SHORT PAPER

International Journal of Recent Trends in Engineering, Vol 2, No. 3, November 2009

Semantic Categorization of Web Services

Shalini Batra, Seema Bawa
Computer Science and Engineering Department, Thapar University, Patiala, Punjab, India
Email: sbatra@thapar.edu, seema@thapar.edu

Abstract --Web services are new solution for dynamic business interactions over the Internet. Describing Web service with semantics provides the ability for automatic Web service discovery, invocation, composition and interoperation, and Web service execution monitoring. A critical step in the process of reusing existing Web services is the automatic discovery of potentially relevant components which can be achieved if some semantic description is provided in description of Web service. Approach proposed in this paper is to provide semantic annotation of Web service's functionality in WSDL's documentation tag which will be extracted and used for categorization of a Web service into one of the pre defined categories by calculating the Normalized Similarity Score (NSS), a Measure of Semantic Relatedness (MSR) of each word with every pre-defined category. Sum of NSS of all the terms in a particular Web service is calculated with all the categories and Web service is assigned a category with highest value. In contrast with previous approaches, this mechanism provides an efficient method for semantic indexing and approximate retrieval of services, leading to automatic semantic categorization of Web Services. Empirical evaluation of the work performed on a small test set is presented.

Keywords: Measures of Semantic Relatedness, Normalized Similarity Score, Web services, Web Service Discovery

I. INTRODUCTION

Web service discovery is normally defined as a matching process in which available services' capabilities can satisfy a service requester's requirements. 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) [13]. The interaction process for Web services consists of three distinct phases: a discovery of possible services, the selection of the most useful, and the subsequent execution. The first two phases are crucial and hence the focus is on the discovery and selection activity that is requested to identify the required Web services.

UDDI allows keyword-based search and category-based browsing of Web services. The keyword-based discovering mechanism supported by Universal Description, Discovery and Integration (UDDI) and most existing service search engines like Yahoo and Google, however, suffer from some key problems. Firstly, it is difficult for a user to obtain the desired services because the number of the retrieved services with respect to the keywords may be huge. Secondly, keywords are insufficient in expressing semantic concepts and semantically different concepts could possess identical representation (homonyms) which will further lead to low precision. As a result, the retrieved services might be totally irrelevant to the need of their consumers.

When it comes to finding appropriate services and composing distributed applications, current technologies still require a large amount of human interaction. This results in the restricted number of use cases for service integration and the limited scalability of solutions involving manual activities in the process of service discovery and compositions. In order to address these problems, the idea of supplementing Web services with a semantic description of their functionality, that could facilitate their discovery and integration, has been proposed by many researchers. However, the information, on which these semantic descriptions are based, is rarely discussed. In addition, there is still no established method for acquiring semantic Web service descriptions.

Classification or categorization is the task of assigning objects from a universe to two or more classes or categories and automatic text classification is an important task that can help people in finding information from huge online resources. 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. Since the categorization of services and maintenance of repositories has to be done manually by human entities, the classification task becomes considerably difficult, heavy and error-prone in practice, due to several issues (e.g. huge size of taxonomies in real-world applications, multiple people involved in maintaining or sharing services in a common repository, several distributed repositories being shared, etc.). 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 [16]. 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 steps required in the service discovery interaction steps for Semantic Web services or traditional Web Services are:

Step 1: The service provider advertises services,

Step 2: The requestor searches for services fulfilling its functional requirements,
Step 3: The registry/matching engine locates (and selects) suitable services,
Step 4: The best matches are returned to the requestor.

To reuse the service components and have efficient service discovery, various matchmaking algorithms have been proposed. Approach proposed in this paper is to provide semantic annotation of Web service's functionality in WSDL's documentation tag which will be extracted and used for automatically categorization of a Web service into one of the pre defined categories by calculating the Normalized Similarity Score (NSS), a Measure of Semantic Relatedness (MSR) of each word with every pre-defined category. Sum of NSS of all the terms in a particular Web service is calculated with all the categories and Web service is assigned a category with highest value.

Our key contributions are:

i. Functional description of Web service operations at development stages by service published or developer.
ii. Automatic semantic categorization of service by using Nearest Semantic Similarity.
iii. Preliminary experiment to support the effectiveness of our approach.

The rest of this paper is organized as follows. The next section discusses the most relevant previous approaches. Section III describes methodology of our approach and its implementation. Then, in Section IV we report the evaluation of our approach. Finally, concluding remarks and future lines of research are described in Section V.

II. Related work

Approaches for automatically or semi-automatically classifying Web services have been proposed in [4, 5, 6, 7 and 14]. However, these approaches have some limitations. Some of them have low accuracy while some of these propose to classify Web services basing on the definitions of operation arguments that belong to a particular category. 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. MWSAF [8] is an approach for classifying Web services based on argument definitions matching. METEOR-S [5] describes a further improved version of MWSAF. The problem of determining a Web service category is abstracted to a document classification problem. The graph matching technique is replaced with a Naïve Bayes classifier. To do this, METEOR-S extracts the names of all operations and arguments declared in WSDL documents of pre-categorized Web services [4].

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 are extracted, 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 [1].

III. Overview of the approach

A. Extracting The Words From The WSDL

Each Web service developer has a unique way of structuring the WSDL; hence it is critical to ensure that the input data is kept to the required standardization. To categorize a Web Service semantically some meta data is required *i.e.* some textual information and the only way to get the data for a Web Service is to extract the terms from the WSDL of a Web Service. Information Extraction process can be followed in two ways:

i) Semantic Annotation of a Web service**:** The approach proposed here is that instead of having useful comments in documentation tag </ documentation/> of WSDL, it can be made mandatory for every service publisher to give a set of n words, which best describe the service functionality, *i.e.*

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

where n_1, n_2, n_3, etc. are the set of words describing the service or in other words, we can say the set correspond to most probable query terms for the web service published. Given a web service with most useful terms in the document tag we can easily access the document part of the WSDL of a Web Service and extract the terms.

ii) The traditional approach : Extracting all the terms form the WSDL and then preprocessing the extracted terms

If second approach is considered for performing semantic categorization of the Web Services then preprocessing steps 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 =(e0, ..., en). Each element in the vector represents the importance of a distinct word w for that document and then their NSS with respect to each category will be calculated. 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.

Although any of the above approaches can be used we consider the first approach as a better alternate as it will give more meaning full and technical annotations to the Web services in hand. The major advantage foreseen by applying first approach is that it will overcome the heterogeneity in the data representation. Since the developer or publisher of a Web Service is the best judge of his service functionalities and capabilities, terms most closely to Web service functionality will be provided in the documentation part of WSDL which can be easily

extracted, which will serve as the input dataset for grouping the similar Web services. In other terms it is equivalent to annotating a Web Service by the service publisher.

B. Measuring the Semantic Similarity

Measures of Semantic Relatedness (MSRs) are computational means for assessing the relative meaning of terms. More specifically, MSRs take the form of computer programs that can extract relatedness between any two terms based on large text corpora. They 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 PMI [2] and NGD [3], are easily implemented on top of search engines (like Google™ search) and thus have a virtually unlimited vocabulary. Vector-based MSRs, such as LSA [9] and GLSA [10], have the capability to measure relatedness between multi-word terms [17].

In our experiments, probability-based MSR – Normalized Similarity Score (NSS) has been used. NSS is an MSR that is derived from NGD. The relatedness between two words *x* and *y* is derived as:

$$NSS(x, y) = 1 - NGD(x, y) \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

where NGD is a formula derived by Cilibrasi, R., & Vitanyi [3]:

$$NGD(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log M - \min\{\log f(x), \log f(y)\}} \quad \ldots\ldots. \quad (2)$$

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 [15].

IV. Evaluating MSR and Categorizing the Services

Initially seven categories were considered: *Zip Code*, *Country Information Stock Market*, *Temperature*, *Weather*, *Fax* and *Currency*. The extracted terms of a particular Web Service are compared with each category say for example the terms 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 (Details provide with the help of example in Section 4.1).

A. Empirical Evaluation

A collection of related services from were retrieved from X-Methods and put under seven categories as mentioned above. Most important and frequently used terms were extracted from their WSDL and put in a file along with the name of the Web Service to which they belonged and the NSS of each word was calculated with all categories. The NSS of every word extracted from a service related to category country information is given in Table 1. Throughout our experimentation we have used a publicly available MSR Web Server [http://cwl-projects.cogsci.rpi.edu/msr] for calculating NSS.

Table 1: NSS of terms with category Country

	Longitude	Latitude	Zip Code	Bar code	FIPS	Time zone	Area Code	Ph. No.	Country Code	ISO
Country	0.381	0.411	0.481	0.303	0.297	0.402	0.261	0.273	0.409	0.346

When same terms are compared to category *weather* the NSS generated is shown in Table 2:

Table 2: NSS of terms related to category Weather

	Longitude	Latitude	Zip Code	Bar code	FIPS	Time zone	Area Code	Ph. No.	Country Code	ISO
Weather	0.831	0.81	0.768	0.357	0.453	0.743	0.526	0.75	0.251	0.488

Now words were extracted from various services and NSS was calculated for the terms extracted from their WSDL with each category *i.e.* if the extracted terms were pressure, temperature, wind speed, rainfall, country and city, then these terms were compared with category *Zip code* and NSS was calculated, then same terms were compared with category country and so on with all the predefined seven categories.

Table 3 given below has seven categories represented in columns and six terms {pressure, temperature, wind speed, rainfall, country and city} and the value in each row represents the NSS with each category. This value varies between 0 and 1 and more is the value, closer is the association of a word to the respective category.

The total of all the values in the last row clearly indicates how much similarities these words have to a particular category. Referring to Table 3 it is seen that the total of 4.961 is achieved for words like pressure, wind speed, etc and hence this Web service will automatically fall under the category weather. On similar lines a Web

service which will have most words related to stock markets will have maximum NSS total for stock market category. Now if we have a set of 400 or 500 Web services in hand and they are all categorized in the one of the pre defined categories semantically and automatically then our job of discovering a relevant service is half done. Say we have a set of 500 Web services and among those 100 are put under weather using the above method and a query related to weather come, out of 500 services available only 100 will be shown to the requestor.

Table 3: NSS of each word with each category

Category / Words	Zip Code	Country info.	Stock Market	Temp.	Weather	Fax	Currency
Pressure	0.563	0.368	0.566	0.483	1.0	0.418	0.902
Temperature	0.151	0.684	0.252	1.0	0.513	0.407	0.776
Wind speed	0.374	0.03	0.0	0.273	0.745	0.0	0.327
Rainfall	0.155	0.302	0.0	0.354	0.737	0.065	0.584
Country	0.755	0.396	0.246	0.311	0.983	0.35	0.804
City	0.797	0.352	0.564	0.207	0.935	0.521	0.805
Total	**2.795**	**2.132**	**1.628**	**2.628**	**4.961**	**1.761**	**4.298**

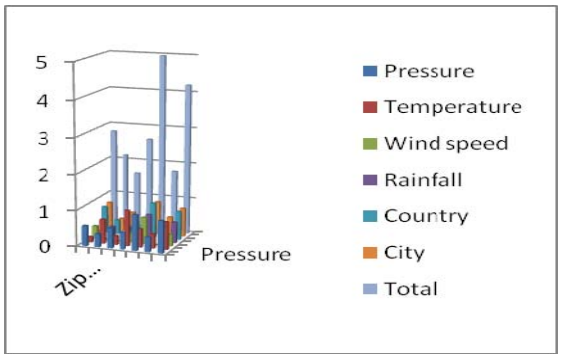

Figure 1: Graphical representation of Table 3

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 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. We have ranked the services on this basis. To speed up our access further we have arranged all the terms of a particular Web service in alphabetic order *i.e.* say there are two Web services in the category of 'weather' and the terms in first Web service are 'rainfall, pressure, humidity and precipitation' and in other Web service are 'pressure, temperature, wind speed, humidity and city'. Both are sorted alphabetically and hence the first Web service words will be put as 'humidity, precipitation, pressure, rainfall and those of second will be 'city, humidity, pressure, temperature, wind speed'.

If the user query is for 'Weather' category and the query is "Find the rainfall of New York 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. An important observation made by us is that technical terms specifying the functional capabilities of the Web Services should be used instead of using generic terms. If general terms are used rather than technical terms the scenario would be something of the type shown in Table 4.

Table 4: NSS of each word with each category

Category / Words	Zip Code	Country info	Stock Market	Temp	Weather	Fax	Currency
Postal Code	0.408	.774	0.0	0.273	0.447	0.295	0.483
City	0.797	1	0.564	0.82	0.935	0.521	0.805
Region	0.648	1	0.188	0.729	1.0	0.226	0.863
Country	0.755	1	0.246	0.783	0.983	0.35	0.804
Total	**2.608**	**3.774**	**.998**	**2.605**	**3.395**	**1.392**	**2.955**

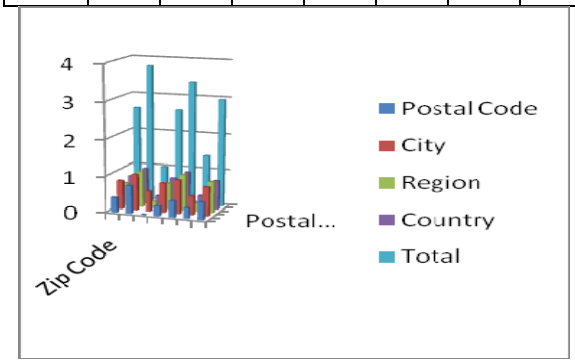

Figure 2: Graphical representation of Table 4

V. CONCLUSIONS

As seen in Table 4, four words Postal code, City, Region and Country are compared with all seven categories and the Total indicated in the end row gives maximum weight age to category *country* while just by looking at words it is indicative of '*Zip Code*' category (Figure 2). Hence the conclusion is that if specific and technical terms are used in document tag of the WSDL, categorization will be more efficient

It has been discussed in some papers [11, 12] 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 [10]. Based on such observations we propose that annotations should be provided by the service provider. 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. The advantage of the proposed methodology is that it works in conjunction with the existing Web service technology, such as WSDL, to support a more automated service discovery and ranking of the services.

The research findings presented in this paper are based on Web services actually available on web. We are planning to develop a framework for the efficient discovery of Web services semantically using the proposed approach.

REFERENCES

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

[2] 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.

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

[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] 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.

[6] 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.

[7] 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.

[8] Abhijit A. Patil, Swapna A. Oundhakar, Amit P. Sheth, and Kunal Verma, "METEOR-S Web service annotation framework", Proc. of the 13th international conference on WWW. ACM Press, 2004.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Jiangang Ma, Yanchun Zhang, Jing He, "Efficiently Finding Web Services Using a Clustering Semantic Approach", *CSSSIA* 2008, April 22, Beijing, China, ACM ISBN 978-1-60558-107-1/08/04.

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

[15] 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.

[16] 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.

[17] 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, 2007, pp 1282-1287.