VOL. 13, NO. 12, JUNE 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A MULTI METRIC OPTIMIZED CLUSTERING FOR MATCHING AND RANKING OF WEB SERVICES

K. Meenakshi Sundaram and T. Parimalam
Department of Computer Science, Erode Arts and Science College, Erode, Tamil Nadu, India
E-Mail: lecturerkms@yahoo.com

ABSTRACT

Web services are software entities that have a well defined interface and perform a specific task. A number of web services are composed together to fulfill the customer requirements because single web service not always fulfill the customer requirements. It can be grouped based on different similarity measures. The domain-ontology-based PSO-inspired balanced iterative reducing and clustering using hierarchies (BIRCH) clustering and improved bipartite graph (IBG), determines the relevancy between web services based on concept distance similarity measure. Based on this similarity measure the web services are clustered and ranked. The major drawback of this method is, there is no single similarity measure is optimal for determining the relevancy between web services. So in this proposed work, an efficient clustering method is proposed that uses different matchers with different similarity measures to calculate the relevancy between web services. It enhances the improved bipartite graph based web service matching. But the different matcher may return different clustering results, so a majority voting algorithm called as Boyer-Moore majority vote algorithm is introduced to finalize the clusters. After clustering the web services, the web services are ranked based on QoS based Fuzzy clustering. Experimental results show that the proposed method the efficiency of web service composition in terms of accuracy, runtime, recall and precision.

Keywords: web service, web service composition, similarity measure, Boyer-Moore majority vote.

INTRODUCTION

Web services [7] are self contained component applications which can be defined, published, located and invoked over internet. Description of web services enables web service to be discovered and used by other web services. These descriptions are defined using a standard XML-based language. These descriptions are divided into two features are functional features and non functional features. The functional features are required to invoke the execution of a web service and non functional features such as reliability, response time, cost, etc. Web service composition is about determining web services which performs a specified task.

Web service composition [11] is considered as an important technology in domain of web service and its target is reusing existing web services. Web service makes it possible to achieve interoperability Business-to-Business (B2B) from interconnection services offered by multiple business partners composition based on business processes. The interconnection of web service to meet a certain business process is called web service composition. The web service composition is broadly classified into two types are syntactic composition and semantic composition. The syntactic web service composition [1] is based on the syntactic description whereas the semantic web service composition [2] is utilized for concepts of ontology for to add semantic descriptions instead of the parameter values. In the recent years several researches have been carried out on the composition of web services.

An efficient automatic web service composition [10] was proposed to improve the performance of web service composition by introduced domain-ontology-based PSO-inspired balanced iterative reducing and clustering using hierarchies (BIRCH) clustering in the pre-processing phase and improved bipartite graph in the service discovery and ranking phase. Domain-ontology-based PSO-inspired BIRCH dynamically and incrementally clusters the web services in hierarchical manner with optimizing the cluster results which reduce the processing time. The best service clusters was discovered by improved bipartite graph and the web services are ranked based on QoS based fuzzy ranking method. It ensured the selection of best composition plan in the planning, verification and execution phase. However, in domain-ontology-based PSO-inspired BIRCH clustering only the concept distance similarity measure was computed for clustering. Single similarity measure is not optimal to find optimal clustering which affects the ranking of web service compositions.

So in this proposed work, an efficient method for web service clustering is introduced based on different similarity measures. In this proposed work, there are four different measures are Lin similarity, Wup similarity, Path similarity and vector similarity. These similarity measures are handled by Lin matcher, Wup matcher, path matcher and vector matcher. The multi metric matching process improves the improved bipartite graph based web service ranking. But these matcher leads to different clusters and it can be finalized by majority voting algorithm called as Boyer-Moore majority vote algorithm. Then the discovered clusters are ranked based QoS based fuzzy ranking method. Thus the proposed clustering method improves performance of web service composition ranking.

LITERATURE REVIEW

A novel architecture was proposed [8] for semantic web service composition using clustering and ant colony optimization. This novel architecture solved the problem of reduction of accuracy, unselect optimal composition and increase of response time. The semantics was expressed through OWL-S language for description of web services. Here, clustering was used for categorizing the web services to allow web service consumers to determine relate services effectively. In addition to that, ant colony algorithm was utilized to determine the best set of web services which have high combining ability. The efficiency and successful rate of the web service composition was improved by novel architecture. But still precision of this method is low.

Two complementary approaches were proposed [9] for ranking and clustering the relevant web services. Initially, a methodology was proposed to rank the relevant web services for a given request by user and introduced objective measures based on dominance relationships described among the services. Then the methods for clustering was investigated in a way that reflects and reveals the different trade-offs between the matched parameters. The proposed clustering method provides high quality clustering results. But it is computationally expensive to cluster the web services.

A novel technique [4] was proposed to cluster web service documents into functionally similar service groups utilizing Cat swarm optimization algorithm. This technique categorizes web services which dealt with their functional diversity. There are two techniques were applied to cluster the web services are K means and bio-inspired Cat swarm optimization algorithm. In the K means clustering, the web services are clustered based on k values, center and Euclidean distance between center and web services. Whereas the Cat swarm optimization algorithm based clustering is based on two sub procedures based on live in nature while hunting pray, termed as the seeking mode and tracing mode. The Cat swarm optimization algorithm performs better than the k means clustering. But in case of tracing mode of Cat swarm optimization algorithm the centers were changed randomly for better clusters which increased time consumption and precision of web service clustering.

A web service recommender system [12] was proposed to improve the fuzzy clustering algorithm. It was achieved by utilizing the similarity calculations as a distance measure function that make prediction for the missing QoS values more accurately in the case of data sparsity. Another contribution of this recommender system was to take services QoS properties as a multi dimensional object. A mapping function was provided that normalize the value of each dimension to a score. Collaborative filtering and fuzzy clustering was combined during the process of making prediction. However, this method does not provide more satisfactory results to service users.

A new backtracking approach [6] was proposed for semantic web service discovery. It generated a sorted list of available operation interface annotations which was utilized to backtrack to the services that are the most matching with the query service. However, the backtracking approach, being an algorithm to yield only an optimal mapping for each of the operations, gives a limited precision.

An efficient web service ranking approach [3] was proposed based on collaborative filtering. It explored user behavior where the query history and invocation were utilized to infer the potential user behavior. CF-based user similarity was calculated through similar invocations and similar queries between users. Three aspects of Web services are QoS utility, CF based score and functional relevance was considered for the final Web service ranking. Three ranks were calculated for three factors which avoid the impacts of different range, units and distribution of variables. The final Web service ranking was obtained by using a rank aggregation method based on rank positions.

A user preference based web service ranking (UPWSR) [5] was proposed to rank the web services based on QoS aspect and user preferences of web services. The proposed framework allows the user to specify the local and global constraints for composite web services which improves flexibility. UPWSR algorithm identifies best fit services for each task in the user request and, by choosing the number of candidate services for each task, reduces the time to generate the composition plans. To tackle the problem of web service composition, QoS aware automatic web service composition (QAWSC) algorithm proposed in this paper was based on the QoS aspects of the web services and user preferences. The proposed framework allows user to provide feedback about the composite service which improves the reputation of the services.

MATERIALS AND METHODS

In this section, the proposed majority voting based Ranking algorithm is described in detail. Initially, the necessary pre-requirements for the efficient service discovery are carried out from the repository. Then the services are clustered using Domain-ontology-based PSO-inspired BIRCH clustering where the concept distance measure is used to cluster the web services. In order to get the accurate matching results for web service clustering, multi metric PSO inspired BIRCH clustering is proposed in this paper. It improves the improved Bipartite graph based matching by considering multiple similarity metric. In the proposed work, different matchers are used where the similarity between web services are determined by using different similarity measures. Then the result of the final clustering is obtained by majority voting between the clustering results of each matcher. Based on the final clustering results, ranking algorithm ranks the web services.

Different matcher based matching and clustering of web services

The domain ontology based PSO inspired BRICH clustering is improved by considering the web service matching process during clustering the web services. A web schema matching is the process of identifying that two web service operations are semantically related to each other. In the matching process, matcher is used to compute the semantic similarity between web services. There are four matcher are Lin matcher, Wup matcher,

ARPN Journal of Engineering and Applied Sciences

path matcher and vector matcher used to match and cluster the web services.

a) Lin matcher

Lin matcher is based on semantic similarity which defines to similarity between two concepts of web services in taxonomy. The similarity between two words w and v using the classes which contain them, such as $w \in C_1$ and $v \in C_2$. The taxonomy is assumed as a tree, if $w \in C_1$ and $v \in C_2$, the commonality between w and v $w \in C_0 \cap v \in C_0$, where C_0 is most specific class which subsumes C_1 and C_2. P (C) is probability that a randomly selected object belongs to C. Thus, the similarity between w and v is:

$$Sim_{Lin}(w,v) = \frac{2 \times \log P(C_0)}{logP(C_1) + logP(C_2)}$$

Based on the above similarity measure, Lin matcher matches and clusters the web services.

b) Wup matcher

Wup is a similarity measure based on path length between concepts of web services. It utilized the depth of the each concept, that is, the length of the path from the root node to the concepts, and the depth of the least common subsume (LCS) of the two concepts, which is the most specific concept is shared as ancestor. The wup similarity measure is calculated as follows:

$$Sim_{Lin}(C_1, C_2) = \frac{2 \times depth\ (LCS(C_1, C_2))}{(depth(C_1) + depth(C_2))}$$

If the two concepts of web services are same then the similarity score is 1 otherwise it will be 0. Based on the above similarity measure, Wup matcher matches and clusters the web services.

c) Path matcher

Path similarity computes the semantic relatedness of word senses by counting the number of paths along the shortest path between senses in the 'is-a' hierarchies of WordNet. The path lengths include the end nodes. It can be computed by using following equation:

$$Sim_{Path}(C_1, C_2) = \max_{c \in S(C_1, C_2)} [-\log p(c)]$$

where, $S(C_1, C_2)$ is the set of concepts that subsume both C_1 and C_2. P (C) is probability that a randomly selected object belongs to C.

By using the above similarity measure the Path matcher matches and clusters the web services.

d) Vector matcher

Vector similarity is a method that calculates similarity between word v and word w utilizing vectors of the two words. A vector for word is derived from the close neighbors of v in the WordNet. Close neighbors are all words that co-occur with w in a larger context or a sentence. Hence this similarity between of the two word can be computed by the cosine between two vectors:

$$Sim(v,w) = corr(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2 \sum_{i=1}^{N} w_i^2}}$$

where $\vec{v}$ is the vector of word v and $\vec{w}$ is the vector of word w and N is the dimension of the vector space.

Based on the above similarity measure, vector matcher matches and clusters the web services.

The different matcher leads to different clustering results. In order to provide final clustering result, Boyer-Moore majority vote algorithm is introduced.

Boyer-Moore majority vote

The Boyer-Moore majority vote algorithm finds a majority of web services in the same cluster, if there is one: that is, a web service that occurs repeatedly in the same cluster for more than half of the clusters of the input. However, there is no majority, the algorithm will not detect that fact, and will still output one of the cluster of web service.

Boyer -Moore majority vote algorithm

Input:	different clusters with different web services from matchers
Output:	Final Cluster
Step 1:	Initialize an cluster m and a counter i=0
Step 2:	for each cluster x of the input sequence
Step 3:	if i=0 then
Step 4:	assign m=i and i=1
Step 5:	else
Step 6:	if m=x and i=i+1
Step 7:	else
Step 8:	assign i=i-1
Step 9:	Return m //m denotes the final cluster

At the end of this process, if the web service has a majority, it will be the cluster stored by the algorithm. After clustering process, the web services are needed to rank the services in the discovered clusters based on QoS parameters. The ranking is performed based on QoS based Fuzzy ranking method.

RESULTS AND DISCUSSIONS

In this section, the experimental results are conducted with a 250 number of web services with service requests ranges from 100 to 400. In the experimental results, comparisons are made with a minimum number of 100 service requests. The efficiency of the proposed method is tested in terms of accuracy, precision and recall.

Accuracy

Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined. Accuracy can be calculated from formula given as follows:

$$\text{Accuracy} = \frac{\text{True positive} + \text{True negative}}{\text{True positive} + \text{True negative} + \text{False positive} + \text{False negative}}$$

Table-1, tabulates the value of accuracy between existing particle swarm optimization inspired balanced iterative reducing and clustering using hierarchies improved bipartite graph based matching and QoS based fuzzy ranking (PSOIBIRCH-IBPG-FQR) and proposed multi metric based balanced iterative reducing and clustering using hierarchies-improved bipartite graph based matching and QoS based fuzzy ranking (MMPSOBIRCH-IBPG-FQR).

Table-1. Comparison of accuracy.

No. of web services	**Methods**	
	PSOIBIRCH-IBPG-FQR	**MMPSOBIRCH-IBPG-FQR**
50	81	88
100	84	92
150	87	95
200	90	97
250	93	98.5

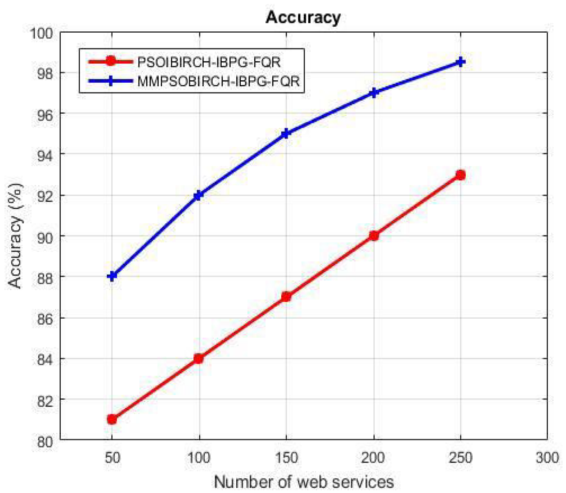

Figure-1. Comparison of accuracy.

Figure-1, shows the comparison of accuracy between existing PSOIBIRCH-IBPG-FQR and proposed MMPSOBIRCH-IBPG-FQR. X axis represents number of web services and Y axis represents accuracy value in (%). From the Figure-1, it is proved that the proposed MMPSOBIRCH-IBPG-FQR method has high accuracy than the PSOIBIRCH-IBPG-FQR method.

Precision

Precision is computed based on the clustering and ranking of web services for web service composition at true positive and false positive prediction.

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive (FP)}}$$

Table-2, tabulates the value of precision between existing PSOIBIRCH-IBPG-FQR and proposed MMPSOBIRCH-IBPG-FQR.

Table-2. Comparison of precision.

No. of web services	Methods	
	PSOIBIRCH-IBPG-FQR	MMPSOBIRCH-IBPG-FQR
50	0.74	0.80
100	0.78	0.86
150	0.83	0.90
200	0.85	0.92
250	0.90	0.98

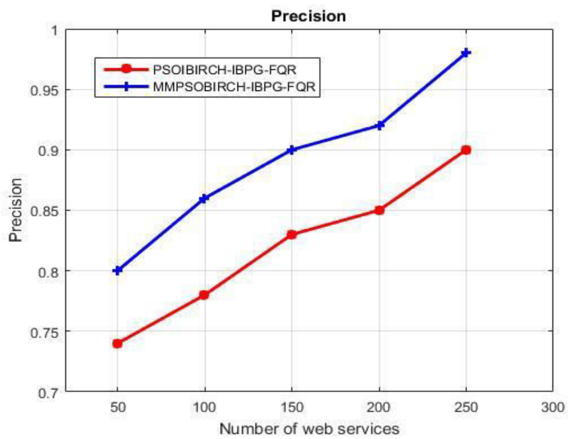

Figure-2. Comparison of precision.

Figure-2, shows the comparison of precision between existing PSOIBIRCH-IBPG-FQR and proposed MMPSOBIRCH-IBPG-FQR. X axis represents number of web services and Y axis represents precision. From the Figure-2, it is proved that the proposed MMPSOBIRCH-IBPG-FQR method has high precision than the PSOIBIRCH-IBPG-FQR method.

Recall

Recall is calculated based on the clustering and ranking of web services for web service composition at true positive and false negative predictions.

$$Recall = \frac{TruePositive\ (TP)}{TruePositive(TP) + FalseNegative\ (FN)}$$

ARPN Journal of Engineering and Applied Sciences

Table-3, tabulates the value of recall between existing PSOIBIRCH-IBPG-FQR and proposed MMPSOBIRCH-IBPG-FQR.

Table-3. Comparison of recall.

No. of web services	Methods	
	PSOIBIRCH-IBPG-FQR	MMPSOBIRCH-IBPG-FQR
50	0.76	0.81
100	0.79	0.86
150	0.85	0.9
200	0.89	0.94
250	0.92	0.99

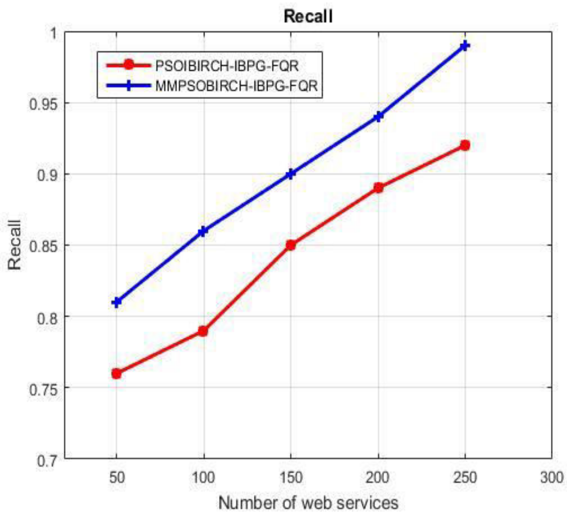

Figure-3. Comparison of recall.

Figure-3, shows the comparison of recall between existing PSOIBIRCH-IBPG-FQR and proposed MMPSOBIRCH-IBPG-FQR. X axis represents number of web services and Y axis represents accuracy value in (%). From the Figure-3, it is proved that the proposed MMPSOBIRCH-IBPG-FQR method has high recall than the PSOIBIRCH-IBPG-FQR method.

CONCLUSIONS

In this paper, an efficient method for web service clustering and ranking is proposed. There are four different similarity measure are Lin similarity, wup similarity, path similarity and vector similarity is computed in four different matchers Lin matrcher, wup matcher, path matcher and vector matcher to cluster the web services. The clustering results of one matcher may vary from the other matcher, so the final clustering is decided based on majority voting algorithm named as Boyer -Moore majority vote algorithm. It calculates the majority vote between matcher to finalize the cluster. Then the web service are ranked using QoS based Fuzzy Ranking method. The experimental results prove that the proposed method performs better than the existing method in terms of run time, accuracy, precision, recall and F-measure.

REFERENCES

[1] Gabrel V., Manouvrier M., Moreau K.& Murat C. 2017. QoS-aware Automatic Syntactic Service Composition problem: complexity and resolution. Future Generation Computer Systems.

[2] Hatzi O., Vrakas D., Nikolaidou M., Bassiliades N., Anagnostopoulos D.& Vlahavas I. 2012. An integrated approach to automated semantic web service composition through planning. IEEE Transactions on Services Computing. 5(3): 319-332.

[3] Kang G., Liu J., Tang M., Cao B. & Xu Y. 2015. An effective web service ranking method via exploring user behavior. IEEE Transactions on Network and Service Management. 12(4): 554-564.

[4] Kotekar S. & Kamath S. S. 2016. Enhancing service discovery using cat swarm optimisation based web service clustering. Perspectives in Science.8, 715-717.

[5] Mallayya D., Ramachandran B. & Viswanathan S. 2015. An automatic web service composition framework using QoS-based web service ranking algorithm. The Scientific World Journal.

[6] Prasanth S. D. 2013, March. A backtracking approach for semantic matchmaking algorithm in Web service discovery. In Circuits, Power and Computing Technologies (ICCPCT), 2013 International Conference on (pp. 1065-1070). IEEE.

[7] Rodriguez-Mier P., Pedrinaci C., Lama M. & Mucientes M. 2016. An integrated semantic Web service discovery and composition framework. IEEE Transactions on Services Computing. 9(4): 537-550.

[8] Rostami N. H., Kheirkhah E. & Jalali M. 2014. An optimized semantic web service composition method based on clustering and ant colony algorithm. arXiv preprint arXiv:1402.2271.

[9] Skoutas D., Sacharidis D., Simitsis A. & Sellis T. 2010. Ranking and clustering web services using multicriteria dominance relationships. IEEE Transactions on Services Computing.3 (3): 163-177.

[10] Meenakshi Sundaram, K. & Parimalam T. 2017. PSO-inspired BIRCH and Improved Bipartite Graph for Automatic Web Service Composition. International

ARPN Journal of Engineering and Applied Sciences

Journal of Applied Engineering Research.12 (8): 1765-1771.

[11] Wang H., Zou B., Guo G., Zhang J. & Yang D. 2015. Integrating trust with user preference for effective web service composition. IEEE Transactions on Services Computing.

[12] Zhang M., Liu X., Zhang R. & Sun H. 2012, June. A web service recommendation approach based on QoS prediction using fuzzy clustering. In Services Computing (SCC), 2012 IEEE Ninth International Conference on (pp. 138-145). IEEE.