Applying Association Rules For Web Services Categorization

Shailja Sharma and Shalini Batra

Abstract- **With the growth of the number of Web Services available on the Web, the need for mechanisms enabling the automatic categorization of Web Services becomes important. Web Services classification, a task of automatically sorting a set of documents into categories from a predefined set is an attractive method as manually organizing document bases is simply not feasible given the time constraints of the application or the number of documents involved. We propose an approach of Web Service classification where text mining and classification of WSDL (Web Service Description Language) documents is done based on association rules *i.e.* association rules are applied to analyze the degree of dependency between contents of WSDL and category of the Web Services. A machine learning classifier is used at the end to categorize the documents under different categories. This classifier deduces a sequence of candidate categories for a preprocessed Web service description. The performance of web services classified with association rules is compared with the primitive classification algorithms.**

Index Term--**Association Rules, Text Mining, Web Services classification**

I. Introduction

Web Services are loosely coupled reusable software components that encapsulate discrete functionality and are distributed and programmatically accessible over the internet. They are self contain, modular business applications that have open, Internet-oriented, standards-based interfaces [1]. Automatic text classification is an important task that can help people finding information on huge online resources. Classification or categorization is the task of assigning objects from a universe to two or more classes or categories. Current technologies for publishing Web Services, for example UDDI[2], enable providers to manually assign a category to their services from a number of predefined choices such as business, educational, finance, scientific, etc .

Assigning a proper category to a Service can be a tedious and error prone task due to the large number of categories usually present in Web Services registries.

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. 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 Services technologies in a better manner.

In this paper we will exploit dependency between the Web Services category and its interface described by WSDL [3] document for classifying Web Services. Association rules have been used for building Web Services Classifier for automatically classifying Web Services; this is, determining the category of a Web Services, given a set of predefined categories. The main goals of this paper are to

1) Build a classification system using association rule applied on the category of a Web Services, its operations and its textual documentation, namely argument definitions and comments written by developers.
2) To analyze how the importance of a term to a particular category varies with its frequency and appearance in other documents.
3) To analyze whether this system gives better accuracy than the primitive methods used for classification or not.

II. Related Work

The work on association rule mining began with the development of the Apriori algorithm [4], and was further modified and extended. Since then, several attempts have been made to improve the performance of these algorithms. The partition algorithm [5] partitions the data into disjoint groups, process each individually, and merge the intermediate results. It improves the overall performance by reducing the number of passes needed over the complete database to at most two. The incremental mining algorithm [6] incorporates the concept of data addition to validate the existing rules. The task of deriving new rules for data sets that grow incrementally is supported by the incremental algorithm.

During the past few years some efforts and research have been placed on assisting the developer to classify Web Services. As a result, some semiautomatic and automatic methods have been proposed. MWSAF [7] is an approach for classifying Web Services based on argument definitions matching. First, MWSAF translates these definitions into a

Shailja Sharma is System Analyst with Kurukshetra University, Kurukshetra, Haryana,India. Email: shailjakaushik@gmail.com Ph. no. 9896097938
Shalini Batra is Sr. Lecturer with Computer Science and Engineering Department, Thapar University, Patiala, India. Email: sbatra@thapar.edu

graph. Then, MWSAF uses graph similarity techniques for comparing both. The main limitation of these matching approaches is that they do not attempt to reduce the distance between different coding conventions. In fact, MWSAF achieves low accuracy, which is shown in its implementation.

METEOR-S [8] describes a further improved version of MWSAF. The problem of determining a Web Services category is abstracted to a document classification problem. To do this, METEOR-S extracts the names of all operations and arguments declared in WSDL documents of pre-categorized Web Services. The main limitation of this approach is that it assumes independence between the name of an operation and its arguments. Although METEOR-S proposes a document classification approach, natural language documentation, usually present in WSDL files and service registries, is not considered.

ASSAM [9] is a machine learning approach for determining Web Services category. It combines the Naive Bayes and SVM machine learning algorithms to classify WSDL files in manually defined hierarchies. Previous efforts for classifying Web Services have several shortcomings. First, the classification approach proposed by MWSAF has shown low accuracy. Second, the classification-based version of MWSAF shown better accuracy, but this version is based on the false premise that an operation and its argument names are independent. After going through the literature related to Web Services classification it has been analyzed that majority of approaches don't consider Web Services interface description and its associated textual documentation, so we thought of working in this direction and apply various information retrieval techniques to the data extracted from interface and textual description associated with the Web Services.

III. Web Services Classification Process

Web Services Classification is the act of determining a category of a Web Services, from several pre defined categories [10]. The automatic classification is done on the basis of information provided by the WSDL documents. There are two stages in the classification process [11] as shown in figure I:

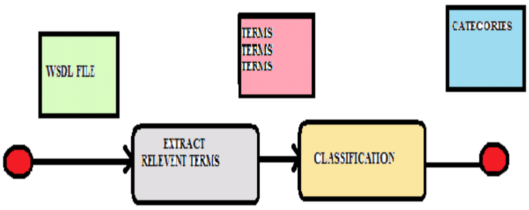

Figure I: Two step classification process

- Text preprocessing or text mining
- Classification

Text mining techniques have been designed for preprocessing textual documentation, operations and arguments accompanying descriptions of Web Services in the WSDL.

The process uses a supervised document classifier which deduces a sequence of candidate categories for a preprocessed Web Services description.

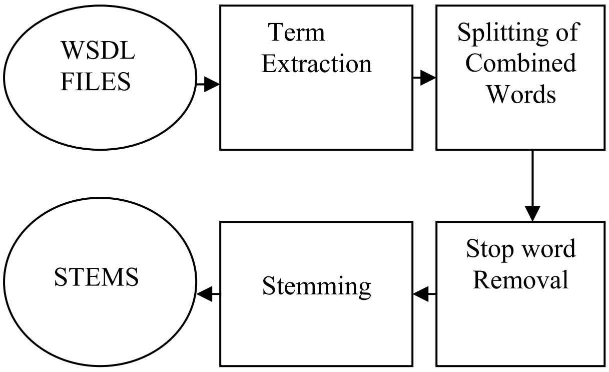

Figure II: Text Mining Process.

Preprocessing includes Detagger, (removing all the tags from the XML document), Tokenizing (breaking the stream of text into distinct meaningful units or tokens), Stop word removal (discarding of certain key phrases and words that may otherwise taint the algorithms and Stemming (reduction of words into their root). After extracting the terms from the WSDL document every document will be represented as a vector ~v = (e0, ..., en). Each element ei represents the importance of a distinct word wi for that document. This importance will be calculated according to the selected word weighting method, TF-IDF, a word weighting heuristic that determines whether a word is important for a document if it occurs very often. For each term ti of a document d weight can be calculated by using the formula:

$$\text{Term Weight} = \quad w_i = tf_i * log\left(\frac{D}{df_i}\right) \tag{1}$$

Where

tfi = term frequency (term counts) or number of times a term i occurs in a document.

dfi = document frequency or number of documents containing term i

D = number of documents in the database.

A. *WSDL Classification*

Document classification refers to the process of assigning an electronic document to one or more categories based on its contents [12].

Automatic service classification helps during:

1. **Service publication** (to classify the new service) and

2. **Service retrieval** (to identify the classes where to restrict the focus of the query) [13].

Automatic document classifiers support classification of documents seen as objects characterized by features extracted from their contents. In general, when some external mechanism, such as human feedback, provides information on the correct classification for documents, we talk about supervised document classification. This approach consists of two phases:

- Training phase.
- Classification phase.

During the training phase, such a learning system receives a collection of categorized documents and builds a classifier. Then, during the classification phase, this classifier deduces one or more categories for a new document. Automatic classification assumes that:

1) The category of a Web Services depends on its textual comments in its WSDL.
2) Method signatures i.e. the operation name and arguments.

It is desirable to measure the dependencies between categories and arguments, i.e., operation interfaces. By finding this dependency we can create category vectors for each category in the pre defined set.

IV. Implementation And Results

In order to implement Association rules and classification algorithms on our data set WEKA [14] and Rapid Miner [15] toolkits are used. Performance of a classifier based upon Association rules has been examined and analyzed to find whether it gives better results than the primitive classification algorithms.

A. Preprocessing

Data set considered contains words extracted from plain text description for each service and the WSDL document associated with each service.

Data sets used here are collection of 165 Web services whose WSDL documents are downloaded from different UDDI repositories including Xmethods.com, SALCentral.com, service-repository.com, etc. We have developed a script in C that parses a WSDL document and pulls out the comments associated with the service, its operations and arguments.

All the WSDL files were preprocessed using the following steps:

1) First argument declarations and comments from each WSDL document were extracted.
2) Tokens were generated using the tokenizer.
3) Stop words were removed from the file. A text file “stopword.txt” containing list of stop words was given as a input to the toolkit. Based upon this text file all stop words were removed.
4) Stemming was done to bring the words to their base words. Porter stemmer was used.
5) TF-IDF was generated .From this TF-IDF matrix the words with high importance were extracted out. Initially, dependency between categories and WSDL contents was measured. An important observation was that an argument is more significant to a category if it has a high importance in the given category but a low importance in the whole collection of categories. Here, “importance” refers to the TF-IDF value for an argument. This hypothesis was verified on a subset of the Web services collection composed of 165 hand-classified Web services, (Table I).
6) Once TF-IDF matrix is calculated then comma separated files (.csv) and attribute relation file format (.arff) files are prepared.

TABLE I: Categories for Web Services.

Category of Web Services	**No. of Web Services**
Country (a)	62
Communication (b)	38
Business (c)	51
Finance (d)	14

B. Association Rules generation

After the file has been preprocessed we tried to find out interface patterns within category related services by using Association Rules, specifically, the Tertius[16] and PridictiveApriori algorithm from Weka and Rapid miner. Some discovered rules are presented below. Association rules generated using PridictiveApriori algorithm are:

1) service_parameter = Stock Quote Service Info ==> class = d {0.080467}

2) service_parameter = Quote Stock credit==> class = d{0.080467 }

3) service_parameter = City Country Codes Response ==> class = a{0.069187 }

4) service_parameter = Phone Number validate Verify ==> class = b{0.074537}

Rules generated form predictive Apriori algorithms are:

1) Class=a ==> service_parameter=Phone Number Area Code Prefix Number Country State City County acc :(0.05141)

2) Class=b ==> service_parameter=Phone Val Response Return Object Status Description acc :(0.06949)

Number of rules generated in Rapid Miner were many times more than the rules generated by WEKA toolkit. Many rules were redundant and the confirmative value was very low of the order around 0.005000.The average accuracy of Association rule based classifier was found to be 30.9091%.

C. Classification based upon Naïve Bayes classifier on the same data set:

The Naïve Bayes categorization approach is a simple probabilistic technique. Probabilities that documents belong to various categories are determined from estimated probabilities that words belong to different categories [17]. The implementation results shows that the Naive Bayes classifier performs better than the classifier based on association rules and its effectiveness is achieved with 80% accuracy.

D. Observations

Recent research in text mining and machine learning have shown significant improvement in automatic classification and labeling of documents. Our implementation tried to find out the association relation between category of Web Services and WSDL descriptions and devise rules for automatic classification system for the Web Services.

The results from our implementation (applying association rules on Web Services) have shown that although there were some degrees of dependency between the category of a service and terms extracted from description document of Web Services i.e. WSDL file, but still the discovered rules were far away from becoming the bases of a classification system, mainly due to the variety of argument names that Web Services can employ. An important observation was that importance of an argument to a particular category increases proportionally to the number of times an argument appears in the services of this particular category, but this importance decreased if an argument is common in the whole collection. By applying Naïve Bayes classifier on the same data set and comparing to results with association rules, the new results appear to be significantly better with a high accuracy of 80%.

TABLE II : COMPARISON OF BOTH METHODS

Classifier	Average Accuracy
Association Rule based	30.9091%
Naïve Bayes	80.0000%

V. FUTURE SCOPE

In future we are planning to add semantic support to the Web Services classification system so that intelligent publishing as well as search of Web Services can be implemented. Further we are also working to devise a tool which extracts only method signatures from WSDL document automatically to improve the classification accuracy.

REFERENCES

[1] Gustavo Alonso, Fabio Casati, Harumi Kuno, Vijay Machiraju, “Web Services – Concepts, Architectures and Applications”, Springer Verlag, Berlin Heidelberg, 2004.

[2] Curbera, F. Duftler, M. Khalaf, R. Nagy, W. Mukhi, N. Weerawarana, “Unraveling the Web Services, Web: an introduction to SOAP, WSDL, and UDDI”, Internet Computing, IEEE, mar/apr 2002, Volume: 6, issue:2.

[3] World Wide Web Consortium (W3C), “Web Services Description Language (WSDL) 1.1”, http://www.w3.org/TR/wsdl/, 2001.

[4] Thomas, S. and S. Chakravarthy, “Incremental Mining of Constrained Associations”, Proc. of the 7th Intl. Conf. of High Performance Computing (HiPC), 2000.

[5] Shenoy, P., “Turbo-charging Vertical Mining of Large Databases”, ACM SIGMOD Int'l Conference on Management of Data, 2000, Dallas.

[6] Thuraisingham, B., “A Primer for Understanding and Applying Data Mining”, IEEE, 2000. Vol. 2, No.1: p. 28-31.

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

[8] Nicole Oldham, Christopher Thomas, Amit P. Sheth, and Kunal Verma, “METEOR-S Web Services 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.

[9] Andreas Heß, Eddie Johnston, and Nicholas Kushmerick, “ASSAM: A tool for semiautomatically annotating semantic Web Services”, CSCWD 2008, 12th International Conference on Web Technologies, pages 470–475.

[10] R .Agrawal, T. Imielinski, A. Swami, “Mining Association Rules between Sets of Items in Large Databases", Proc. SIGMOD Conference, 1993.

[11] Marco Crasso, Alejandro Zunino and Marcelo Campo, “AWSC: An approach to Web Services classification based on machine learning techniques”, CONICET, Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No 37 (2008), Pages 25-36.

[12] Suman Saha,C. A. Murthy and Sankar K. Pal, “Classification of Web Services Using Tensor Space Model and Rough Ensemble Classifier”, 2008.

[13] Marcello Bruno, Gerardo Canfora, Massimiliano Di Penta, and Rita Scognamiglio, “An Approach to support Web Services Classification and Annotation”, In Proc. the IEEE International Conference on Web Technologies, 29 March-1 April 2005 Pages 138 – 143.

[14] Eibe Frank, Mark Hall, Geoffrey Holmes, Richard Kirkby, Bernhard Pfahringer, Ian H. Witten, and WEKA A Machine Learning Workbench for Data Mining”, Pages 1305-1314, Berlin: Springer, 2005.

[15] Rapid-I GmbH Stockumer Str. 475 44227 Dortmund,Germanyhttp://www.rapidminer.com/

[16] Peter A. Flach and Nicolas Lachiche. “Confirmation-guided discovery of first-order rules with Tertius”. Machine Learning, 42(1/2):61–95, January 2001.

[17] C. J. Fall, K. Benzineb, “Literature survey: Issues to be considered in the automatic classification of patents”, Volume 1.0, CLAIMS, WIPO, Geneva, Switzerland, Oct 2002.

Shailja Sharma was born in Karnal, India in 1984. She received her B.Tech degree in Computer Sc. & Engineering from Kurukshetra University, Kurukshetra in 2005, M.B.A from Guru Jambheshwer University, Hisar and M.E degree from Thapar University, Patiala in 2009.Currently, she is serving as System Analyst in Kurukshetra University, Kurukshetra. She is author/co-author of 4 publications in national and international conferences. Her research interests are focused on Data Mining and Machine Learning.

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.