ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(22), DOI: 10.17485/ijst/2016/v9i22/95136, June 2016* ISSN (Online) : 0974-5645

SSC Based RS: An Efficient Service Recommendation System for Handling Big Data Applications

R. K. Saranya[1*] and V. L. Jyothi[2]

[1]Sathyabama University, Chennai -600119, Tamil Nadu, India;
saranya.rks@gmail.com
[2]CSE Department, Jeppiaar Engineering College, Chennai -600119, Tamil Nadu, India;
jyothivl15@yahoo.com

Abstract

Objectives: To find an appropriate web service and reduce the time taken for introducing web service, to improve parallel processing, to reduce the complexity and to improve its scalability and efficiency in big data environment. **Methods:** MapReduce framework in Hadoop platform is for increase the efficiency and scalability in big data domain, SSC based RS, web service information are structured in hierarchical format. The proposed system calculates the semantic comparison between the big data applications. **Findings:** SSC based RS (Semantic Similarity Calculation Based Recommendation System) is used to efficiently suggest better services for the requested users, by using semantic dictionary the semantic similarity will be calculated. Here, the services are stored in the hierarchical structure will increase the recommendation process faster. An experimental result shows that our proposed algorithm provides a suitable recommended service compared to other existing approaches. **Applications/Improvement:** In Big Data the proposed technique improves the efficiency and scalability by applying MapReduce parallel processing standard on Hadoop environment.

Keywords: Big Data, Hadoop, Map Reduce, Recommender System, Web Service

1. Introduction

For Information Technology (IT) organizations, Big Data[1,2] is considered as a big challenge. The answer to these types of challenge in transferring more and more from offering hardware to provisioning more convenient solutions to software. It also provides critical challenges and new opportunities to academic world and industry[3]. With the fame of cloud computing and service computing, affluent functionality services are increasingly set up in cloud computing infrastructures. Nowadays, service consumers encountered unique complexities in identifying suitable ones from the great services. RSs (Recommender systems) algorithms and smart applications to help consumers in a selection process where the consumer wants to choose several products among a probably overwhelming set of alternative services or items. Similarly, in various big data applications, the tendency of big data masquerade serious impacts on SRSs (Service Recommender Systems). Based on increasing amount of different web services, successfully suggest an appropriate web service to the service consumer with his/ her preference become a significant research issue[4]. SRS was illustrated as precious devices to assist customers to handle service overload and offer suitable recommendations to the customers[5]. An example of such realistic applications contains books, web pages, Compact Discs and several other products use RSs[6,7] now. Over the period of last ten years, there has been much research done in academics and industry on implementing novel mechanisms for SRS[8,9]. The measurements of similarity among the web services are converse in the literature, traditional research work use various algorithms and that diverge in their performance. Some of the various algorithms are taken into considerations which are of GA (Genetic Algorithm), VSM (Vector Space Model) and LMS (Least Mean Square). Since LMS is an iterative algorithm which has largest number of iterations and it consumes more time. Also it identifies only minimum square error. Another existing technique which is

* *Author for correspondence*

genetic algorithm has more number of iterations since which requires more processing time compared to LSM. Compared to other two existing algorithms vector space model is better than genetic algorithm and LSM because of its simple iteration and efficient for indexing data. The vector space model is efficient in present search results on keywords, but it fails to take into account. Research work[10] google Normalized distance is used to compute the semantic comparison among two concepts, this is a statistical process based on the outcomes returned by the Google search engine and it is not taken into account. Research work[11] uses some metrics to calculate the semantic comparison, and use some metrics to compute the syntactic resemblance. Disambiguation of terms for which they want to compute the resemblance is not used. Moreover, in this work precision is not measured in their technique. In [12] authors measure the similarity between the web services by calculating the similarities between the Based on the account of the different concepts included in the WSDL file the similarities will be calculated. But the mainstreams of web services are not documented, so this method is not very convenient. Research work[13] utilizes the same mechanism to assess a similarity matrix except their mechanism doesn't surpass seventy percent in recall and precision values. The author in[14] identifies two significant conditions with evaluation in the Collaborative Filtering (CF) RSs. CF system with several consumers and offering suggestion for new customers. Moreover, it proposes a hierarchical clustering is a new mechanism for CF which attempt to equalize robustness, prediction correctness and realistically prove that this algorithm is particularly efficient in dealing with the prior situations[15]. The author in[16] proposed a clustering technique named Bi-clustering algorithm disclosing the duality among items and customers by concurrently group them in together available dimensions. The proposed algorithm utilizes a new parallel computation attains limited user similar predilection. The authors apply adjacent bi-clusters in aggregation with two dissimilar kinds of bi-clustering approaches x Motif and B I max for coherent and constant bi-clustering, correspondingly. The research work[17] proposed a collaborative filtering algorithm on Hadoop distributed file system. This algorithm divides the datasets for solving the scalability problem. This research work does not have any favorable efficiency and scalability if the number of data increases. The research study[18] presents a user profiling system, which is derived from a folksonomy data and executes a scalable RS by executing MapReduce and Cascading concepts. Research work[19] proposed an efficient (WS) web service clustering algorithm to create a classification structure by using the mixture of human knowledge and AI (Artificial Intelligence). This research work also utilizes Web Service Description Language document for converting the WSs are into vector format. To guarantee the presented approach performance, a self-organized NN (Neural-Network) approach is implemented. In[20], the author proposed Architecture based clustering and filtering approach to reduce the consuming time period for execution because of the difficult reasoning. It contains a large amount of components aggregated to handle the single features of a process or task. Finally, the aggregated processes were clustered for gradual filtering. As per the preferences of user and ontology, the author derived a WS clustering technique in[21]. This algorithm is attained to remove different WSs and produces the results of semantic similarity computation in a short period of time. The author in[22] presented a semi-supervised clustering algorithm for WS composition. This clustering algorithm is done with the help of tags, constraints, and the supervised learning using a large number of unlabeled datasets. The research work[23] presents a novel clustering technique namely WT Cluster which is developed to both tag annotations and Web Service Description Language documents in its procedure. This research work considers Web Service Description Language information which will limits accuracy of clustering process. WT Cluster technique clusters Web Service Description Language information with semantic information, functional description and contextual information. Research work[24] presents a search results related to a WS clustering and ranking in a dominance notion. This algorithm executes similarity calculation by presenting numerous criterions which does not aggregates the similarity values of each web service parameters. Research work[s] presented a novel clustering mechanism named probabilistic clustering algorithm that develops the earlier distributed vector space based search engine for WS clustering.

2. Proposed Work

2.1 Overview

Our proposed system develop a novel service recommendation system named SSC based RS (Semantic

Similarity Calculation Based Recommendation System) to efficiently suggest better services for the requested users. Our proposed mechanism utilizes two important concepts. These are semantic similarity calculation and Semantic based K-means algorithm. Figure 1 shows the proposed system architecture for proposed SSC based RS approach. In that figure, initially big data applications are stored in Distributed storage environment like Hadoop. Our proposed system applies on MapReduce framework in Hadoop platform for increase the scalability and effectiveness in big data domain. In this file system, the files are stored in text format. To improve parallel processing and reduce the complexity, the files are transformed into chunks. Chunk means large piece of data. To develop the performance of our SSC based RS, web service information is structured in hierarchical format. Using semantic similarity process, our proposed systems work out the semantic resemblance between the big data applications. Here, our system utilizes semantic dictionary for compute the semantic resemblance between the words. Based on the similarity values the applications are clustered with the help of semantic based k-means algorithm. Using our proposed mechanism, most suitable results and modified service recommendation list to the requested user might be displayed. Finally, our proposed system economically presents an adapted service recommendation list and recommending the most appropriate service(s) to the end users.

2.2 Working Methodology

Our proposed Semantic Similarity Calculation Based Recommendation System consists of five important stages. They are (i) Distributed Storage Configuration and web service big data preparation, (ii) Hierarchical Structure, (iii) Query Retrieval Process (iv) Semantic Similarity Calculation and (v) Clustering Process

2.2.1 Distributed Storage Configuration and Web Service Big Data Preparation

Major web service Applications are publicly available, from that enormous Collection of data is retrieved from open source datasets. Using Java API the text files were read and manipulated that is easily modifiable. By using the recommendation system, text files in distributed Systems are taken from the web services running in the server machine through a web service client process. The data retrieved through the recommended systems are provided with a clean graphical user interface for the query on demand. Light weighted traversal of data using XML is used on the recommendation application for each process. All the datasets are stored in the format of text.

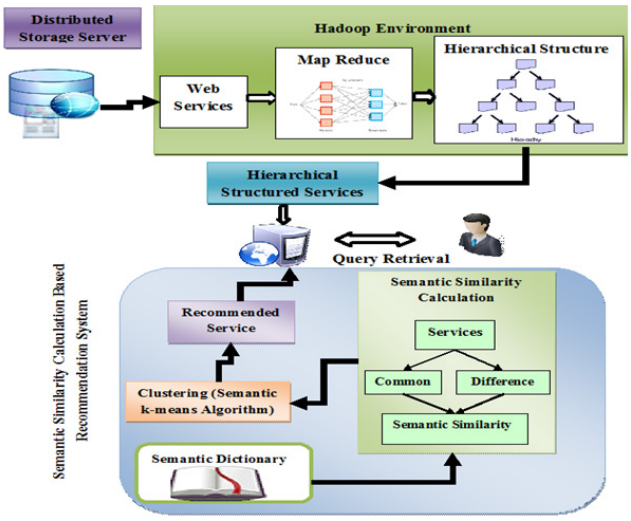

Figure 1. Proposed system architecture.

2.2.2 Hierarchical Structure

Hierarchical structure is a structure which groups the datasets based on the similarities in the dataset. In brief, our proposed system considers book domain which contains huge amount of books. So, in sorting process to reduce the complexity, the books are partitioned into hierarchical structure. Our proposed system use MapReduce for convert the web services into hierarchical structure as shown in Figure 2. This structure helps to categorize the different types of web services. It also improves the efficiency of data retrieval process.

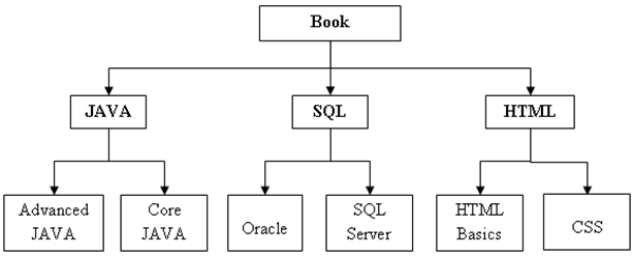

Figure 2. Hierarchical structure.

2.2.3 Query Retrieval Process

This stage describes the details about query retrieval process. In this stage, user or consumer gives query to the

server. At the time the server matched the relevant details or suitable applications based on the user's query. If any of the applications matched, the result should be displayed to the respected user. In query retrieval process, the server predicts better services based on the query of lively user. Based on the semantic resemblance calculation and semantic based k-means algorithm results, our system suggests and predict the better web services for requested user which improves the service scalability. The details of algorithms used for the data retrieval process is described in the below sub section

2.2.4 Semantic Similarity Calculation

The semantic similarity calculation is a process of allocating a words pair W1, W2 and a actual number R that specifies the similarity degree among the words. The measurement of the similarity is being done with the comparison of 2 words. Thus, the 2 words are semantically similar in case of synonyms (direct meaning) or in antonyms (opposite meaning). These words are also used in similar manner or in a similar method and are used in same situation or in a different situation. Our proposed approach utilizes the similarity calculation mechanism, which is described in method 1. To achieve more accurate similarity, our proposed system utilizes semantic dictionary. Dictionary is used for identify the meaning of words with substituting a synonym keyword.

Method 1: Semantic Similarity Calculation

The Calculation of proposed Semantic Similarity algorithm is based on the semantic similarity among the active user query and the stored documents. Proposed work considers the distance of two concepts in semantic domain which is determined by their common and difference. Assume that the taxonomy is a tree, if (user keyword) $x_1 \in C_1$ and (domain keyword) $x_2 \in C_2$, the unity among x_1 and x_2 is $x_1 \in C_0 \cap x_2 \in C_0$, where C_0 is the most specific class that include both C_1 and C_2. With the help of semantic dictionary, we can easily find the similar words of x_1 and then finds the similar words in x_2. This improves the retrieval process more effectively because it provides more keywords, which increases the similarity level between two words. Therefore, the similarity between x_1, x_2 is defined as,

Method 1:

$$\text{Simi}(x_1, x_2) = \frac{2^*\log_2(c_0)}{\log_2(c_1) + \log_2(c_2)}$$

2.2.5 Clustering Process

Clustering is an important stage in our proposed approach. Clustering is a method to partition a set of objects into clusters; the objects in the same cluster are more related to each other than objects in dissimilar clusters according to some defined criteria. Our proposed clustering mechanism contains two important parts. Such as (i) Cluster initialization and (ii) Similarity computation

2.2.6 Cluster initialization

In cluster initialization, the cluster size is initialized based on the number of data point and its cluster size. In cluster initialization process, number of clusters is predefined. After cluster initialization, create the array which includes the data array until the value is equal to the size of the cluster.

2.2.7 Similarity Computation

Similarity computation is one of the major sub processes of clustering mechanism. This computes the similarity values using method 1 and the values like user keyword, cluster size, web services and data objects. Method 2 describes the details about our proposed clustering algorithm.

Method 2: Semantic Based K-means clustering

Input: Web Services WS, integer K, user keyword x_1, and domain keyword x_2;

Output: Cluster Cl, Cluster Best_Service; // K- Cluster size

Process:

Step 1: for init: = 1 . . . Maxi_Initialization do //Cluster Initialization

Step 2: C_1: =Randomly_initialize (x_1, x_2 WS, K); //C-Cluster

Step 3: for each Simi(x_1, x_2) $\in C_1$ do

Step 4: Similarity_Calculation: = Method 1 (Simi(x_1, x_2));

Step 5: while iterat < maxi_Iterat do //Create object S for cluster

Step 6: for each S ∈ WS do //Assignment part C_l
Step 7: $S._{service\_id} = max_{simi}(x_1, x_2) \in C_l$
Step 8: for each $Sim(x_1, x_2) \in C_l$ do//update part
Step 9: Similarity_Calculation:= Method 1 ($Simi(x_1, x_2)$);
Step 10: if maximum similarity in user key word and WS
Step 11: Best_Service = C_l;
Step 12: end while
Step 13: end for
Step 14: return Best_Service

3. Results

3.1 Experimental Setup

To calculate the performance of our proposed approach, a sequence of experiments on a below mentioned dataset were conducted. In this experimentation, we implemented and calculate the proposed methods with the subsequent configuration: Intel i5(R), CPU G2020, 4GB RAM and processer speed 2.90 GHz. The software technology is used for implementation is HADOOP in Linux. To compute the performance and accuracy of SSC based RS approach, Mean Absolute Error (MAE - is a calculation of the recommendation deviation from their similarity values) which is used in this research work. Consider one online book shopping domain; in this shopping domain there are seven web services (i.e., JAVA, SQL, ASP.Net, PHP, HTML, CSS and PYTHON). In Table 1, list of web services taken for semantic similarity calculation, number of description and service providers for each web services are described.

Table 1. Web services and its details for similarity calculation

Web Services	No. of Description	Service Providers
JAVA	10	15
ASP.Net	12	10
SQL	9	15
PHP	8	15
HTML	13	15
CSS	11	15
PYTHON	7	10

3.2 Accuracy of Proposed System

MAE (Mean Absolute Error) values of SSC based RS approach decreases as per increasing cluster value because web services are alienated into additional clusters, so the services in a cluster will be more similar with each other. Furthermore, neighbours of a better (target) service are selected from the clusters. Therefore, these neighbours may be more similar to the better web service. Figure 2 shows our proposed system achieves minimum Mean Absolute Error in recommending the web services.

3.3 Comparative Study

Figure 3 illustrates the computation time comparison of our proposed SSC based RS approach, ClubCF and IbCF. From the below figure, our Semantic Similarity Calculation based Recommendation System spends less calculation time than other existing mechanisms. Since the number of services in a cluster is less than the total number of services, the time of ranking resemblance computation among all pair of services will be reduced.

3.4 Precision and Recall Comparison

Assume a Data Base includes 200 records for a specific topic. In this we conducted a search on a topic and 190 records were retrieved. As a result, 185 records were relevant out of 190 retrieved records. Based on this the Precision and Recall is calculated in Table 2. Precision = (X /(X +Z))*100 (where X - amount of relevant records, Z - amount of irrelevant records). Recall = (X/(X+Y)) * 100 (where X - amount of relevant records, Y - amount of relevant records not considered).

Table 2. Precision and recall percentage

Precision		Recall	
Technique	Percentage	Technique	Percentage
LSM	60	LSM	60
GA	70	GA	70
VSM	90	VSM	88
SSC based SR	94.87	SSC based SR	92.5

The Figure 3 and Figure 4 explain about the precission and recall comparisions for existing techniques and proposed system.

Various algorithms consider in existing system those are GA (Genetic Algorithm), VSM (Vector Space Model) and LMS (Least Mean Square). Since LMS is an iterative algorithm which has largest number of iterations and it consumes more time. Also it identifies only minimum square error. Another existing technique which is genetic algorithm has more number of iterations since which requires more processing time compared to LMS. Compared to other two existing algorithms vector space

model is better than genetic algorithm and LMS because of its simple iteration and efficient for indexing data. Search results on keyword basis are very effective based on vector space model, but fails to obtain into account. Proposed technique improve parallel processing and reduce the complexity, the files are transformed into chunks. To improve the performance of our SSC based RS, web service information is structured in hierarchical format.

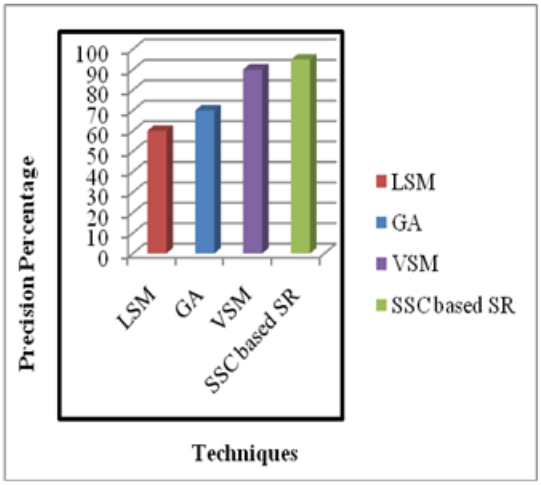

Figure 3. Precision comparison of proposed system with existing techniques.

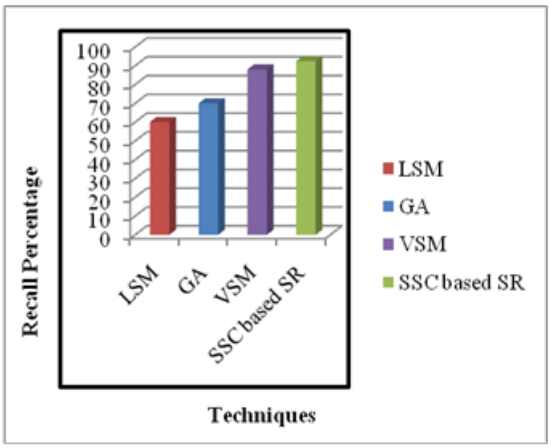

Figure 4. Recall comparison of proposed system with existing techniques.

3.5 Performance Measure (F) Comparison

F = precision * recall / (precision + recall)

Table 3. Performance measure percentage

Technique	Percentage
LSM	27.85
GA	31.26
VSM	44.49
SSC based SR	46.86

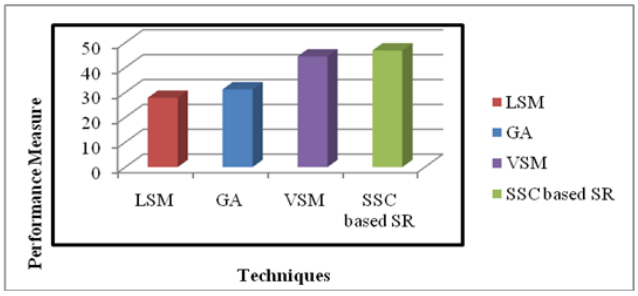

Figure 5. Performance measure comparison of proposed system with existing techniques.

4. Conclusion

In this research work a Service Recommendation using SSC based RS approach is presented to the requested user. In this the semantic resemblance among the big data web applications is computed. To present the suggested service list to the requested user a Semantic Based k-means Clustering Algorithm is used. This provides a personalized service recommendation list and recommends the most suitable services to the users. The proposed algorithm calculates the semantic similarity using the common and difference terms in the web application that increases the forecast accuracy and excellence of Service. In the Hadoop environment the scalability and efficiency in Big Data is improved based on this method. Experimental results Table 3 and Figure 5 shows that our proposed algorithm provides a suitable recommended service compared to other existing approaches.

5. References

1. Kim KW, Park WJ, Park ST. A study on plan to improve illegal parking using big data. Indian Journal of Science and Technology. 2015 Sep; 8(21):1–5. DOI: 10.17485/ijst/2015/v8i21/78274.
2. Lynch C. Big data: how do your data grow?. Nature. 2008; 455(7209):28–9.

3. Chang F, Dean J, Ghemawat S, Hsieh WC. Bigtable: a distributed storage system for structured data. ACM Transactions on Computer Systems. 2008; 26(2):1–14.
4. Dou W, Zhang X, Liu J, Chen J. HireSome-II: towards privacy-aware cross-cloud service composition for big data applications. IEEE Transactions on Parallel and Distributed Systems. 26(2):455–466.
5. Linden G, Smith B, York J. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing. 2003; 7(1):76–80.
6. Bjelica M. Towards T V recommender system experiments with user modelling. IEEE Transactions on Consumer Electronics. 2010; 56(3):1763–69.
7. Alduan M, Alvarez F, Menendez J, Baez O. Recommender system for sport videos based on user audiovisual consumption. IEEE Transactions on Multimedia. 2013; 14(6):1546–57.
8. Chen Y, Cheng A, Hsu W. Travel recommendation by mining people attributes and travel group types from community-contributed photos. IEEE Transactions on Multimedia. 2012; 25(6):1283–95.
9. Zheng Z, Wu X, Zhang Y, Lyu M, Wang J. QoS ranking pre-diction for cloud services. IEEE Transactions on Parallel and Distributed Systems. 2013; 24(6):1213–22.
10. Raj JR. Web service discovery based on computation of semantic similarity distance and QOS normalization. Indian Journal of Computer Science and Engineering. 2012 Apr–May; 3(2):566–68.
11. Tibermacine O, Tibermacine C, Cherif F. WSSim: a Tool for the Measurement of web service interface similarity. Proceedings of the French-speaking conference on Software Architectures (CAL'13), Toulouse: France; 2013 May.
12. Kokash N. A comparison of web service interface similarity measures. Proceedings of the 2006 conference on STAIRS 2006. Third Starting AI Researchers' Symposium, Amsterdam, The Netherlands; 2006. p. 220–31.
13. Liu F, Shi Y, Yu J, Wang T, Wu J. Measuring similarity of web services based on WSDL. Proceeding of: IEEE International Conference on Web Services, ICWS 2010, Miami, Florida, USA; 2010 Jul 5–10.
14. Kohrs A, Merialdo B. Clustering for collaborative filtering applications. Proceedings of CIMCA'99. IOS Press; 1999.
15. Ungar LH, Foster DP. A formal statistical approach to collaborative filtering. Proceedings of Conference on Automated Leading and Discovery (CONALD); 1998.
16. Symeonidis P, Nanopoulos A, Papadopoulos A, Manolopoulos Y. Nearest bi-clusters collaborative filtering. WEBKDD; 2006.
17. Zhao ZD, Shang MS. User-based collaborative-filtering recommendation algorithms on Hadoop. Third International Workshop on Knowledge Discovery and Data Mining; 2010. p. 478–81.
18. Liang H, Hogan J, Xu Y. Parallel user profiling based on folksonomy for large scaled recommender systems: an implementation of cascading mapreduce. Proceedings of the IEEE International Conference on Data Mining Workshops; 2010. p. 156–61.
19. Suchithra M, Ramakrishnan M. A survey on different web service discovery techniques. Indian Journal of Science and Technology. 2015 Jul; 8(15):1–5. Doi no:10.17485/ijst/2015/v8i15/70773.
20. Abramowicz W, Haniewicz K, Kaczmarek M, Zyskowski D. Architecture for web services filtering and clustering. Proceedings 2nd International Conference on Internet and Web Applications and Services; 2007.
21. Li M, Yang Y. Efficient clustering index for semantic Web service based on user preference. Proceedings International Conference on Computer Science and Information Processing; 2012. p. 291–94.
22. Zheng Q, Wang Y. The application of semi-supervised clustering in web services composition. Advances in Intelligent and Soft Computing. 2012; 169:683–88.
23. Chen L, Hu L, Zheng Z, Wu J, Yin J, Li Y, Deng S. WTCluster: Utilizing tags for web services clustering. Lecture Notes in Computer Science. 2011; 7084:204–18.
24. Skoutas D, Sacharidis D, Simitsis A, Sellis T. Ranking and clustering web services using multicriteria dominance relationships. IEEE Transactions on Services Computing. 2010; 3(3):163–77.