WEB SERVICE DATA DISCOVERY BASED ON CONTEXTUAL FEATURE INDEXING AND TIMELY REFRESHING OF CRAWLED THREADS

A.Abdul Rahuman [1] & C. Chandrasekar[2]

ABSTRACT

Web data service discovery has become a new sign of internet technology improvement for end users. User behavior is usually sensitive to various contextual features which help in improving the semantic search performance significantly. However, contextual feature selection and extraction fails in providing result while discovering data in web services. The lack of precise user ratings on web as well as the sparse environment of data poses serious challenges to standard contextual feature extraction in terms of scalability and performance. The data discovery in the web does not create threads at a timely manner, and as a result, the user satisfaction degrades. This motivated us to address the problem by discovering the data based on the online leaning approach and include the feedbacks (i.e., ratings) from users. To improve the scalability rate of data discovery in web, Refresh Crawled Web based Indexing (RCWI) mechanism is proposed in this paper. At first, dynamic web service data discovery is carried out through Semantic Source Root translator. In Semantic Source Root translator, the service requests (source) and responses is performed in one semantic root form for easy selection of contextual features. Secondly, Indexer is used to index data services for performing search and extraction process. Then, Satisfied Data Discovery in RCWI mechanism is used to extract the most appropriate service based on user ratings. Finally, timing factor based thread creation helps to achieve user satisfaction and also perform the updates on refreshed threads. The data discovery process using RCWI improves the performance by applying neighboring cache mechanism which allows users to discover the services based on the online learning experiences. Experiment is conducted on factors such as scalability rate, data discovery performance level, time taken on refreshing the crawled threads.

Keywords - *Refresh Crawled Web, Indexing, Contextual Features, Data Discovery, Cache, Semantic Root, Service Requests, Online Learning.*

I. INTRODUCTION

With the recent advancements in web technology, it is significantly important to design and implement the service that satisfies the requirements of the user. Web service discovery is a method that identifies the service that highly meets the needs and wants of the user. Many researchers have contributed in the area of web service discovery.

[1]Research scholar, Department of Computer Science, Karpagam University, Coimbatore, Tamil Nadu, India E-mail : [1]a.r.rahuman@gmail.com,

[2]Associate Professor and Reader, Department of Computer Science, Periyar University, Salem, Tamil Nadu, India. E-mail: [2]ccsekar@gmail.com

Online Social Networks (OSNs) [1] designed a rule based system that filtered unwanted messages using machine learning languages. With this though the behavior of user at different time periods is highly sensitive to several contextual features it helped to improve semantic search. But, contextual feature selection and extraction failed in discovering data for web services. FOrum Crawler Under Supervision (FoCUS) [2] reduced the overhead while identifying relevant forum by applying implicit navigation path. However, the lack of user ratings and also sparse environment of data affected the scalability performance.

One of the main criticisms faced by search engines is that whenever the queries are issued, most of the search engines provide the similar results to the customers. At the same time, the responses related to the queries are short and ambiguous. Personalized Web Search (PWS) [3] improved ranking accuracy by using several features for personalizing a given query using historical click-based algorithm. However, PWS was a time consuming process as automatic prediction was not performed. Web Information Extraction using Bayesian Approach (WIE-BA) [4] was designed to improve the information being extracted minimizing the human intervention. VIsion based Data Extraction (ViDE) [5] extracted the data records from deep web pages.

World Wide Web comprises of enormous amounts of data. But, without proper organization, this enormous amount of data from raw web pages is of no use. Application of Natural Language Programming in Web (WebNLP) [6] provided a bidirectional model for efficient extraction of information from Web pages. However, the quality of service was not well explained. To address the Quality of Service (QoS) Semantic Web Service Composition (SWSC) [7] was introduced to provide an optimization model using genetic algorithm. Though QoS was provided, flexibility remained unaddressed because of the pre computation based on the context availability of web services. Web Page Recommendation (WPR) [8] used ontology for efficient representation of domain terms, web pages and the inter relationships between them.

In the current scenarios many business establishments provide their applications utilities on the web. Semantic Web Service Discovery (SWSD) [9] using web service composition provided a means for efficient categorization for similarity identification. But, the model seemed to increase the computation cost with the increase in the web service composition. To reduce the computation cost, semantic web and semantic web services were designed in [10] based on the annotation of data and functionality.

Hybrid Recommender System (HRS) [11] was designed using context-aware method to produce recommendation based on the requirements of the users. Current works on Semantic Web Services serverely poses the crisis of scalability and as a result, service repositories of complex nature cannot be handled in an efficient manner. In [12], semantic web services discovery was improved with the aid of SPARQL-based repository.

In this work, focus is made on improving the scalability rate of data discovery in web using Refresh Crawled Web based Indexing (RCWI) mechanism. The contributions of RCWI mechanism include the following:

(i) To improve the scalability rate of data discovery in web by applying Refresh Crawled Web based Indexing (RCWI) mechanism

(ii) To perform easy selection of contextual features by performing dynamic web service data discovery with the aid of Semantic Source Root translator that performs the service requests and responses in single semantic root form

(iii) To design an effective search and extraction process using indexer

(iv) To extract the most appropriate service based on user ratings using Satisfied Data Discovery

(v) To perform the updates on refreshed threads and to achieve user satisfaction with the application of creation of thread based on timing factor

(vi) To improve the overall performance during the data discovery process by applying neighboring cache mechanism based on online learning experiences

The structure of paper is as follows. In Section 1, data discovery on web with respect to scalability with existing works is described. In Section 2, literatures related to web service and data discovery is elaborated by comparing the current history. Section 3 explains about the proposed work Refresh Crawled Web based Indexing (RCWI) mechanism with neat architecture diagram and algorithmic steps to increase the scalability. Section 4 analyzes the experimental results and Section 5 provides the result analysis using table and graph values. Finally, the concluding remarks are included in Section 6.

II. Related Works

With basics of Service Oriented Architecture (SOA) and real time utilities of web has gained the attraction of most of the organization and developers to use the technology behind web service. With respect to services computing, service discovery is one of the important activities to identify and measure the required service according to the client's requirement. However, with the increasing availability of similar web services, QoS metric has become as one of the import measure for service differentiation.

Normalized Google Distance (NGD) [13] was measured in order to identify the similarity between the web services which resulted in better precision and recall values. However, the distributed nature remained unaddressed. Service Oriented Computing (SOC) [14] measured the life cycle of web services composition based on atomic versus composite web services. A Temporal Web Ontology Language (TOWL) [15] provided a means towards temporal aspects from the financial domain. Natural Language Processing (NLP) [16] techniques were introduced to improve the matching rate extracted from web service description and user query.

With the increase in purchases of products made in online, searching of products has helped the customers to identify their products of interest, as the customers nowadays make their shopping more often on the Web. Efficient identification of products and mapping the requirements of the customers with the product list using Annotation based Resource Description Framework (ARDF) [17] was introduced to address the needs and

want of the customers on web. But, graphical model of mapping remained an open issue. A general purpose algorithm called as the Mediation Spaces (MS) [18] was introduced to address similarity comparison problem and also reduced the human intervention to larger extent. However, scalability was not focused.

Taxonomy of web service discovery system was introduced in [19] to address the problems related to scalability using domain ontology-based approach. Though issues related to scalability were addressed, quality of web service discovery was not focused. To improve the quality of web service discovery, linked social service-specific principles was introduced in [20]. Based on the aforementioned techniques and methods, in this work we provide a p Refresh Crawled Web based Indexing mechanism to increase the scalability rate of data discovery in web.

III. REFRESH CRAWLED THREADS AND CONTEXTUAL FEATURE INDEXING FOR DATA DISCOVERY ON WEB SERVICES

In this section, we provide an overview of the proposed work Refresh Crawled Web based Indexing (RCWM) mechanism with the aid of structure of web service data discovery and a neat sequence diagram. In our proposed work, the architecture is built using a data discovery system with dynamic, scalable and flexible search capability index structure. Web services in Refresh Crawled Web based Indexing (RCWI) mechanism is characterized by contextual features.

The contextual features are selected from user queries and the features are placed in the index table. The index table is maintained by the indexer that helps widely to improve the performance of searching. The searched result is then presented to the users based on the contextual information using RCWI mechanism. The contextual approach on web for user result retrieval is built based on query context features, indexing of features and relevance ratings from the previous search process.

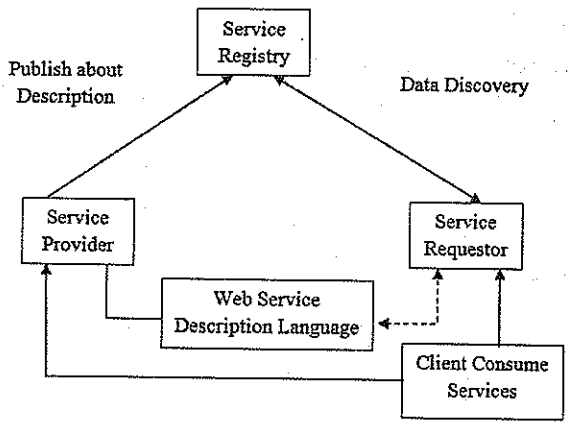

Figure 1: Structure of Web Service Data Discovery

Web services data discovery is the process of identifying the appropriate services on web that meets the client desired requirements. As illustrated in the figure 1, the structure of web service data discovery includes four components. These four components are Web Service Description Language, Service Provider, Service Requestor and Service Registry. The RCWI Service Providers register their web services at Service Registry whereas the Service Requestors search the web services from the Service Registry as illustrated in Figure 1. Finally,

the Web Service Discovery Language describes the complete information regarding the different contexts in RCWI mechanism. The Web Service Description Language in web is used by the clients to invoke the appropriate web services based on the contextual information using Satisfied Data Discovery Approach. The overall sequence diagram of data discovery using the proposed Refresh Crawled Web based Indexing (RCWI) mechanism is illustrated in Figure 2.

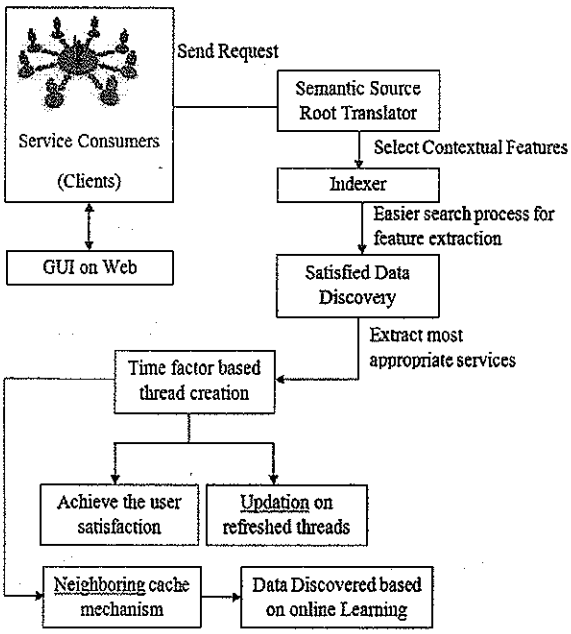

Figure 2 : Overall Sequence Diagram of Data Discovery using RCWI mechanism

With the aid of RCWI mechanism, the rapid development of data discovery process on the web service satisfies specific client requirements based on the contextual information. Figure 2 illustrates the sequence flow of the proposed work with two major functions. The initial function is to select and extract contextual feature for efficient location of services depending on the client requests. The second major function is to refresh the crawled web forms in timely manner to improve the performance and quality of services.

As illustrated in figure 2, the client send the request query on web search engine to convert the request query term into a single semantic root form for easy selection of contextual features. In this manner, the contextual features are selected and placed in an index table to improve the search efficiency. Next, the satisfied data discovery extracts the most appropriate services with higher scalability level. With the aid of online learning the RCWI mechanism predicts the label of feedbacks (i.e., ratings) at single instance at a time for the close true label context features. Then subsequently, timing factor based thread is created to achieve the user satisfaction and updation process. Thread created on web services using Neighboring cache mechanism as a result improve data discovery precision rate. An elaborate description about the semantic source root translator, indexer and satisfied data discovery is provided in an elucidated form in the forthcoming sections.

3.1 Semantic Source Root Translator

Data discovery framework in the web accepts the user input query request and perform the operation of semantic source root that efficiently utilize the request query for further processing. The effective utilization is carried out in RCWI mechanism using the semantic source root translator for the effective selection of contextual features. The contextual feature selection performed by the

semantic source root translator using RCWI mechanism is described as,

$$CFS = \frac{K^{fr/t}}{N} \frac{|R|}{fr/t} \quad (1)$$

Where 'K' denotes the keywords in the user query set in a contextual feature 'fr' and 't' denotes the time taken to select the features. The selected features are then rated as 'R' for easy interaction according to the requirements of the user. Finally, a single semantic root form is chosen with the selected context features on the web.

3.2 Indexer

RCWI mechanism uses the Context FIXING approach to enhance the searching capability. Indexer is used to index the services stored in the neighborhood cache based on the history information. With the aid of indexing mechanism, the context feature index structure is created and then performs the query operation to improve the search efficiency rate. The context feature based Indexing (Context FIXING) is illustrated in Fig 3.

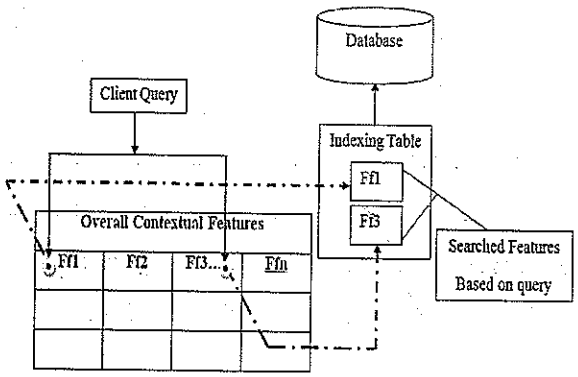

Figure 3 : Systematic View of Context FIXING process

Figure 3 describes the construction or systematic view of context feature based indexing and produces the search results for client query on the web. As illustrated in figure 3, to start with, the client query is submitted on the web, and the features are selected through the semantic root. The features are selected from the set of overall context features and indexed in the database table. The table uses the B-tree procedure to search the client result with minimal processing time.

The data discovery process using the Context FIXING in RCWI mechanism obtains the collection of the selected context features as input and insert the B-tree procedure. The B-tree procedure based context FIXING works in two steps. During the initial step, sequential search and extraction of the features is carried out. The second step creates the indexed units which are high effective in updating the client features based on insertion and deletion operation. The algorithmic step of the Context FIXING with B-Tree procedure is described as,

// Context FIXING B-Tree Construction

Input: Context Features (CFi = CF1, CF2,..,CFn), DT, Client Request Query (QRi = QR1, QR2,.., QRn), Tree D, Query_Request_Context_Terms, IT = 0

Begin

1: for each QRi do

2: Construct B-Tree

2.1: Perform Sequential Searching of features from DT

2.2: Perform update process

3: if (Query_Request_Context_Terms < Size (D) then

3.1 Extract Client Query Result

3.2: else

3.3: Perform Sequential Search with updated index table

3.4: end if

4: end for

End

Output: Fetches client result using data discovery information

The above algorithm is constructed with the collection of the context features 'C', and 'D' is the depth of the B-tree and 'i' entries denotes the indexing features in the B-tree. As described in the algorithm for each query request, a B-Tree is constructed and performs sequential search of features from the database table DT. Followed by this the query request context text is checked with the depth of the tree D. If the query request context terms are greater than the size of the tree D, then the client query result is extracted. If the conditions provide a false result, then sequential search is performed with updated index table. The depth is considered as an indexable units in RCWI mechanism Context FIXING. Finally, the context feature is searched and extracts the web information with higher performance data discovery services.

2.1 Satisfied Data Discovery

Satisfied Data Discovery in RCWI mechanism is used to extract the most appropriate service based on user ratings. The rating using RCWI mechanism is obtained through history information is formularized as,

$$Rating\ |R| = \frac{N\ (C_{rc})}{\sum_{I,C=1}^{n} Client_{IC}} \qquad (2)$$

where 'rc' denotes the rate count of the context features 'C' to the total number of client's requests on different set of contexts'Client c' for n clients. In order to obtain the overall rating, the total client count requests of every user are used.

3.4 Neighboring cache mechanism

RCWI mechanism adapts the neighboring cache mechanism and as a result the client obtains the rating value based on the history information. In RCWI, the neighboring cache mechanism is used to store the web services and improve the data discovery process using online learning approach. Web services through online learning approach predict the closer label of context feature ratings for each client request. RCWI mechanism also stores the service position in neighborhood cache using indexer associated with the time stamp to maintain the data cache in the repository.

3.5 Creation of Thread

RCWI mechanism creates a thread for each registry listed context features. The thread is created based on the time factor for fast and efficient retrieval of web services based

on data discovery. The RCWI mechanism provides the thread services in parallel to multiple clients to improve the web service response management for different clients. As a result, using RCWI mechanism, multiple clients can requests the query using the web thread crawler. The thread construction for multiple clients is described as,

$$Thread\ Construction = Input\ (Client1.Req1, Client2.Req2, Client3.Req3 \ldots . Clientn.Reqn) \quad (3)$$

With the aid of thread construction in web service data discovery, both larger and smaller set of client query requests is executed and managed to produce the feedback results. Each client may also request query more than one time to fetch the appropriate data discovery from web services. The timing factor is described as,

$$Timing\ Factor\ (T) = \frac{PIQR}{TTP} \quad (4)$$

Where 'PIQR' denotes the process of input query request based on thread creation with a Total Time Period 'TTP' denoting the start time of the client request to the end time (i.e., respond send to the request based on data discovery process in web). The thread construction based on timing factor helps to achieve user satisfaction and also performs the updates on refreshed threads. The thread is also updated and refreshed based on the changes in the context feature selection. The time taken on working with updated thread (i.e., updated context feature selection from user request query) is evaluated using (4) on the newly refreshed thread.

IV. Experimental Evaluation

In this section, the performance and experimental evaluation of Refresh Crawled Web based Indexing (RCWI) Mechanism is presented and experimented in JAVA platform. Amazon EC2 instances have been selected to perform the data discovery using the KDD cup dataset to evaluate the proposed system performance. By hosting an important data it quickly and easily processed and produces the feedback to the end users. Public Amazon Web Services uses the information for experimenting RCWI mechanism. Knowledge of all the present facts and declaration in the Freebase system is provided with an open database covering millions of theme with contextual features. KDD cup dataset provide the rating of the movie through online learning approach.

A data dump is the essential information provided on identifying the facts concerning each subject in Freebase. Freebase is an open database of the world's information, cover millions of theme in hundreds of group. Drawing from huge open datasets like Wikipedia, MusicBrainz, and the SEC archives simultaneously contain prearranged contextual information on a lot of popular topics. RCWI mechanism compares with the existing flexible rule-based system on Online Social Networks (OSNs) [1] and Forum Crawler under Supervision (FoCUS) [2] method. RCWI mechanism is experimented on factors such as data discovery performance level, time taken on retrieval of web services, F-score value of indexing, and contextual feature extraction rate.

The data discovery performance level $Data\ Discovery_{rate}$ is the ratio of user ratings $Ratings\ |R|$ to different number of clients I on different context C given as below. The data discovery performance level is measured in terms of percentage (%).

$$Data\ Discovery_{rate} = \frac{Ratings\ |R|}{\sum_{I,C=1}^{n} Client_{IC}} \quad (5)$$

The time taken TR on retrieval of web services is the time taken to perform the thread construction for multiple clients Client1, Client2,...,Client n. The time taken on retrieval of web services is measured in terms of milliseconds (ms).

$$TR = Time\ ((Client1.Req1 + Client2.Req2 + Client3.Req3 + \ldots + Clientn.Reqn)) \quad (6)$$

The F-score value of indexing considers both the precision and recall value. Precision (p) is the number of correct features indexed divided by the number of all returned features and the recall (r) is the number of correct features divided by the number of correct features that should have been returned.

$$F - score = 2 * \frac{p * r}{p + r} \quad (7)$$

The contextual feature extraction rate is the rate at which the service request $Service_{req}$ to service response $Service_{res}$ are made at a time interval T.

$$CFE_{rate} = \frac{Service_{res}}{Service_{req}} * T \quad (8)$$

V. Results Analysis of RCWI Mechanism

The RCWI mechanism is analyzed against Online Social Networks (OSNs) [1] and Forum Crawler under Supervision (FoCUS) [2] method. Each technique has its own respective data discovery performance level. The existing and proposed result is analyzed by providing several clients' request queries in JAVA with the help of table values and graph points. Table 1 tabulates the Data Discovery Performance Level with respect to the number of client request queries given as input. We make a comparison of our model RCWI with OSNs and FoCUS.

Table 1 : Data discovery performance level for data discovery in web

Client Request Queries (QR)	Data Discovery Performance Level (%)		
	RCWI	OSNs	FoCUS
3	55.35	43.31	35.29
6	59.75	47.72	42.70
9	63.85	51.83	46.81
12	68.45	56.42	51.40
15	60.25	48.22	43.20
18	71.35	59.33	54.31
21	74.55	62.55	57.53

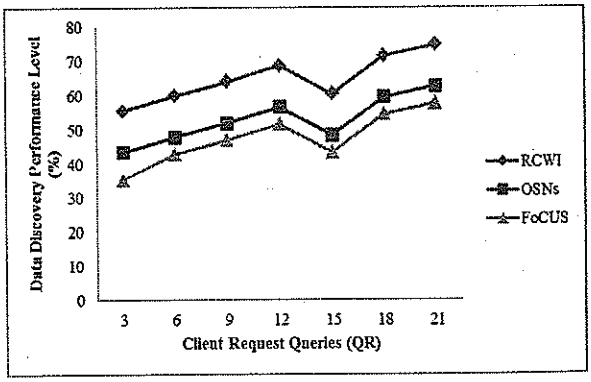

Figure 4 : Measure of Data discovery performance level

Figure 4 shows the performance of data discovery based on the client request queries using various mechanisms. Comparisons of Data discovery performance level are made with two other methods, Online Social Networks (OSNs) [1] and Forum Crawler under Supervision (FoCUS) [2] method. Though the data discovery performance level is not linear, a drastic improvement is observed when 3 to 12 queries were issued. But when the client request queries were in the range of 15, it reduces and than a drift changes has occurred. This is because the time taken to fetch the service request from the service registry is not fixed and varies according to the nature of query being issues. From the figure it is illustrative that the Data discovery performance level is improved using RCWI. This is because of the application of Satisfied Data Discover that obtains most appropriate service with the aid of neighboring cache mechanism using history information. As a result, the data discovery performance level is maximized by 16 – 21 % compared to OSNs. Moreover, with the application of satisfied data discovery the neighboring cache mechanism stores the web services using online learning approach resulting in the improved data discovery performance level by 22 – 36 % compared to FoCUS.

Table 2 : Comparison of time taken on retrieval of web service using RCWI, OSNs and FoCUS

Client Request Queries (QR)	Time taken on retrieval of web services (ms)		
	RCWI	OSNs	FoCUS
3	0.135	0.156	0.250
6	0.147	0.168	0.218
9	0.153	0.174	0.224
12	0.162	0.183	0.233
15	0.149	0.170	0.220
18	0.165	0.186	0.236
21	0.172	0.193	0.243

The time taken on retrieval of web service of our RCWI mechanism is presented in table 2. It is easy to find that the time taken on retrieval of web service is improved using RCWI mechanism than the state-of-art methods.

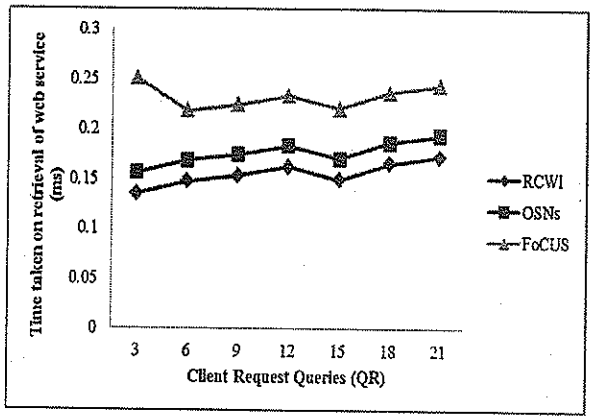

Figure 5 : Measure of Time taken on retrieval of web services

Figure 5 shows the sample set of result for Time taken on retrieval of web services based on different sizes of client request queries in the range of 3 to 21 at different time periods. From the figure it is illustrative that with the increase in the client request queries being made, the time taken for retrieving the results of web services also increases using all the methods. From the figure we can note that the time taken on retrieval of web services attains 0.153 ms for 9 client request queries and reduced to 0.149 ms for 15 client request queries. The Time taken on retrieval of web services is decreased using the proposed mechanism by applying thread that retrieves the web services in an efficient manner. Moreover, multiple clients requests the query in parallel using the web thread crawler using the timing factor for also updated threads resulting in the minimization of time taken on retrieval of web pages by 12 – 15 % and 41 – 85 % compared to OSNs and FoCUS respectively.

Table 3: F-score value of indexing for data discovery in web

Number of Contextual Features (CF)	F-score value of indexing (%)		
	RCWI	OSNs	FoCUS
1	0.33	0.22	0.19
2	0.38	0.27	0.24
3	0.44	0.33	0.30
4	0.49	0.38	0.35
5	0.51	0.40	0.37
6	0.45	0.34	0.31
7	0.65	0.54	0.51

In table 3 a summary of f-score value of the proposed RCWI mechanism and comparison made with OSNs [1] and FoCUS [2] is provided.

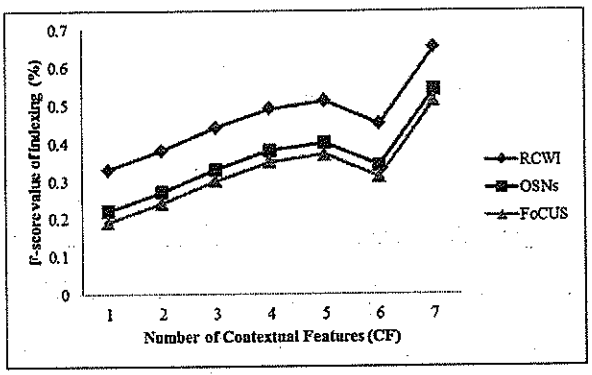

Figure 6 : Measure of F-score value of indexing

Figure 6 describes the F-score value of indexing with respect to the increasing amount of contextual features given as input for the RCWI mechanism, OSNs and FoCUS. We can see that when the number of contextual features increases the F-score value of indexing increases dramatically. In contrast, the performance of F-score value of indexing drops less than 20 % in terms of average F-score value of indexing. The F-score value of indexing is improved using the proposed RCWI mechanism because of the application of an effective search and extraction process using the indexer resulting in the increasing value of F-score. It shows that RCWI mechanism provides better F-score performance compared with the state-of-the-art-methods. With the application of Context FIXING with B-Tree procedure, the overall context features are indexed and stored in the database table. This in turn extracts the web information from the service registry with higher performance data discovery resulting in the improvement of F-score value of indexing by 16 – 33 % and 21 – 42 % compared to OSNs and FoCUS respectively.

Table 4 : Comparison of Contextual Feature Extraction rate using RCWI, OSNs and FoCUS

Number of Contextual Features (CF)	Contextual Feature Extraction rate (%)		
	RCWI	OSNs	FoCUS
1	55.78	49.76	42.73
2	60.25	54.23	48.20
3	63.33	57.31	50.28
4	69.12	63.10	58.7
5	73.45	67.43	60.40
6	65.37	59.35	52.32
7	75.86	69.84	62.81

Table 4 summarizes the three methods that we experimented for data discovery on web based on Public Amazon Web Services.

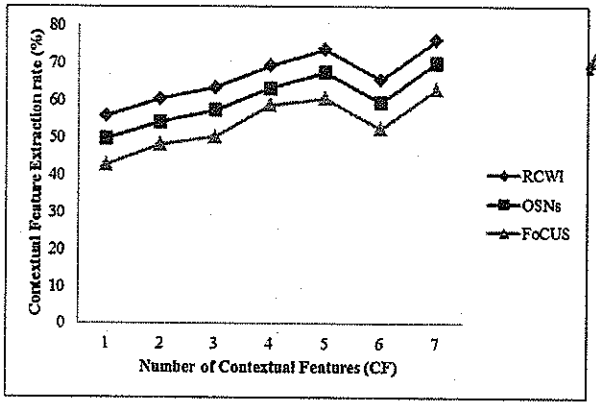

Figure 7 : Measure of Contextual Feature Extraction rate

Table 4 and Figure 7 shows the contextual feature extraction rate using the proposed RCWI mechanism and comparison made with two other existing mechanisms, OSNs and FoCUS. The Contextual Feature Extraction rate observed is high using RCWI mechanism by applying Semantic Source Root translator. The contextual feature extraction rate is improved with the application of Semantic Source Root translator. This is because the contexture feature is selected based on the semantic root that performs the service requests and responses in single semantic root form improving the contextual feature extraction rate by 7 – 10 % and 17 – 23 % compared to OSNs and FoCUS respectively.

VI. Conclusion

A Refresh Crawled Web based Indexing (RCWI) Mechanism based on contextual feature indexing, have been designed to improve the scalability rate of data discovery in web. We adopt web service data discovery using web service description language, design a service registry to obtain service request and provide efficient service response on web. With the effective design of semantic source root translator, easy selection of contextual features is performed improving the contextual feature extraction rate. With an efficient design of an Indexer model, search effectiveness and extraction is improved by applying satisfied data discovery resulting in the improvement of data discovery performance level. Finally, with the design of time factor based thread creation, user satisfaction is achieved with the help of the neighboring cache mechanism improving the time taken on retrieval of web services. The proposed context FIXING algorithm reduces the complexity and improves the scalability of search. Experimental evaluation is conducted with the Amazon EC2 instances using the KDD cup dataset to analyze the web service data discovery and measured the performance in terms of data discovery, time taken, F-score value on indexing and contextual feature extraction rate. Performances results reveal that the proposed RCWI mechanism provides higher level of data discovery and F-score value of indexing and also strengthen contextual feature extraction rate by consuming less time taken of retrieval of web services. Compared to the existing web service data discovery, the proposed RCWI mechanism is 18.97 % high in data discovery and F-score value of indexing improved by 22.24 % compared to state-of-art works.

References

[1] Marco Vanetti, Elisabetta Binaghi, Elena Ferrari, Barbara Carminati, and Moreno Carullo, *"A System to Filter Unwanted Messages from OSN User Walls,"* IEEE Transactions on Knowledge and Data Engineering, Vol. 25, No.2, February 2013.

[2] Jingtian Jiang, Xinying Song, Nenghai Yu, and Chin-Yew Lin, *"FoCUS: Learning to Crawl Web Forums,"* IEEE Transactions on Knowledge and Data Engineering, Vol. 25, No.6, June 2013.

[3] Zhicheng Dou, Ruihua Song, Ji-Rong Wen, and Xiaojie Yuan, *" Evaluating the Effectiveness of Personalized Web Search"*, IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No.8, August 2009.

[4] Tak-Lam Wong and Wai Lam, *" Learning to Adapt Web Information Extraction Knowledge and Discovering New Attributes via a Bayesian Approach"*, IEEE Transactions on Knowledge and Data Engineering, Vol. 22, No.4, April 2010.

[5] Wei Liu, Xiaofeng Meng, and Weiyi Meng, *" ViDE: A Vision-Based Approach for Deep Web Data Extraction"*, IEEE Transactions on Knowledge and Data Engineering, Vol. 22, No.3, March 2010.

[6] Chunyu Yang, Yong Cao, Zaiqing Nie, Jie Zhou, and Ji-Rong Wen, *" Closing the Loop in Webpage Understanding"*, IEEE Transactions on Knowledge and Data Engineering, Vol. 22, No.5, May 2010.

[7] Freddy Lecue and Nikolay Mehandjiev, *" Seeking Quality of Web Service Composition in a Semantic Dimension"*, IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No.6, June 2011.

[8] Thi Thanh Sang Nguyen, Hai Yan Lu, Jie Lu, *" Web-page Recommendation based on Web Usage and Domain Knowledge"*, IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No.6, July 2011.

[9] M.Gopianand, S.Jayakumar, *"Semantic Web Service Discover for Web Service Composition"*, IJCSMC, Vol. 3, Issue. 4, April 2014, pg.307 – 311

[10] Charles Petrie, *" Semantic Web and Semantic Web Services"*, IEEE Internet Computing, Apr 2006

[11] Atefeh Jajvand, Mir Ali Seyyedi, Afshin Salajegheh, *" A Hybrid Recommender System for Service Discovery"*, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1, Issue 6, August 2013

[12] Jose Marýa Garcýa, David Ruiz, Antonio Ruiz-Cortes, *"Improving Semantic Web Services Discovery Using SPARQL-Based Repository Filtering"*, Web Semantics, Elsevier, July 2012

[13] Jeberson Retna Raj, Dr.T.Sasipraba, *" Wed Service Discovery Based on Computation of Semantic Similarity Distance and QOS Normalization"*, Indian Journal of Computer Science and Engineering (IJCSE), Vol. 3 No. 2 Apr-May 2012

[14] Quan Z. Sheng, Xiaoqiang Qiao, Athanasios V. Vasilakos, Claudia Szabo, Scott Bourne, Xiaofei Xu, *"Web services composition: A decade's overview"*, Information Sciences, Elsevier, May 2014

[15] Viorel Milea, Flavius Frasincar, and Uzay Kaymak, *" tOWL: A Temporal Web Ontology Language"*, IEEE Transactions on Systems, Man and Cybernetics, Vol:42, Issue:1, Jan 2012

[16] Jordy Sangersa, Flavius Frasincara, Frederik Hogenboom, Vadim Chepegin, *" Semantic Web Service Discovery Using Natural Language Processing Techniques"*, Expert Systems with Applications, Elsevier, Jan 2013

[17] Damir Vandic, Jan-Willem van Dam, Flavius Frasincar, *" Faceted Product Search Powered by the Semantic Web"*, Decision Support Systems, Elsevier, Dec 2012

[18] Stefan Dietze, Alessio Gugliotta, John Domingue, Michael Mrissa, *"Mediation Spaces for Similarity-based Semantic Web Services Selection"*, International Journal of Web Services Research, Vol.x, No.x, 2011

[19] Soodeh Pakari and Esmaeel Kheirkhah and Mehrdad Jalali, *"Wed Service Discovery Methods and Techniques : A Review "*, International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol. 4,No.1, February 2014

[20] Wuhui Chen, Incheon Paik, and Patrick C.K Hung,, *" Constructing a Global Social Service Network for Better Quality of Web Service Discovery"*, IEEE Transactions on Services Computing, Mar 2013.

AUTHOR'S BIOGRAPHY

A. Abdul Rahuman, is currently pursuing his Ph.D. degree in computer science at Karpagam University in India. He has obtained his B.Sc. degree from Bharathidasan University, M.C.A. degree from Bharathidasan University and M.B.A. degree from Alagappa University. He has published 2 papers in international journals. His research interest lies in data mining – semantic search.

Dr. C. Chandrasekar, is an Associate Professor at Department of Computer Science - Periyar University, Salem, India. He perceived M.C.A, M.Phil, and Ph.D. and trained 16 plus students for their research work.