

Web Services Ontology Population through Text Classification

José A. Reyes-Ortiz Autonomous Metropolitan University Azcapotzalco, Mexico City, Mexico Email: jaro@correo.azc.uam.mx Maricela Bravo Autonomous Metropolitan University Azcapotzalco, Mexico City, Mexico Email: mcbc@correo.azc.uam.mx Hugo Pablo Autonomous Metropolitan University Azcapotzalco, Mexico City, Mexico Email: hpl@correo.azc.uam.mx

Abstract—In this paper, we describe the process by which web services ontologies are populated from a web services collection. The general approach relies on a global ontology model that is used to represent automatically web services. The model is enriched with web service instances classified into a taxonomy. The main idea is to extract taxonomic relations (*isTypeOf*) from web services using a supervised classifier of textual descriptions attached to web services. The entire process for ontology population involves the following tasks: text extraction from web service descriptions, classification of text descriptions and extraction of taxonomic relations (instances of classified web services). An experimentation was carried out with a collection of web service, which shows promising results and the feasibility of our approach.

Index Terms—web services classification; ontology population; web services ontology population; text classification.

I. INTRODUCTION

W EB services are reusable software components through which it is possible to build and integrate new applications without having to implement all elements of a system. Nowadays, Web services have become more popular due to their proliferation for offering storage services and resource management in the cloud. Web services are available in both public and private repositories using descriptions in XML and natural language, such as: English, Spanish or German. There are several public repositories of Web services, for example: SOAP Web service directory supported by Membrane¹; Repository of Visual Web Service²; ProgrammableWeb³; OWLS-TC⁴.

Web services are described using the standard WSDL and OWL-S. Both consist of an XML file, in which necessary elements to achieve a detailed description of web services is defined.

Programmers and application developers can use web services like software components, but they need search them into a large volume of web service published in repositories. This task is commonly known as web services discovery. However, web services discovery remains a difficult and error-prone task, since web services repositories offer keywords-based search mechanisms. In addition, web service repositories

are organized in static structures that do not allow a flexible and dynamic organization of services. As a solution to this problem, ontologies can organize web services repositories with semantic relations and taxonomic relations in order to offer a semantic organization of them. In addition, this structure can help to discover and test web services in the work presented in [1].

In this paper, we present an approach for web services classification based on the frequency of *1-grams* (words), in order to populate a web services ontology. The main aim of this paper is to improve the structure repositories in order to facilitate web services discovery by providing an ontology-based semantic structure.

The rest of the paper is organized as follows. In Section II, we report the state of the art related with classification and ontology population in web services. Section III describes Web Services Descriptions Language (WSDL) and an ontology-base service description language (OWL-S). Section IV presents the global ontology model used for populate web services. Section V presents our approach for web services ontology population. Experiments and results are presented in Section VI. Finally, conclusions and future work are shown in Section VII.

II. RELATED WORKS

Web services are described using parameters (input and output) names, data type names and operation names. These elements have been exploited for several purposes. As in [2] and [3] that have used text processing with parameters and operations names in order to obtain the similarity between web services; content-based approaches for web services classification have been proposed in [4], [5], [6], [7], [8] and [9], which are described below. Web services clustering from WSDL documents in order to facilitate web services discovery proposed in [10] and [11], also in [12] a clustering-based approach to web service taxonomy is presented.

In this paper, we rely on text classification to enable web services ontology population. Text classification is a task widely used, as in [13]. Moreover, web services classification is a task that has been addressed in different ways. Regarding with web services ontology population using text descriptions

¹http://www.service-repository.com/

²http://www.visualwebservice.com

³http://www.programmableweb.com

⁴http://projects.semwebcentral.org/projects/owls-tc/

classification of web services, poor works have been propounded. However, web services classification is close to web service ontology population since it needs a class name to be instantiated. Thus, we have conducted our research of related works in two aspects: approaches for supervised classification of web services; and works for ontology population.

Using OWLS-TC collection, we have the following works: [4] uses web services textual descriptions, such as: operations, inputs/outputs textual descriptions. They classify web services with a *Naive-Bayes* classifier. In [5], the classification of web services is based on support vector machine algorithm and it is achieved by calculating a similarity between words using *WordNet* and a domain taxonomy in order to reach an efficient classification of web services in the collection. A similar web services classification has been proposed in [6], in this case it is based on sets, they propose a representation of web service descriptions with vector space model and an entropy-based weighting of all terms.

There are works that not use the public collection mentioned above. They use private collections of web services. [7] showed that using quality attributes (reliability, documentation, performance, and response time), it is possible to classify and predict the quality of a web service; they have used a private collection of 364 web services. Unsupervised classification is applied in [8], where an automatic classifier is presented based on tags embedded in WSLD documents for each web service, its method was tested with 951 web services distributed in 19 categories. In addition, other relevant work is presented in [9], they expose a text mining approach to web services classification by identifying key concepts in textual documentation services, but only in a specific domain.

Another field for this paper is web services ontology population. Some works that have presented ontology learning methods includes ontology population for web services domain. As in [14] that enhances an existing ontology with similarity relations between operation of web services. An ontology learning mechanism is proposed in [15] in order to enable *RESTful* semantic web services using syntactic and semantic descriptions. And [16] propose an automatic extraction method that learns domain ontologies from textual documentations attached to Web services.

III. WEB SERVICES DESCRIPTION

The recommended SDL for the Web Service implementation is named Web Service Description Language (WSDL), which is currently a well-established W3C standard. WSDL defines an XML grammar for describing networked services as collections of communication endpoints capable of exchanging messages. In this work, we consider WSDL 2.0, that is the latest version, which incorporated important changes in the description of a service. WSDL 2.0 changes the *definitions* tag with the *description* tag. The main difference between WSDL 2.0 and previous versions are: the *targetNamesSpace* is a required attribute of the definitions element in WSDL 2.0; message constructs are removed in WSDL 2.0; operator overloading is not supported in WSDL 2.0; *PortType* is renamed as Interface; Interface inheritance is supported by using the extends attribute; and Port is renamed as Endpoint.

Also, OLW-S is an ontology-based service description language, which provides a semantic description of web services. OWL-S is based on *Service* class that presents a *Profile* class in order to describe the service functionality, which is described by *Service Model* class. OWL-S contains descriptions in natural language focused on the understanding by humans. A challenge is faced when this user-focused description need to be processed.

IV. GLOBAL ONTOLOGY MODEL

This section presents a global ontology model exposed in [17], which is used to populate web services from text descriptions in our approach. It is also widely explained here.

We use Manchester Syntax for OWL 1.1 [18] in order to present the global ontology due to it is a user-friendly syntax. Thus, the ontology model created in [17] represents the following relevant classes for our approach.

Class:	GeneralService	
Class:	GeneralOperation	
Class:	GeneralEffect	
Class:	GeneralPrecondition	
Class:	GeneralParameter	
Class:	InputPara	meter
SubClassOf: GeneralParamete		
Class:	OutputPara	ameter
SubClassOf: GeneralParameter		

The ontology model also includes object properties, relation between classes, as follows:

```
ObjectProperty: hasOperation
Domain: GeneralService
Range: GeneralOperation
ObjectProperty: hasEffect
Domain: GeneralOperation
Range: GeneralEffect
ObjectProperty: hasPrecondition
Domain: GeneralOperation
Range: GeneralPrecondition
ObjectProperty: hasInput
Domain: GeneralOperation
Range: InputParameter
ObjectProperty: hasOutput
Domain: GeneralOperation
Range: OutputParameter
```

The idea of this paper is that the web service instances are created and classified into subclasses of *GeneralService* class and thus, we have an extended ontology model from presented in [17]. Under this assumption, Figure 1 shows nine subclasses that we have proposed to be populated.

V. WEB SERVICE ONTOLOGY POPULATION

Web Service Ontology Population is carried out by text classification of each Web service description in WSDL tags and natural language. Text classification associates predefined

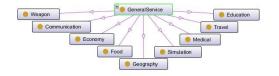


Fig. 1. Our taxonomy proposed from GeneralService class

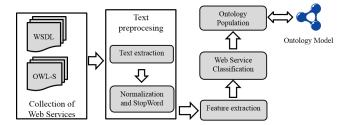


Fig. 2. Architecture of Web service ontology population

categories to a Web service from text analysis. Figure 2 shows our architecture for web service ontology population from web service descriptions through text classification. From a general ontology model, our process of ontology population is able to discover taxonomic relations to enhance our initial model and thereby achieve the classification of web services.

Web service ontology population uses a collection of public web services that are classified in any of nine proposed categories. Each web service in our collection has associated a WSDL description and an ontology-based extension of this description in OWL-S. We have proposed four phases in order to populate a web service ontology, which are described below.

A. Text preprocessing

WSDL files and OWL-S files that describes web service functionality are analyzed in order to identify and extract the text descriptions, which is useful for achieve contentbased classification. Thus, the content of *serviceName* and *textDescription* labels from *Profile* class is extracted, such labels contain natural language text. From WSDL file, the service name (*wsdl: service name*), names of operations (*wsdl: operation name*) and the data type names of messages, both simple or complex (*xsd: simpleType name* and *xsd: complexType name*). This natural language text and full names of elements are used to characterize, classify and represent the web service instances into the corresponding class in our extended ontology model.

Text extracted is segmented into canonical words. In web services domain is common to find complex names of services, operations and data type names, which are formed by two or three complex words. They are segmented into lexical words by considering the switching from lowercase to uppercase and the use of subscripts as separator of lexical words. As an example: *getAddressLocation* or *get_address_location* is

separated into the following lexical words: [get] [address] [location].

In addition, the texts are normalized by applying a conversion to lowercase, removing punctuation and stop words, which they not add meaning to the services and therefore are considered non-functional for classification based on service content.

B. Features extraction and selection

Lexical words of each web service were normalized and selected in order to represent a space model to have a formal representation of each web service. This model uses the lexical words (*1-gram*) like features. From our collection of web services 1801 words are extracted and represented in vectors.

Features are weighted using the *Bag-of-Word(BoW)* model, which consists of a collection of texts and their vocabulary that is considered features in our research. Each web service is expressed like a vector $S_j = (w_{1j}, w_{2j}w_{nj})$, where w_{ij} expresses the relevance that produces a feature *i* in a text *j*. In our case, a lexical word (*1-gram*) of vocabulary represents a feature *i* and web service descriptions represent a text *j*.

The approach used to obtain the relevance of a feature (*1-gram*) in the text of a web service is based on applying Term Frequency Inverse Document Frequency (*TF-IDF*) in order to determine what lexical word in the collection of texts might be more important for a text description of a Web service. This weighting uses the Term Frequency (*TF*) of our vocabulary in a text (1) and Inverse Document Frequency (*IDF*) that determines whether the term is common in the text collection (2), then the final equation to calculate a *TF-IDF* of a *1-gram* is shown in (3).

$$TF(t_i, S_j) \tag{1}$$

In order to determine the value of *Term Frequency* (TF), we use the number of times that the term t occurs in a web services description *S*.

$$IDF(t_i, S_j) = \log \frac{|S|}{1 + |sS: t_i s|} \tag{2}$$

Inverse Document Frequency (*IDF*) is obtained by dividing the total number of web service descriptions by the number of web service descriptions that contain the term, and then, logarithm of such quotient is taken.

$$w_{ij} = TF(t_i, S_j) \times IDF(t_i, S_j)$$
(3)

A feature selection process was carried out in order to reduce our vector space. The aim of features selection is to reject irrelevant features to obtain the best subset to improve the accuracy in web services classification. For this process, we apply an analysis of attribute correlation and wrapping scheme based on decision trees, widely used as the features selection algorithm, in order to select relevant features. Such process has obtained a subset of 69 lexical words (*1-gram*) like relevant features.

C. Web service classification

Web services classification is based on features extracted from web service descriptions in WSDL and OWL-S and weighted with *TF-IDF*. The features are represented in vectors, which are used by a supervised classifier, widely used in machine learning to estimate the predictive function of each class of a collection.

As a supervised classifier is used to obtain the category of a web service, we have divided our collection in two sets: a training and test set. The main aim of this phase is to build a classifier of web services using their text description. Nine categories are considered by the classifier to be built: *Communication, Economy, Education, Food, Geography, Medical, Simulation, Travel and Weapon.* They are automatically assigned to each web service.

A rule-based classifier was built under machine learning schema using the *1-gram* words like features. C4.5 [19] was used like classification model for web services. This classifier builds decision trees from a set of training data by a selection of the best attributes that obtain effectively partitions from training set, then, the rules of the decision tree are applied to the test set in order to evaluate the results.

C4.5 classifier was trained with 899 web service descriptions, adjustments in the trees are applied for obtaining the corresponding rules, which are used with the test data for obtaining the similar output. The difference in the output determines the performance of our web services classifier. We have implemented the C4.5 for web services classification using WEKA tool [20].

D. Ontology population

This section presents the process of automatic population of Web services ontology. This task uses the assigned categories in the previous phase, they are used to create class instances in order to populate the global ontology model used for the representation of Web services.

Ontology population is the process of adding class instances and relation instances between individuals of the ontology into an existing ontology [21]. Thus, ontology population takes the category assigned to each Web service and it creates an ontological instance into the corresponding class from global ontology model. A *isTypeOf* relation is created between the instance of the Web service and the corresponding class.

Our ontology population process is illustrated with an example. Given a pre-processed text of the web service identified like *ActivityDestination*, as shown below, we determine his class in order to be populated.

Web service description: name: activity destination service; operation name: get destination; input name: activity name; output name: destination name; description: this web service provides the destinations where an activity is available.

The following code represents the population of class *Travel* service with the *ActivityDestination* service.

Individual: ActivityDestination

```
Types: Travel
Facts: hasOperation GetDestination
Facts: hasName "Activity destination
```

service"

And the *GetDestination* operation has an input and an output, which are described as follows:

```
Individual: GetDestination
Types: GeneralOperations
Facts: hasInput ActivityName
hasOutput DestinationName
```

VI. EXPERIMENTATION AND RESULTS

The main idea of our experiments is to evaluate the classification task, which was carried out with C4.5 classifier. Thus, with the evaluation of our classification task, we are evaluating the correct population of global ontology model used to represent Web Services. It is because that the population of web services ontology corresponds to create instances in the class discovered by C4.5 classifier.

According to the assessment presented above, we carried out our experiments. We use selected word of the vocabulary like features.

Our evaluation was carried out with version 3.0 of OLWS-TC collection, which consist of 1129 web services described using WSDL and OWL-S. As this collection is considered to test, their web services are pre-classified into the following classes:*Communication, Economy, Education, Food, Geography, Medical, Simulation, Travel and Weapon.* It was divided into two groups: 899 web services for training web services classifier and 230 web services for testing our approach.

The experimentation was performed with 899 web services in order to obtain the classification model and then, it was applied to 230 web services to be tested. Also, they used the vector of *1-grams* like features weighted with *TF-IDF* for each web service and the *C4.5* classifier with the following parameters: *confidence factor used for pruning the tree* = 0.25 and *the minimum number of instances per leaf* = 2.

We evaluate the results in terms of Precision(P), Recall(R) and F-measure(F1). Metrics widely used in classification tasks. In our case, these metrics compare the results to be evaluated with external values of trust (web services previously classified).

Table I shows the results of our evaluation, which consists of testing *C4.5* classifier with features extracted (*1-grams*) and weigh them using *TF-IDF*.

The results showed in Table I emphasize that the use of selected features (words) present promising results reaching 92.1 % of correctly classified web services.

We have used OLWS-TC collection for our experiments. This collection has been used as a benchmark for works that have proposed web services classification. Therefore, it is possible to compare our results with previously proposed approaches, such as Zhang and Pan [4], Wang et al. [5], Chen et al. [6] and Yuan and Jian [8]. In Table II a comparison of proposed approaches with our approach is presented in terms

 TABLE I

 Results of the Web services classification

Class	Precision	Recall	F1
Communication	0.927	0.879	0.903
Economy	0.955	0.953	0.954
Education	0.843	0.944	0.891
Food	0.857	0.706	0.774
Geography	0.982	0.900	0.939
Medical	0.955	0.863	0.906
Simulation	0.933	0.875	0.903
Travel	0.961	0.909	0.934
Weapon	0.974	0.925	0.949
Average	0.924	0.921	0.921

of accuracy, representing the number of correctly identified true or false classifications of web services.

TABLE II RESULTS WITH OWL-TC COLLECTION

Approach	Accuracy
Zhang and Pan	0.41
Wang et al.	0.89
Chen et al.	0.85
Yuan and Jian	0.87
Our approach	0.921

The results show the effectiveness of our approach compared with other approaches under the same Web services collection and the same evaluation criteria. Although the results are not so encouraging for *Food* class as for other classes, our solution can help developers to recovery Web services and reuse existing software components in a disorganized repository of web services.

VII. CONCLUSIONS AND FUTURE WORK

This paper has presented an approach for web service ontology population through a text classification technique. We have employed C4.5 classifier with *1-grams* using the weighting *TF-IDF* like features to represent a web service using the vector space model. A selection of features was carried out in order to reduce the space of representation, obtaining 69 words like relevant features.

The main contributions of this paper are as follows: (a) we present an approach to classify web services using their descriptions and a text classification technique; (b) we populated a web service ontology; (c) and we have used a benchmark collection for testing called OWLS-TC, in which we have demonstrated that our solution outperforms other approaches in terms of average resulting classification.

As future work, we plan to discover non-taxonomic relations or concepts [22] between web services and functionalities, input/output descriptions to enhance web service ontology in order to facilitate discovery and composition of web services.

In addition, semantic similarity between operations and between classified web services is relevant to extract for helping to software developers in web service discovery.

ACKNOWLEDGMENT

This work has been supported by *PRODEP* as part of project no. UAM-PTC-478. Authors thank the *SNI-CONACyT*.

REFERENCES

- I. Bluemke, M. Kurek and M. Purwin, "Tool for Automatic Testing of Web Services", Proc. of the 2014 Federated Conference on Computer Science and Information Systems, Warsaw, 2014, pp. 1553-1558. doi: 10.15439/2014F93
- [2] F. Liu, Y. Shi, J. Yu, T. Wang and J. Wu, "Measuring similarity of web services based on WSDL," in *Proc. of the 2010 IEEE International Conference on Web Services*, Florida, USA, 2010, pp. 155-162.
 [3] M. Bravo and M. Alvarado, "Similarity measures for substituting Web
- [3] M. Bravo and M. Alvarado, "Similarity measures for substituting Web services," Web Service Composition and New Frameworks in Designing Semantics: Innovations, pp. 143-170, 2012.
- [4] J. Zhang and D. Pan, "Web Service Classification," Dan Pan, Jing Zhang [EB/OL], 2008.
- [5] H. Wang, Y. Shi, X. Zhou, Q. Zhou, S. Shao and A. Bouguettaya, "Web service classification using support Vector Machine," in *Proc. of the 22nd IEEE International Conference on Tools with Artificial Intelligence*, Arras, France, 2010, pp. 3-6.
- [6] L. Chen, Y. Zhang, Z. L. Song and Z. Miao, "Automatic web services classification based on rough set theory," Journal of Central South University, vol. 20, pp. 2708-2714, 2013.
- [7] R. Mohanty, V. Ravi and M. R. Patra, "Web-services classification using intelligent techniques," Expert Systems with Applications, vol. 37(7), pp. 5484-5490, 2010.
- [8] L. Yuan-jie and C. Jian, "Web service classification based on automatic semantic annotation and ensemble learning," in *Proc. of the 26th International on Parallel and Distributed Processing Symposium Workshops* & *PhD Forum*, Shanghai, China, 2012, pp. 2274-2279.
- [9] R. Nisa and U. Qamar, "A text mining based approach for web service classification," Information Systems and e-Business Management, pp. 1-18, 2014.
- [10] J. Wu, L. Chen, Z. Zheng, M. R. Lyu and Z. Wu, "Clustering web services to facilitate service discovery," Knowledge and information systems, vol. 38(1), pp. 207-229, 2014.
- [11] K. Elgazzar, A. E. Hassan and P. Martin, "Clustering WSDL documents to bootstrap the discovery of web services," in *IEEE International Conference on Web Services (ICWS)*, Florida, USA, 2010, pp. 147-154.
- [12] Q. Liang, P. Li, P. C. Hung and X. Wu, "Clustering web services for automatic categorization," in *IEEE International Conference on Services Computing*, Bangalore, India, 2009, pp. 380-387.
- [13] H. S. Nguyen, S. H. Nguyen and W. Swieboda, "Semantic explorative evaluation of document clustering algorithms," *Proc. of the 2013 Federated Conference on Computer Science and Information Systems*, Krakow, 2013, pp. 115-122.
- [14] M. Bravo, J. Rodríguez and A. Reyes, "Enriching Semantically Web Service Descriptions," in On the Move to Meaningful Internet Systems: OTM Conferences, Amantea, Italy, 2014, pp. 776-783.
- OTM Conferences, Amantea, Italy, 2014, pp. 776-783.
 [15] Y. J. Lee and C. S. Kim, "A learning ontology method for restful semantic web services," in *IEEE International Conference on Web Services* (*ICWS*), Washington DC, USA, 2011, pp. 251-258.
- [16] M. Sabou, "Learning web service ontologies: an automatic extraction method and its evaluation," Ontology learning from text: methods, evaluation and applications, vol. 123, pp. 125-139, 2005.
- [17] M. Bravo, J. Rodríguez and J. Pascual, "SDWS: Semantic Description of Web Services," International Journal of Web Services Research, vol. 11(2), pp. 1-23, 2014.
- [18] M. Horridge and P. F. Patel-Schneider, "Manchester syntax for OWL 1.1," OWL: Experiences and Directions, Washington, USA, 2008.
- [19] R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
- [20] S. R. Garner, "Weka: The Waikato environment for knowledge analysis," in Proc. of the New Zealand Computer Science Research Students Conference, 1995, pp. 57-64.
- [21] P. Buitelaar and P. Cimiano, "Ontology learning and population: bridging the gap between text and knowledge," vol. 167, IOS Press, 2008.
- [22] P. Szwed, "Concepts extraction from unstructured Polish texts: A rule based approach," Proc. of the 2015 Federated Conference on Computer Science and Information Systems, Lodz, 2015, pp. 355-364. doi: 10.15439/2015F280