ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 4, April 2013

Ontology Evolution by Significant Analysis of Terms using WordNet for Web Services

Priya C V, Janaki Kumar

***Abstract*— Ontologies have become an effective modelling tool. Several applications make use of them and significantly in the semantic web. However, construction of ontology is an intimidating job. Ontology evolution is one of the latest methods for the construction of ontology which focus on automatic generation of the concepts and their relations in a specified area. The difficulty with numerous, largely dissimilar concepts must be tackled by the evolution of ontology which depends on the collection of already defined textual sources like web services. The ontology evolution process for web services by utilizing the WSDL and free text descriptors is proposed. Term Frequency/Inverse Document Frequency (TF/IDF) and web context generation are the two techniques used to assess the WSDL descriptor. The proposed ontology evolution process improves the TF/IDF ranking by introducing WordNet tool to produce more significant tokens. The outcomes of these techniques are combined to obtain significant concepts. The service free text descriptor is the third technique applied to validate the concepts generated. Thus the combined ontology evolution approach contributes to a furthermore accurate ontology definition.**

***Index Terms*—Ontology evolution, Web services**

I. INTRODUCTION

A variety of applications makes use of ontology, especially the semantic web and it is mostly opted as a modelling tool. But the difficulty lies in the designing and maintenance of ontologies [2], [5]. Ontology evolution is a significant technology for the construction of ontology that has recently emerged. It includes automatic recognition of concepts significant to a specified area and the relations between the concepts [6].

Either a restricted area [8] or expanding the present ontology [9] was considered in the earlier work of ontology evolution. Considering web services, Universal Description, Discovery and Integration (UDDI) registries were created for storing information about web services and to support interoperability. But there are some disadvantages for the UDDI registries [7]. Specifically, these registries either are available in public or have several outdated entries or to restrict the access, it needs registration. In any of these situations, a registry holds only a restricted amount of description of the available services. This problem can be addressed by creating ontologies for classifying and making use of web services. But, when the amount of available web services rise, the classification of web services become a tedious task with single domain ontology or a collection of already defined ontologies for other purposes. In addition, when the amount of web services rise invariably, continuous manual work is needed to construct ontology.

A web service can be divided into two types of descriptions 1) Web Service Description Language (WSDL) describing "how" the web service should be used and 2) the free text with a textual description of the web service describing "what" the service does. The ontology evolution process proposed in this paper makes use of these two descriptions. Thus, ontology is constructed base on WSDL and the free text descriptor is used to validate the process.

Three different techniques are used to analyse a web service in the ontology evolution process. Each technique represents a different point of view of the web service. Thus, the ontology is defined more accurately by this process to produce better outcome. The Term Frequency/Inverse Document Frequency assess the web service from an internal view point, i.e., what concept in the text can describe the WSDL document content in the best way. By introducing WordNet tool, TF/IDF ranking can be improved. It generates a more significant set of tokens and thus support in obtaining a set of significant concepts. The external view point of the WSDL document is described by the Web Context Extraction technique, i.e., based on the WSDL content, what is the most common concept that describes the answers to the web search queries. In the end, to solve the inconsistencies with the current ontology, the Free Text Description Verification technique is used. When the three techniques agree with the recognition of the new significant concept or a change in relation between the ontology concepts, an ontology evolution takes place. The descriptors related to both concepts define the relation between the two concepts. The maintenance work can be reduced considerably and the evolution of ontology is supported by our process. Our process provides a way for automatic construction of ontology that helps in classifying, expanding and retrieving significant services.

Manuscript received April, 2013.

***Priya C V**, PG Scholar, Computer Science and Engineering, Coimbatore Institute of Engineering and Technology., . Narasipuram, Coimbatore, Tamil Nadu, ,India, 9895319376*

***Janaki Kumar**, Professor, Computer Science and Engineering, Coimbatore Institute of Engineering and Technology, Narasipuram, Coimbatore, Tamil Nadu,,India, 9894049094.*

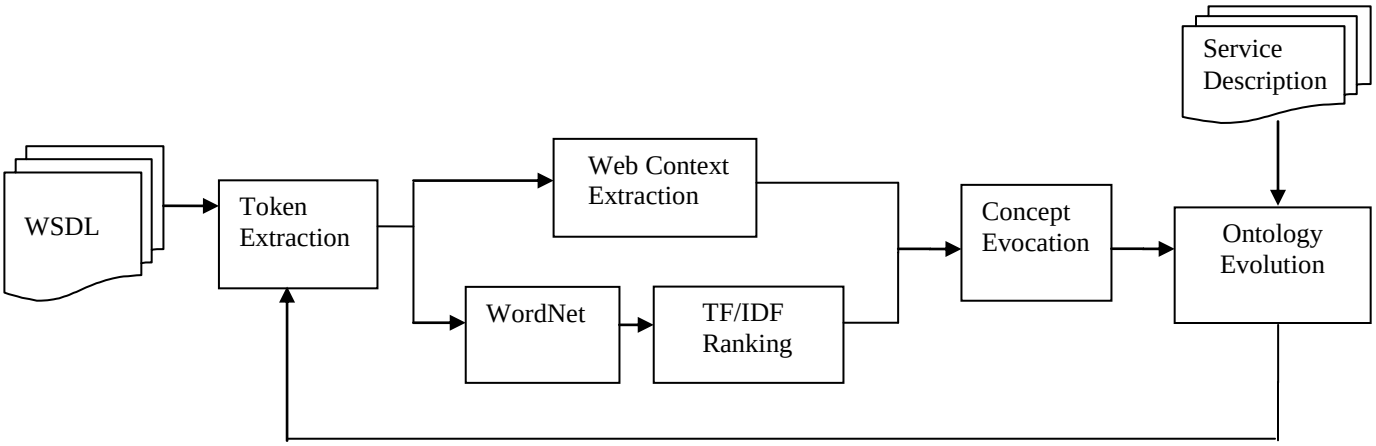

Fig. 1 Web Service Ontology Evolution Process

II. THE ONTOLOGY EVOLUTION PROCESS

The ontology evolution process proposed in this paper focus on continuous assessment of WSDL documents and provides a n ontology model depending on the concepts and their relationships. The improvement of the proposed ontology evolution process focus on 1) the integration of the two extraction techniques, TF/IDF and the Web Context Extraction, 2) improvement of TF/IDF ranking by the use of WordNet tool and 3) assessing the external service descriptor for the result verification using Free Text Descriptor Verification technique.

A. An overview of the evolution process

The ontology evolution process is explained in Fig. 1. The process includes four steps. The first step is the token extraction where the tokens representing the relevant information are extracted from the WSDL document. Here, all the name labels are extracted, the tokens are parsed and initial filtering is performed.

The second step assesses the extracted WSDL tokens using two techniques in parallel. The WordNet tool used to improve the TF/IDF ranking determines the tokens with same meaning. The TF/IDF assess the terms appearing the most in the web service document and those appearing less frequently in the other documents. In Web Context Extraction, the query to the search engine is the set of tokens extracted, according to the textual descriptors the results are clustered and the context of the web service is identified by classifying the set of descriptors.

The third step is Concept Evocation where the descriptors appearing in both the TF/IDF and Web Context Extraction techniques are identified. The ontology evolution can make use of the possible concept names that are determined from these descriptors. The process of convergence of the relations between the concepts is supported by the context descriptors.

The last step is the ontology evolution where the newly determined concepts help in the expansion of the ontology and the modification of the relations between them. The conflict between the new concept and the current ontology is solved by the external web service textual descriptor. These conflicts arise when there is a need to define the concept more specifically or to define the relation between the concepts. Correct understanding of the new concepts can be verified against the free text descriptor. The most common context descriptors between the concepts define the relations. Once ontology is evolved, the whole process moves to the next WSDL with the evolved ontology concepts and relations. The processing order of the WSDL documents is considered arbitrary.

The process is illustrated using the following three web services:

DomainSpy – A web service that identifies the domain registrants by region or registrant name.

AcademicVerifier – A web service that determines whether a domain name or an email address belong to an academic institution.

ZipCodeResolver – A web service that resolves partial US mailing address and returns proper zip code.

B. Token Extraction

The first step in the ontology evolution is token extraction where the *descriptor* represents the set of tokens for a web service S. The tokens extracted from the WSDL document are textual terms. The WSDL document is represented by the descriptor, formally put as $D^{S}_{wsdl} = \{t_1, t_{2, \dots}, t_n\}$, where t_i is a token. Sequence of tokens combined together form meaningful tokens where the first letter of the words is capitalized (GetDomainsByZipSoapIn). Fig. 2 represents a WSDL document of the DomainSpy web service. The token list is bolded. The extracted tokens list serves as the *baseline*. Here, the tokens are extracted from the WSDL of DomainSpy.

```
<s:element name="GetDomainsByRegistrantName">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="FirstMiddleName" type="s:string"/>
<s:element minOccurs="0" maxOccurs="1" name="LastName" type="s:string"/>
<s:element minOccurs="0" maxOccurs="1" name="LicenseKey" type="s:string"/>
</s:sequence>
</s:complexType>
</s:element>
```

Fig. 2 Example of DomainSpy Webservice

The tokens that are less significant are extracted from the elements classified under names. List of stop words are used to remove the tokens with no considerable meaning and thus filtering is performed.

C. Applying WordNet

WordNet is a lexical database consisting of the English language. The words are separated into the parts of speech such as adverbs ,verbs, noun, adjectives, They are linked using semantic relationships. The basic building block of WordNet is a synset. A synset is a collection of synonymous words i.e; set of synonyms.

The improvement in TF/IDF ranking is done by using WordNet tool. The tokens with same meaning are determined by the use of WordNet. The similar set of tokens is grouped. From this set, the token with highest frequency is determined and replace the other tokens in the group wit that token. Change the frequency count of the token by adding the frequency of the replaced tokens. By performing this, the tokens that are most significant are obtained, removing the irrelevant ones.

D. TF/IDF Ranking

The TF/IDF analysis is applied to the WSDL descriptors. $freq(t_i, D_i)$ is defined as the number of times the token t_i occur within the document descriptor D_i. The term frequency of each token t_i is defined as,

$$tf(t_i) = \frac{freq(t_i, D_i)}{|D_i|} \qquad (1)$$

D_{wsdl} is defined as the corpus of WSDL descriptors. The inverse document frequency is determined as the ratio between the total number of documents and the number of documents that contain the term

$$idf(t_i) = \log \frac{|D|}{|\{D_i : t_i \varepsilon D_i\}|} \qquad (2)$$

D is defined as the specific WSDL descriptor. The TF/IDF weight of the token is represented as $w(t_i)$, is calculated as

$$w(t_i) = tf(t_i) * idf^2(t_i) \qquad (3)$$

Using the token weight the descriptor tokens can be ranked. The filtering of the tokens is performed by ranking using a threshold. The tokens with weight less than the standard deviation are filtered out. The standard deviation is obtained from the average weight of token w value. Fig. 3 on the left circle represent set of tokens that has a higher weight than the threshold. After the filtering process, few tokens in the baseline list (Fig. 2) were removed. For example, the tokens "Get", "Result" with TF/IDF weight less than the threshold were filtered out.

E. Web Context Extraction

To identify a record of information such as web service, a context descriptor c_i is defined from domain DOM as an index term. It comprises of a word, phrase, or alphanumerical term. The importance of descriptor c_i in relation to the web service is determined by the weight w Є R. A descriptor set $\{<c_i, w_i>\}_i$ is defined by a set of pairs, descriptors and weights. A different view point is defined by each descriptor of the concept. The descriptor set ultimately defines all the different perspectives and their significant weights which define the importance of each perspective.

The context is obtained by the collection of different point of views defined by the different descriptors. The finite set of descriptors define the context $C = \{\{<c_{ij}, w_{ij}>\}_i\}_j$, where i represents each context descriptor and j represents the index of each set. For example, a context C may be a set of words defining a web service and the weights can represent the significance of a descriptor to the web service.

The context recognition algorithm is adapted from [10]. It consists of the following major phases: 1) selecting contexts for each set of tokens, 2) ranking the contexts, and 3) declaring the current contexts.

The tokens extracted from the web service WSDL descriptor is the input to the algorithm. From elements classified as name, each set of tokens is sent as query to a web search engine and by clustering the web page search results into possible contexts.

The clustering of the web pages is based on concise all pairs profiling (CAPP) clustering method [1]. This method is for concise, intelligible and approximate profiling of large classifications. The minimization of total number of features necessary to assure that each pair of classes is contrasted by at least one feature is done by the comparison of all classes pair wise. Then, minimized list of features is assigned to its own class profile, which is characterized by how these features differentiate the class from the other features.
The ranking stage result will be the current context or a setoff highest ranking contexts. The set of preliminary contexts that has the top number of references, both in number of Web pages and in number of appearances in all the texts, is declared to be the current context and the weight is defined by integrating the value of references and appearances.

F. Concept Evocation

Concept evocation determines a possible concept definition that can be refined further in the ontology construction. The concept evocation is performed based on context intersection. The descriptors that appear in the intersection of both the TF/IDF outcome and web context outcome are defined as the ontology concept. tf/idf_{result} is defined as one descriptor set from the TF/IDF results, depending on extracted tokens from descriptor set extracted from the web and representing the same document. As a result, the ontology concept is represented by a set of descriptors, c_i, which belong to both sets,

$$Concept = \{c_1, c_2, \ldots, c_n \mid c_i \in tf/idf_{result} \cap c_i \in C\} \quad (4)$$

Fig. 3 represents an example of the concept evocation method. The two overlapping circles describe each web services. TF/IDF results are represented in the left circle and the web context results in the right circle. The intersection represented in the overlap between both methods defines the possible concept. For the DomainSpy Web service, the concepts based on the intersection of both descriptor sets are identified as location, city, name, zip. The context is expanded to include the descriptors identified by the TF/IDF, the web context and the concept descriptors. The expanded context, $Context_e$, is represented as the following,

$$Context_e = \{c_1, c_2, \ldots, c_n \mid c_i \in tf/idf_{result} \cup c_i \in C\} \quad (5)$$

The relation between two concepts, Con_i and Con_j can be defined as the context descriptors common to both concepts, for which weight w_k is greater than a cutoff value of a,

$$Re(Con_i, Con_j) = \{c_k \mid c_k \in Con_i \cap Con_j, w_k > a\} \quad (6)$$

However, since multiple context descriptors can belong to two concepts, the cutoff value of a for the relevant the WSDL text. The context, C, is initially defined as a descriptors needs to be predetermined. A possible cutoff can be defined by TF/IDF, Web Context, or both. Alternatively, the cutoff can be defined by a minimum number or percent of web services belonging to both concepts based on shared context descriptors.

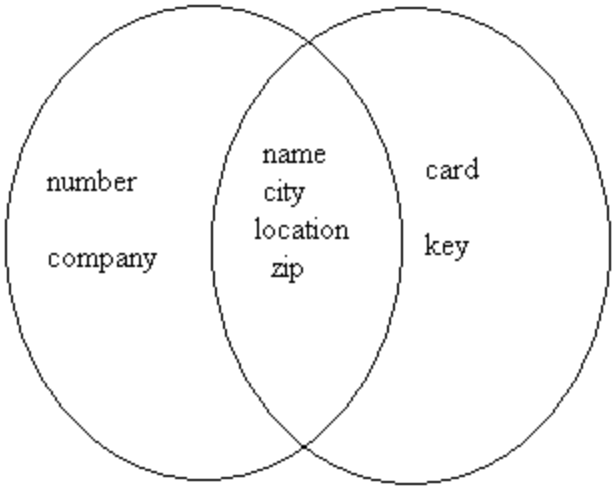

Fig. 3 Example of Concept Evocation for DomainSpy

G. Ontology Evolution

On refining the possible identified concepts, new concept can be build. The evocation of a concept in the previous step does not guarantee that it should be integrated with the current ontology. Instead, the new possible concept should be analyzed in relation to the current ontology.

Validation of the descriptor is done using the textual service descriptor. The second descriptor, $D^S_{desc} = \{t_1, t_2, \ldots, t_n\}$, represents the textual description of the service supplied by the service developer in free text. These descriptions are comparatively short and consist of few sentences describing the web service. The verification process includes matching the concept descriptors in simple string matching against all the descriptors of the service textual descriptor. A simple string-matching function, $match_{str}$, returns 1 if two strings match and 0 otherwise. For the DomainSpy web service, the concept generated passes the textual description verification.

III. EXPERIMENTAL RESULTS

The data for the experiments were the web services. Each web service had a WSDL document and a short textual description.

A. Ontology Evolution without using WordNet

The token extraction step extracts the name labels from the WSDL document. These tokens are then passed on to the TF/IDF ranking step. The TF/IDF assess the terms appearing the most in the web service document and those

appearing less frequently in the other documents. A large number of terms are obtained after ranking through TF/IDF and these terms are not significant. Hence it is difficult to evoke better concepts. Since, significant concepts are not evoked, the ontology evolved is not significant. Fig 4 represents the ontology evolved without using WordNet tool. Since a large number of tokens were obtained through TF/IDF ranking, only a significant concept *domain* is obtained.

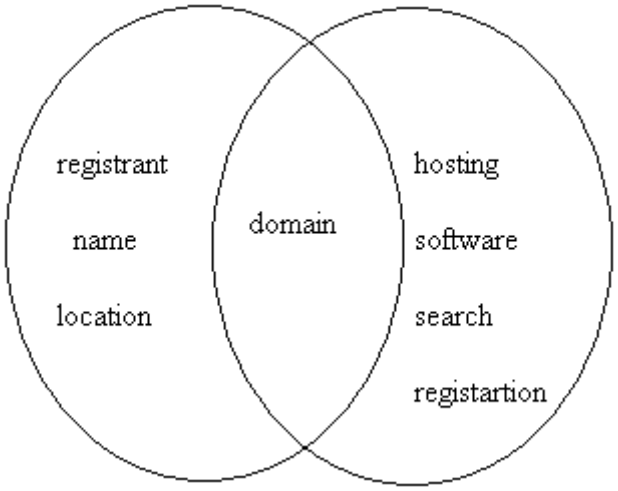

Fig. 4 Example of Ontology Evolution without WordNet

B. *Ontology Evolution using WordNet*

The tokens extracted at the token extraction step from the WSDL document are passed to the WordNet tool. The improvement in TF/IDF ranking is done by using WordNet tool. The tokens with same meaning are determined by the use of WordNet. The similar set of tokens is grouped. From this set, the token with highest frequency is determined and replace the other tokens in the group wit that token. Change the frequency count of the token by adding the frequency of the replaced tokens. By performing this, the tokens that are most significant are obtained, removing the irrelevant ones. The tokens are then passed to the TF/IDF ranking step. Since the tokens are obtained from WordNet tool, the TF/IDF ranking is easier due to the less number of relevant tokens. Hence it is easy to evoke better concepts. Since, significant concepts are evoked, the ontology evolved is significant. Fig 5 represents the ontology evolved using WordNet tool. Since only a few number of tokens were obtained through TF/IDF ranking, only significant concepts such as name, location, key and zip are obtained.

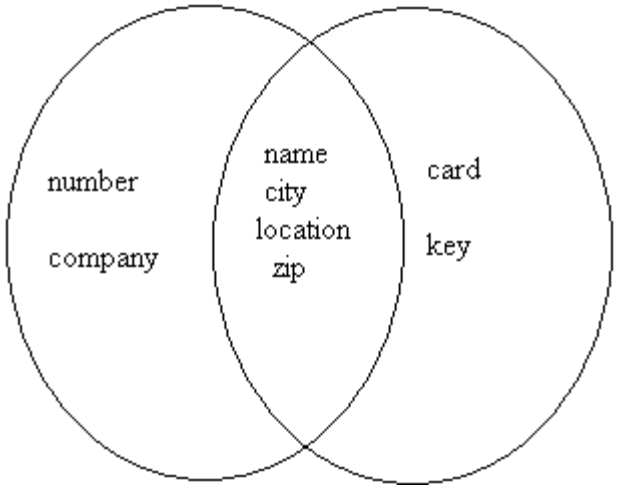

Fig. 5 Example of Ontology Evolution using WordNet

IV. CONCLUSION

A process of ontology evolution is proposed in this paper based on web service descriptions. This process analyzes web services from multiple view points and combines the results. The WSDL and free text descriptors of the web services provides an added advantage for the evolution of ontology. This ontology evolution depends on the WSDL and the process is verified using the free text descriptors. The improvement in TF/IDF ranking by including the WordNet tool helps in identifying the significant tokens. The integration of the results helps to generate more significant concepts. By analysis of the union and intersection of the concept results the value of the concept relations is obtained. The process provides a way for automatic construction of ontology that helps in classifying, expanding and retrieving significant services. Thus, the effort required in building and the maintaining the ontology is substantially reduced. As more significant concepts are generated by the use of WordNet tool, it helps in improving the overall performance in evolution of ontology.

REFERENCES

[1] R.E. Valdes - Perez and F. Pereira, "Concise, Intelligible, and Approximate Profiling of Multiple Classes," Int'l J. Human-Computer Studies, pp. 411-436, 2000.

[2] N.F. Noy and M. Klein, "Ontology Evolution: Not the Same as Schema Evolution," *Knowledge and Information Systems*, vol. 6, no. 4, pp. 428-440, 2004.

[3] L. Ding, T. Finin, A. Joshi, R. Pan, R. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs, "Swoogle: A Search and Metadata Engine for the Semantic Web," Proc. 13th ACM Conf. Information and Knowledge Management (CIKM '04), 2004.

[4] S. Robertson, "Understanding Inverse Document Frequency: On Theoretical Arguments for IDF," *J. Documentation*, vol. 60, no. 5, pp.503-520, 2004.

[5] D. Kim, S. Lee, J. Shim, J. Chun, Z. Lee, and H. Park, "Practical Ontology Systems for Enterprise Application," *Proc. 10th Asian Computing Science Conf. (ASIAN '05)*, 2005.
[6] M. Ehrig, S. Staab, and Y. Sure, "Bootstrapping Ontology Alignment Methods with APFEL," *Proc. Fourth Int'l Semantic Web Conf. (ISWC '05)*, 2005.
[7] C. Platzer and S. Dustdar, "A Vector Space Search Engine for Web Services," *Proc. Third European Conf. Web Services(ECOWS '05)*, 2005.
[8] G. Zhang, A. Troy, and K. Bourgoin, "Bootstrapping Ontology Learning for Information Retrieval Using Formal Concept Analysis and Information Anchors," *Proc. 14th Int'l Conf. Conceptual Structures (ICCS '06)*, 2006.
[9] S. Castano, S. Espinosa, A. Ferrara, V. Karkaletsis, A. Kaya, S. Melzer, R. Moller, S Montanelli, and G. Petasis, "Ontology Dynamics with Multimedia Information: The BOEMIE Evolution Methodology," *Proc. Int'l Workshop Ontology Dynamics(IWOD'07), held with the Fourth European Semantic Web Conf.(ESWC'07)*,2007.
[10] A.Segev, M. Leshno, and M. Zviran, "Context Recognition Using Internet as a Knowledge Base," *J. Intelligent Information Systems*, vol. 29, no. 3, pp. 305-327, 2007.
[11] Y. Chabeb, S. Tata, and D. Belad, "Toward an Integrated Ontology for Web Services," Proc. Fourth Int'l Conf. Internet and Web Applications and Services (ICIW '09), 2009.
[12] V. Mascardi, A. Locoro, and P. Rosso, "Automatic Ontology Matching via Upper Ontologies: A Systematic Evaluation," IEEE Trans. Knowledge and Data Eng., doi:10.1109/TKDE.2009.154, 2009.
[13] Aviv Segev and Quan Z.Sheng, "Bootstrapping Ontologies for Web Services," *IEEE Transactions on Service Computing*, vol 5, No 1, January-March 2012.

Priya C V is currently pursuing M.E Computer Science and Engineering at Coimbatore Institute of Engineering and Technology, Coimbatore, Tamil Nadu, (Anna University, Chennai). She completed her B.E in Computer Science and Engineering from Hindusthan Institute of Technology, Coimbatore, Tamil Nadu, (Anna University, Coimbatore) in 2011. Her research interests include Ontology and Web Services

Janaki Kumar is currently Professor in the Department of Computer Science, at Coimbatore Institute of Engineering and Technology, Coimbatore, Tamil Nadu, (Anna University, Chennai). She completed her B.E (Hons.) in Electronics and Communication Engineering from the University of Madras in 1978 and M.E in Computer Science and Engineering from the University of Madras in 1980. She has about 20 years experience in industry and 13 years experience in teaching. Her research interests include Testing, SOA and Web Services.