are compared with each category say for example the words extracted from the WSDL of a Web service are pressure, humidity, rainfall, etc, all belonging to weather information are compared to all seven categories mentioned above and Normalized Similarity Score is calculated . Now the sum of all the Scores is calculated corresponding to every category and the highest cumulative score is indicative of the category to which the service belongs.

A. *Empirical Evaluation*

To evaluate the efficacy of the presented matching method WSDL related services was retrieved from X-Methods and seven categories were considered which included: *Zip Code, Country Information Stock Market, Temperature, Weather, Fax* and *Currency*. Most important and frequently used words were extracted from their WSDL and put in a file along with the name of the Web Service to which they belonged. Care was taken to include word giving the functional description of a Web Service and words which would normally come up for query corresponding to the published Web Service.

For experimental purposes six words {pressure, temperature, wind speed, rainfall, country and city} were considered as semantic annotations of published service. The NSS of each word was calculated with each category i.e. all six words were compared to category *Zip Code, then to Country Information, to Stock Market, Temperature, Weather, Fax* and *Currency* and sum of all words was computed with respect to each category. The value of NSS varies between 0 and 1 and more is the value, closer is the association of a word to the respective category.

If there are is a large set of Web services in a services repository and all of them are assigned in the one of the pre defined categories semantically and automatically then the job of discovering a relevant service is half done. Say if there is a set of five hundred Web services in a service repository and among those hundred are put under weather using the above method and a query related to weather come, out of five hundred services available only those assigned to weather category will be listed to the service requestor.

B. *Ranking the Services of a Particular Category*

Once the category has been assigned to a particular Web service the next job is to rank them semantically, according to requestor's requirements. There are many strategies to acquire a single-number dimension-independent measure in order to compare sets of matching pairs, the simplest of which is the matching average. Here |X| is the number of entries in the first part, |Y | is the number of entries in the second part, and |X \Y | denotes the number of entries that are common to both sets. Finally, |X \ Y | defines the number of entries in the first set that are not in the second, and |Y \ X| is the number of entries in the second set that are not in the first. Same procedure has been used by us to rank the services.

Categorization of the services using NSS and ranking of the service within a category ensures that user's request for a particular functionality will provide the set of semantically related Web services falling in that particular category. If the query is related to weather of a city then the top rated services will be those in the category of 'Weather' having the words 'rainfall' and 'city', followed by those having only 'rainfall' and then those having only 'city' and then rest of the services in this category will be followed.

Web service provider or developer should use technical words specifying the functional capabilities of the Web Services instead of using generic words. If general words are used rather than technical words the categorization would not be very efficient. Say we take 4 words Postal code, City, Region and Country and compared with all seven categories and the NSS calculated for all the categories gives maximum weight age to weather while just by looking at words it is clearly indicative of 'Zip Code' category. Hence we conclude that if specific and technical words are used in document tag of the WSDL results will more promising and accurate.

V. Conclusions

The future of Web services greatly depends on their ability to automatically identify the Web resources and execute them for achieving the intended goals of user. We have proposed a novel method that combines text mining and machine learning techniques for categorizing Web services. There is no need for text extraction or text normalization instead we just need to apply machine learning technique only to a pre-defined set of word. The methodology proposed by us will categorize the Web services in conjunction with the existing Web service technology, such as WSDL, to support a more automated service discovery process by calculating the NSS of the words with the specified categories and then ranking the services according to service requestor's query.

The research findings presented in this paper are based on Web services actually available on web. It has been discussed in some papers [16, 17] that the discovery and selection process of user-centered Web services involves a high degree of respect for user preferences to be flexible enough for real world use [18]. Based on such observations we propose that set of words should be provided by the service provider. One major finding in our experiment is that the set of words given by the service provider is static, that is, once the service provider has added a set of words they cannot be changed, we will try to introduce a mechanism that service consumer or requestor can provide new words in the existing set if he feels that such words will improve the recall of the respective Web Service. For example, if on querying a web service related to *temperature* for temperature conversion, a Web Service consumer gets an access to very good service but he realizes that word 'Celsius' should be added to the existing set of words in the Web Service related to *temperature*, then he should be allowed to do so. In other words, we can say that set N should be dynamic and then the categorization of Web services will be more precise and accurate. We are planning to develop a framework for the efficient discovery of Web services semantically using the proposed approach.

REFERENCES

[1] John Garofalakis, Y. Panagis, E. Sakkopoulo and A. Tsakalidis. "Web Service discovery mechanisms: looking for a needle in a haystack? ", International Workshop on Web Engineering, August 10, 2004.

[2] Jiangang Ma, Yanchun Zhang, Jing He, "Efficiently find ing Web services using a clustering semantic approach", CSSSIA 2008, April 22, Beijing, China , ACM ISBN 978-1-60558-107-1/08/04.

[3] Miguel Ángel Corella and Pablo Castells, "Semantic-based taxonomic categorization of Web services" 1st International Workshop on Semantic Matchmaking and Resource Retrieval: Issues and Perspectives (SMR 2006) at the 32nd International Conference on Very Large Data Bases (VLDB 2006). Seoul, Korea, September, 2006.

[4] Marco Crasso, Alejandro Zunino and Marcelo Campo, "AWSC: An approach to Web service classification based on machine learning techniques", Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No 37 (2008), pp. 25-36. ISSN: 1137-3601.

[5] Jorge Cardoso, Semantic Web Services: Theory, Tools, and Applications, Information Science Reference, University of Madeira, Portugal, ISBN 978-1-59904-045-5, 2007

[6] Kristina Lerman, Anon Plangprasopchok, Craig A. Knoblock, "Automatically labeling the inputs and outputs of Web Services", American Association for Artificial Intellegence, 2006

[7] Turney, P. (2001), "Mining the Web for synonyms: PMI- IR versus LSA on TOEFL", L. De Raedt & P. Flach (Eds.), Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001) (pp. 491-502). Freiburg, Germany.

[8] Cilibrasi, R., & Vitanyi, P. M. B. (2007), " The Google similarity distance", IEEE Transactions on Knowledge and Data Engineering, 19(3), 370-383.

[9] Nicole Oldham, Christopher Thomas, Amit P. Sheth, and Kunal Verma, "METEOR-S Web service annotation framework with machine learning classification", Semantic Web Services and Web Process Composition, Volume 3387 of LNCS, pages 137–146, San Diego, CA, USA, 2004. Springer.

[10] Miguel ´ Angel Corella and Pablo Castells, "Semi-automatic semantic-based Web service classification", Business Process Management Workshops, Volume 4103 of LNCS, pages 459–470, Vienna, Austria, September 4-7 2006. Springer.

[11] Zhang Duo, Li Zi, , and Xu Bin, "Web service annotation using ontology mapping", IEEE International Workshop on Service-Oriented System Engineering, 2005, pages 235–242.

[12] Andreas Heß, Eddie Johnston, and Nicholas Kushmerick, "ASSAM: A tool for semi automatically annotating semantic Web services", McIlraith et al., pages 320–334.

[13] Landauer, T. K., & Dumais, S. T. (1997), "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge", Psychological Review, 104(2), 211-240.

[14] Matveeva, I., Levow, G., Farahat, A., & Royer, C. (2005), "Term representation with generalized latent semantic analysis", Conference on Recent Advances in Natural Language Processing, 2005.

[15] Vladislav D. Veksler Ryan Z. Govostes Wayne D. Gray, "Defining the dimensions of the human semantic space" 30th Annual Meeting of the Cognitive Science Society. pp 1282-1287

[16] Lindsey, R., Veksler, V. D., Grintsvayg, A., & Gray, W. D, "Be wary of what your computer reads: The effects of corpus selection on measuring semantic relatedness", 8th International Conference of Cognitive Modeling, ICCM 2007, Ann Arobor, MI.

[17] W.-T. Balke, M. Wagner, "Cooperative discovery for user centered Web service provisioning", Proceedings of the First International Conference on Web Services (ICWS'03), Las Vegas, USA, 2003.

[18] W.-T. Balke, M. Wagner, "Towards personalized selection of Web service", Proceedings of the 12th International World Wide Web Conference (WWW 2003) Alternate Track on Web Services, Budapest, Hungary, 2003.

.

Mrs. Shalini Batra, Author Shalini Batra is working as Senior Lecturer in Computer Science and Engineering Department, Thapar University, Patiala since 2002. She has done her Post graduation from BITS, Pilani and is perusing Ph.D. from Thapar University in the area of Semantic and Machine Learning. She has guided fifteen ME s and presently guiding four. She is author/co-author of more than twenty-five publications in national and international conferences and journals. Her areas of interest include Web semantics and machine learning particularly semantic clustering and classification. She is taking courses of Compiler construction, Theory of Computations and Parallel and Distributed Computing.

Dr. Seema Bawa, Author Dr. Seema Bawa has done her M. Tech. from IIT, Kanpur and Ph.D. from Thapar University, Patiala. She joined Computer Science and Engg. Dept., Thapar University, Patiala as Asstt. Professor in 1999 and she is presently serving as Professor since 2004. She has guided four Ph.Ds and more than thirty M.E thesis. She has served Computer industry for more than five years before joining the University and has teaching experience of more than ten years. She has undertaken various projects and consultancy assignments in industry and academia. She is the author/co-author of more than 70 publications in technical journals and conferences. She has served as Advisor / Track chair for various national and international confrences. Her areas of intrest include Software Engineering, Parallel and Distributed and Grid Computing.