Clustering of Bootstrap for Web Service Discovery

S. Sagayaraj[1], M.Santhoshkumar[2]
Sacred Heart College, Tirupattur, India
sagisara@gmail.com[1], sannatcs@gmail.com[2]

Abstract—Web services are accessed using URLs in a distributed environment. WS WSDL document URLs are manually tabulated and clustered which increases the cost and timing for the developer. This paper introduces a new Clustering of URLs (CU) framework for clustering of bootstrap for web service discovery and clustering them in various domains using transfer, filter, spell check and domain set methods. These methods set them under the specific domain or general category. The CU framework is implemented with a sample URLs. The result shows the efficiency of the clustering of WSDL URLs.

Keywords-*Clustering URLs; CU Techniques;Clustering time; Domain counts*

*****

I. INTRODUCTION

Nowadays the software developer collects the services without actually writing the code from the heterogeneous environment.Service is a predefined software components with interface are accessed through Uniform Resource Locator (URL). In Service-oriented Architecture (SOA), service is defined by aWeb Service Description Language (WSDL) that supports interoperable operations between different software applications using existing web standard messaging protocol through the internet [1-2]. Web services are common application of SOA. Service OrientedComputing (SOC) is anassuming model that is determined by SOA. SOC uses services as the basic concepts to support quick, low-cost, and easy configuration of distributed applications even in mixed environments [3].Web Services needs less time for developing infrastructure, and computing environment. Web services have four components [4] such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), Universal Description Discovery and Integration (UDDI) and eXtensible Markup Language (XML). Web service is labeled by description of WSDL document using XML that asset endpoints operations and messages containing either document-oriented or procedure-oriented information [5]. Structure of WSDL document describes the non-semantic Web services and the operations that can be performed by certain web service as well as its location.

WSDL document describes five components such as,

- *Types*– XML type that pronounces the data containers used in message exchanges[6-7]
- *Messages*–theoretical representation of the transmitted information contains one or more logical parts. These parts are associated with a type definition [8-10]
- *PortType*–defines *a* set of abstract operations that can be performed by the Web service and each operation is associated with an input and/or output message [11-13]
- *Binding* – specifies the communication protocol and data format for each operation and message defined in a particular PortType element [14-16]
- *Service*–composite operation that aggregates multiple related Ports or functions.

The user understands the WS operations and descriptions [17-22] through the WSDL URLs which takes much time. Hence WSDL document URLs is considered as an invisible sixth component of WSDL document. The web services developed interfaces with description of text and that is published in UDDI registry or Service Search Engine (SSE). The WS key providers are Amazon, Yahoo and Google. They published their WS through their own websites and also use public registries or brokers. These facilities are utilized for project development. The following are the major issues in the current web service discovery

- User search the WS using syntactic matching in SSE or UDDI. The result is un-clustered WS which is manually clustered. For instance, searching "*wether*" in WS may result in a mismatch; when there is no WS with this termor part of vocabulary.
- The WS WSDL document URLs are clustered manually that raises the complexity, mismatch and errors in spelling and meaning in the web services discovery.
- User-intensive clustering WSDL document URLs under specific domain takes much time and cost is better for small number of URLs. The scalability of WS creates complexity in the clustering.

The above issues motivated clustering of WSDL document URLs using clustering method will solve the bottleneck of mismatch in accordance with the keyword. The manual tabulation is converted to clustering under the specific domain. This clustering can help in software development to search and fix the specific services.

The rest of the paper is organized as follows. The earlier research work and background technology is discussed in section 2 and the framework for clustering the WSDL URLs is elaborated in section 3. Implementation and description of the framework is briefed in section 4 and the result analysis is explained in section 5. Conclusion and future work are provided in section 6.

II. RELATED WORK

Bootstrap is the origin of web service and represented as URL. Bootstrap services discovery and its usability have grown significantly in WS clustering. The existing works and various

technology backgrounds used in the bootstrap service discovery are summarized in this section.This section provides an overview ofbootstrap service discovery proposed by following authors.

Khalid Elgazzar*et al*[23]developed the Clustering WSDL Documents to Bootstrap the Discovery of Web Servicesin 2010. They proposed a novel technique to mine WSDL documents and cluster them into functionally similar Web service groups. The application proposed concept for clustering Web services basedon function similarity using Word Vector Tool, as a predecessor step to retrieve the relevant Web services for a user request by search engines.The Word Vector Tool is a simple but flexible Java library to create word vector representations of text documents. Word vectors was used for various text processing tasks such as text classification and manual clustering of information retrieval of terms from WS WSDL documents URL[24].

Tingting Liang *et al* explored the Co-clustering WSDL Documents to Bootstrap Service Discovery in 2014 and explains manual clustering of WS. They proposed a novel approachnamed WCCluster to simultaneouslycluster WSDLdocuments andthe words extracted from them to improve the accuracy of clustering. It poses co-clustering as a bipartite graph partitioning problem, and uses a spectral graph algorithm in which proper singular vectors are utilized as a real relaxation to the Natural Process (NP)-complete graph partitioning problem.WCCluster induces WSDL documents word clustering and it induces word clustering. WCCluster is easy to verify the best word or portion of a word in a specific domain using manual clustering. This take more time and cost which are avoid through the clustering [25].

Jian Wu *et al*explained the web services discovery using noisy tags, and utilizing tag mining by tag recommendation. They selected experimental data as using real datasets and web service recall 14% in most cases in discovery[26].

Qianhui Liang et al discussed the clustering web services for automatic categorization. The unclassified web services is compare with classified web service using latent inter-relatioships among the individual factor [27].

III. FRAMEWORK FOR CLUSTERING WSDL URLS

This section proposes a new framework called Clustering of URLs (CU) framework is set the specific domain for WSDL URLs . This framework split up the WSDL document URLs from non WSDL document URLs and stores in different groups. It uses Web Ontology Language – Test Collection (OWL-TC) model which contains collection of 1076 Web Service WSDL document. The CU framework incorporates a method for web services clustering.

OWL-TC model runs the web services in the host through Xamp server. It has plenty of WSDL document URLs. The large number of .WSDL document makes the process harder for web service discovery. The CU framework collects .WSDL extension URLs from Non-WSDL document URLs.It stores the coefficient of the corresponding index value, which counts the number of WSDL URLs in a specific vector sampling. WSDL document URLs have plenty of meaningful words. Transfer and Filter methods extracts the meaningful words or error words from WSDL document URLs. Transfer deletes the URLs words, natural language stop words and special characters from the list. Filter removes the file name extensions and numbers. The Spell-check method corrects miss-spelled words by matching WorldNet and sets the domain for WSDL URLs; otherwise stays in general category as shown Figure 1.

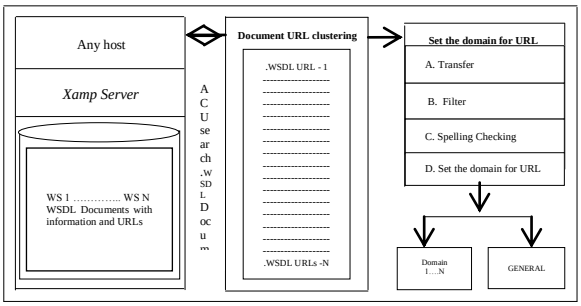

Figure 1 – CU Framework

Algorithm - 1
Input : URLs from any host
Ouput : To set the domain forWSDL_URLs

```
1:Procedure Automatic_Domain_Setting(urls)
2:int j=1;
3: for i←1 to m
4:  x(i) = urls
5:If x(i)=[ .wsdl| .WSDL] then
6:      y(j) = x(i);
7:        T and F methods applied on y(j) and extractthe words(F)
8:        If F = D then
9:            y(j)←[sd1, sd2,…, sdn] → D
10:           else
11:               y(j) Stay in the G
12:           endif
13:  end if
14:  j++;
15: repeat
16:End
```

i. *Transfer* - applied to n number of WSDL document URL and the words are found and ordered. Stop words(T_1) and Special characters(T_2)are eliminated from WSDL document URLs.

$T_1 = \sum_{i=1}^{A}$ [stop words Ai such that CAT(Ai)]

$T_2 = \sum_{j=1}^{B}$ [special characters words Bj such that CAT(Bj)]

CAT- Categories.

$$\text{Transfer (T)} = (y(j)) - T1 - T2 \ldots (1)$$

ii. *Filter* - removes numbers (F_1) and secondary file names(F_2) provided by the transfer method using equation 2 from the URLs.

$F_2 = \sum_{k=1}^{C}$ [File Extension Words$C_k$such that *CAT* (C_k)]

$F_1 = \sum_{l=1}^{D}$ [Numbers D_l such that *CAT* (D_l)]

$$F = (T) - F1 - F2 \ldots (2)$$

Spell check and Domain Setting - Filter words are erroneous is corrected by the spell-checker and stored in the $\boldsymbol{w_{sd}}$ vector; otherwise the unmatched terms are removed from the filtered list.The $\boldsymbol{w_{sd}}$ is matched with WordNet term. This term considered as subdomain $\boldsymbol{sd}$. Thus, the vector $\boldsymbol{w_{sd}}$ will be replaced by domain vector term $\boldsymbol{d_{sd}}$ where

$$d_{sd} = \left(df(sd, d_1), \ldots, df(sd, d_z)\right) \ldots(3)$$

The domain vector with $z = |D|$ and *df (sd,d)* denotes the frequency. Sub domain *sd* appears in adomain d ∈ D. All the subdomains enhances the domain. The domain frequency is follows:

$$df\,(sd\,,d\,) = df\,\{sd\,,\{d \in D \mid d \in ref\,(\,sd\,)\}\} \ldots(4)$$

The sub domain *sd* is set the domain D for WSDL URLs, otherwise it stays in general category. The CU framework aims to improve the execution efficiency by the following two ways:

a. Elimination of redundancy in URL words.
b. Extracted URLs words are rearranged or rewritten. It will help to set the domain name for specific web service that will provide efficient web service discovery.

It is evident that CU should not change the meaning of WSDL document URLs.

IV. IMPLEMENTATION

This section discusses the application of CU framework. developed using C#. It loads the web service WSDL document URLS from different types of URLs. Transfer method separates the words and removes the stop words and special characters from URLs. For instance the transfer method applied to the following URL http:\\127.0.1.1\wsdl\1personbicycle4wheeledcar_priec_service.wsdl extracted 7 words and Filter method removed the secondary file name and numbers as (personbicyclewheeledcar, priec) within 0.3987ms. The spell-checker corrects and eliminates the words and sets the domain. From the output of the two words, the first word is meaningless from the WordNet and the second word is corrected as price within 0.5837ms. The WSDL document URL has taken 0.9837ms for the entire process. Similarly the remaining URLs are processed and if the URLs are not placed in a domain will stay in the general category. The domain setting is shown in Figure 2. $tf(t_{sd},d_z)$ denotes the number of times subdomain t_{sd} occurs in domain d_z. $df(t_{sd})$ denotes the number of domain in which times t_{sd} occurs. |D| denotes the number of Domain.

standard *tfzdf* function, defined as:

$$tfzdf(t_{sd},d_z) = tf(t_{sd},d_z) * \log(|D|/df(t_{sd})) \ldots(5)$$

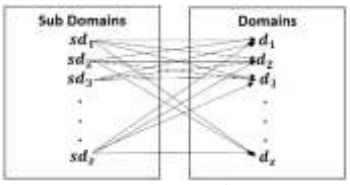

Figure 2 – Domain Setting for WSDL URLs

V. RESULT ANALYSIS

The advantage of clustering is realized in search and discovery of WS. The result of the framework is investigated through time based and count based analysis based on bootstrap sampling.Subdomains are gathered under the domain by the framework.The $\hat{X}$represents all the WSDL document URLs count and X_1, X_2, X_3, $X_{4,\ldots,}$ $X_z$are subdomains count.Bootstrap sampling is define as,

$$\hat{X} = (X_1+X_2+ X_3+ X_4+\ldots +Xz) \ldots(6)$$

Here,

$X_1 = \sum_{a=1}^{Z} x_a$; $X_2 = \sum_{b=1}^{Z} x_b$; $X_3 = \sum_{c=1}^{Z} x_c$; $X_4 = \sum_{d=1}^{Z} x_d$; $X_z = \sum_{z=1}^{Z} x_z$

The average computation time taken for clustered the WSDL URLs is calculated as follows

$$\hat{X}\,(AT_t) = \sum_{t=1}^{z} \quad ((X_1\,(AT_t\,)+X_2\,(AT_t\,) + X_3\,(AT_t\,) +\ldots+Xz\,(AT_{tZ}\,)) \ldots(7)$$

As a sample, the analysis has taken 1076 WSDL document URLs and clustered 835 URLs in an average time AT_tin *4.01 sec*as shown Table-3. Here, $\mathbf{R}_1$ = No of clustered relevant URLs and $\mathbf{R}_2$ = unclustered relevant URLs. Recall value (R_G) is calculated as follows

$$\mathbf{R}_G = \frac{\mathbf{R}_1}{\mathbf{R}_1+\mathbf{R}_2} \text{ X } 100 \ldots(8)$$

From the above anlysis the $\mathbf{R}_G$ value is 78% for $\mathbf{R}_1$ *and* $\mathbf{R}_2$.

Domains(*d*)	**Sub Domain(*sd*)**	**Sub Domain Cluster Counts (t_{sd})**	**Averaging computation time (AT_t)**
Economy (412)	Price	371	0.35 ms
	Trade	023	0.14 ms
	Car	018	0.18 ms
Health(082)	Care	010	0.12 ms
	Diagnostic	019	0.13 ms
	Diagnosis	001	0.13 ms
	Hospital	017	0.12 ms
	Health	001	0.34 ms
	Emergency	001	0.17 ms
	Patient	003	0.18 ms
	Medical	024	0.13 ms
	Check	006	0.14 ms
Travel(279)	Geo	083	0.12 ms
	Address	039	0.13 ms
	Zip	011	0.13 ms
	City	047	0.12 ms
	Country	034	0.12 ms
	Municipal	019	0.16 ms
	Town	006	0.12 ms
	National	013	0.12 ms
	Surfing	027	0.15 ms
Education(062)	Publication	028	0.12 ms
	Academic	021	0.13 ms
	Lecture	012	0.13 ms
			0.33 ms
1076-835 ($\mathbf{R}_1$) = 241 ($\mathbf{R}_2$)	-	-	= 4.01s / 26 =0.15423 ms

Table-3: Clustering computation time duration

After clustering of WS based on WSDL document URLs, the economic domain is 53%, travel is 31%, health is 9% and education is 7% as shown in Figure-6. The WS developers concentrates two third on travel and economic domains and one third of the WS development is on education and health domains.

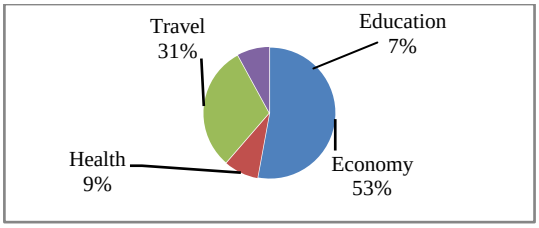

Figure 6 - Clustering the URLs based on percentage

The $\hat{\theta}$ turns calculate the ratio between domains and idendifying the confidence level between them in clustered sampling by using clustered order. The confidence level represent the WS development in particular domain. The$\hat{\theta}$is identify the confidence levels and calculate the ratio between the domains [28-30].

The clustered order as follow $DO_1,\ldots, DO_n$ using WSDL URLs domain count and the $\hat{\theta}$as follows

$$\hat{\theta} = \{\frac{DO_1}{DO_2}, \frac{DO_3}{DO_4}, \ldots, \frac{DO_{n-1}}{DO_n}\} \quad \ldots(9)$$

Here, *DO* is the Descending Order of the WSDL URLs for eachdomain value.

DO_1, DO_2, DO_3, DO_4 values 412, 279 and 82,62 are taken from Table-3 and apply on equation 9 as follows

$$(\text{confidence } 1)\hat{\theta} = \frac{412}{279} = 1.4768$$

$$(\text{confidence } 2)\hat{\theta} = \frac{82}{62} = 1.319$$

Interval for the domain ratio turns out to be$1.4768<1.319<\hat{\theta}_{(1)}$. This includes the neutral value $\hat{\theta} = 1$. Economic and travel domain ratios have provided significant development in web services and health and education web services are not concentrated by the developer. This proves that only business oriented web services motivated the web service industry. In this result analysis bootstrap sampling clustered URLs in specific domain and time taken. The Recall value is show the WSDL URLs reterived percentage. Calculated the confidence level $DO_1,\ldots,DO_n$ between domains by using clustered WSDL document URLs count and proves the siginificant development domains of WS.

VI. CONCLUSION

This paper introduced the clustering of WSDL document URLs. This framework applied CU Techniques on WSDL document URLs to remove unwanted words. The remaining words are set as valid domain name; otherwise stay in the general domain. The clustering process is implemented and the result analysis showed how it outperformed. Developers concentrated on the business oriented web services rather than the service oriented web services, because of the revenue to the web service industry. The research will continue further to convert or transform the clustered web services into semantic web services.

REFERENCES

[1] Thomas Erl, "Service-Oriented Architecture: A Field Guide to integrating XML and Web Services",Pearson Education,Publishing as Prentice Hall PTR, 2004.

[2] M. Parazoglou, P. Traverso, S. Dustdar, and F. Leymann, “Service oriented computing: State of the art and research challenges”, Computer, vol. 40, no. 11, pp. 38–45, 2007.

[3] H. Haas and A. Brown, “Web services glossary,” W3C Working Group Note (11 February 2004), 2004.

[4] J. Garofalakis, Y. Panagis, E. Sakkopoulos, and A. Tsakalidis, “Web service discovery mechanisms: Looking for a needle in a haystack,” in International Workshop on Web Engineering, vol. 38, 2004.

[5] E. Al-Masri and Q. H. Mahmoud, “Investigating web services on the world wide web,” in Proceedings of the 17th international conference on World Wide Web. ACM, 2008, pp. 795–804.

[6] Ethan Cerami, “Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL”, 1st Edition,O'Reilly & Associates, 2001.

[7] Ngoc-Khang Le and Ngoc-Khang LE, “An O (n (log n)^3 algorithm for maximum matching in tapezoid graphs”, RIVF international conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future(RIVF),IEEE,2013.

[8] Stephen Potts,"Sams Teach Yourself Web Services in 24 Hours”, Kindle edition,Que Publish, 2003.

[9] T. Chabard`es, P. Dokl´adal, M. Faessel, M. Bilodeau, “A PARALLEL, O (N) ALGORITHM FOR UNBIASED, THIN WATERSHED”, IEEE, 2013.

[10] Travis Garrett, “An O (N3) algorithm for the calculation of far field radiation patterns”, IEEE transactions on antennas and propagation, 2014.

[11] Alex Ferrar and Mathew MacDonald, "Programming .Net Web Services", 1st Edition, O'Reilly & Associates, 2002.

[12] Shiqi Li, Chi Xu, and Ming Xie, “A Robust OðnÞ Solution to the Perspective-n-Point Problem”, IEEE Transactions on pattern analysis and machine intelligence, 2012.

[13] Saad Omar and Dan Jiao, “O (N) Iterative and O(NlogN) Direct Volume Integral Equation Solvers for Large-Scale Electrodynamic Analysis”,IEEE,2013.

[14] James Snell, Doug TidWell and Pavel Kulchenko, "Programming Web Services with SOAP", 1st Edition, O'Reilly & Associates, 2002.

[15] Sweetesh Singh, Tarun Tiwari, Rupesh Srivastava and Suneeta Agarwal, “Back-Forth Sorting Algorithm Analysis and Applications Perspective”, IEEE, 2009.

[16] Ratthaslip Ranokphanuwat and Surin Kittitornkun, Sissades Tongsima, “Performance analysis & improvement of SNPHAP on Multi-core CPUs”, IEEE,2013

[17] Chi-Chia SunGene Eu Jan and Shao-Wei Leu, Kai-Chieh Yang, and Yi-Chun Chen, “Near-Shortest Path Planning on a Quadratic Surface With O(n log n) Time”, IEEE sensors journal, VOL. 15, NO. 11, november 2015.

[18] Banage T. G. S. Kumara , Incheon Paik and Hiroki Ohashi, Yuichi Yaguchi, “Web service filtering and visualization with context aware similarity to bootstrap clustering”, iCAST-UMEDIA, International Joint Conference on IEEE, 2013.

[19] Qi Yu, "Decision Tree Learning from Incomplete QoS to BootstrapService Recommendation”,19th International Conference on Web Services (ICWS), IEEE, 2012.

[20] LING LIU and M. TAMER OZSU, "Encylopeida of Database Systems: Bootsrap Sampling", PP 264-264,Springer,2009.

[21] Subbu Allamaraju, " RESTful Web Services Cookbook: Solutions for Improving Scalability and Simplicity", All Rights Reserve byYahoo, O'Reilly Media, 2010.

[22] Claude Sammut and Geoffrey I. Webb, "Encylopedia of Machine Learning: Bootstrap Sampling",Springer,2010.

[23] K. Elgazzar, A. E. Hassan, and P. Martin, “Clustering wsdl documents to bootstrap the discovery of web services,” in Web Services (ICWS), 2010 IEEE International Conference on. IEEE, 2010, pp. 147–154.

[24] Michael Wurst, "The Word Vector Tool - User Guide, Operator Reference, and Developer Tutorial", Publishing under the GNU License, 2006.

[25] Tingting Liang, Liang Chen, Haochao Ying, Jian Wu, "Co-clustering WSDL Documents to Bootstrap Service Discovery", IEEE 7th International Conference on Service-Oriented Computing and Applications, 2014, pp. 215-222.

[26] Jian Wu, " web services discovery using noisy tags, and utilizing tag mining by tag recommendation”, IEEE, 2016.

[27] Qianhui Liang, Peipei Li, Patrick C.K Hung, Xindong Wu, "Clustering webservices for automatic categotizazation", Service computing 2009,IEEE,2009.

[28] Denni D Boos and L A Stefanski, "Essential Statistical Inference : Bootstrap", Volume 120,PP 413-448,Series of Texts in Statistics,Springer,September,2012.

[29] Bryan Parno, Jonathan M. McCune and Adrian Perrig, "Bootstrapping Trust in Modern Computers",Volume 10, SpringerBriefs in Computer Science,2011.

[30] C.R Kothari, "Research Methodology –Methods and Techniques", New Age International (P) Limited, 2004.