Best Keyword Set Recommendations For Building Service-Based Systems

Dr. Gajula Ramesh, Dr.J.Somasekar, Dr. Karanam Madhavi, Dr. Gandikota Ramu

Abstract: Software development has witnessed tremendous changes. One of the important paradigm shifts it has undergone is reusable services that get rid of the need for reinventing the wheel. Thus it could achieve productivity and service orientation rather having multiple disjointed applications. This was made possible due to technology innovations and emergence of distributed computing technologies like Web Services. With integrated interoperable services reusability is promoted in unprecedented fashion. In this context Service Oriented Architecture (SOA) has assumed significance beyond imagination. With third party services available and accessible in public domain, new service oriented application development became easier on top of the state-of-the-art services already available. It has resulted in building useful Service Based Systems (SBSs). Such systems are made up of multiple services integrated with Single Sign On (SSO) service on top of SOA. However, building robust SBS is very challenging. The literature found many contributions but still a framework that renders desired services in service discovery and composition to have robust SBSs is the need of the hour. Towards this end, in this paper a framework is designed to have multiple phases to have different activities like pre-processing, building SVM classifier to predict class labels like SBS with anti-patterns and SBS with no anti-patterns, generating keyword set recommendations for effective service discovery and composition of SBSs. Collaborative filtering based algorithm under multi-user and multi-item (SBSs) settings. A prototype application is built to demonstrate proof of the concept. The experimental results and evaluation revealed that the proposed method outperforms state of the art methods.

Index Terms: Keyword search, Service Based System (SBS), web service, service discovery, service composition, SVM classifier.

1 INTRODUCTION

SERVICE Oriented Architecture (SOA) based systems revolutionized the way customer centric applications are built. Instead of having multiple applications, the service oriented approach targets services to end users by simplifying and integrating applications into services. Distributed computing technologies paved way for this. With SOA many complex chain of applications are transformed into seamless services provided to end users. For instance in tourism domain, many applications used for reservations, hotel bookings, cab bookings etc. are brought into a single service where a Single Sign On (SSO) in an SOA application can take care of everything and users are provided required service. This service based systems integrate all complexities of real world into simple services that are automated. The SBS is realized using web services technology. Web service is a program that is interoperable and can run in remote servers and invoked dynamically. They are reusable components. Based on the Software Engineering (SE) principles, such applications avoid reinventing wheel and promote reusability. SOA takes the reusability to next level by allowing integration of similar applications into a seamless service rendering to end users sans of time and geographical restrictions. In the literature many contributions are found in this area. They are broadly divided into two related activities. They are known as service discovery and service composition. The service discovery is widely used as explored in [4], [14], [16] and [20]. The service composition is investigated with different approaches in [1], [6], [8], [11], [13], [16], [19] and [23]. Monitoring strategies for SBS are also found in the literature as in [15], [21] while some simulation studies are found on the service discovery and composition as investigated in [22], [23] and [25]. A common thread in all these researches is that there is need for improvement with a comprehensive framework that takes care of search keyword recommendations besides supporting service discovery and composition. This is the motivation behind the research in this paper. Our contributions in this paper are as follows.

- A framework is proposed for service discovery and composition of SBS based on the generated keyword set recommendations that are used for best SBS creation.
- An SVM based classifier is built to predict SBS as anti-pattern or not anti-pattern based on the given training set.
- An algorithm is proposed to have collaborative filtering in a multi-user and multi-SBS (items) environment for keyword set recommendations.
- A prototype application is built using JDK and Eclipse IDE in order to demonstrate proof of the concept. The experimental results revealed the usefulness of the prototype which realizes the proposed framework.

The remainder of the paper is structured as follows. Section 2 provides review of literature on the web service anti-patterns and business process anti-patterns. Section 3 defined the problem considered. Section 4 presents the proposed system. Section 5 presents implementation details. Section 6 presents experimental results. Section 7 concludes the paper provides directions for future work.

2 RELATED WORK

This section provides review of literature on automatic generation of SBS systems. Skyline services are explored in [1] for web service composition. Quality of Service (QoS) is considered while composing services. Similarly in [2] new generation web based application is automatically built with the concept of service mashups. Often they are also known as situational web applications that are dynamically built. Service Oriented Architecture (SOA) is widely used to have service oriented applications. SOA based systems dynamically adapted at runtime as studied in [3]. In [4], automatic service discovery is made using a method known as probabilistic match making. The notion of keyword proximity is used for better search criteria and complex data graphs in order to

- *Dr. Gajula Ramesh is currently working as Associate Professor in CSE Department, GRIET, Hyderabad, India PH-9440862112. E-mail: ramesh680@gmail.com*
- *Dr. J. Somasekhar is currently working as Professor in CSE Department, Gopalan College of Engineering and Management, Banglore, Karnataka PH-7348842635. E-mail: jsomasekar@gmail.com*
- *Dr. Karanam Madhavi is currently working as Professor in CSE Department, GRIET, Hyderabad, India PH-9849364116. E-mail: bmadhaviranjan@yahoo.com*
- *Dr. Gandikaota Ramu is currently working as Professor in CSE Department, Institute of Aeronautical Engineering, Hyderabad, PH-9703962233. E-mail: g.ramucse@gmail.com*

have dynamic applications in [5]. The service composition process is investigated in cloud based systems in [6] where network-aware service composition is the crucial contribution. Of it is important to discover web services before composing new applications with required services. In [8] such investigation is made using semantics based service discovery and composition. Sometimes, it is possible to have query expansion to have better search criteria to discover web services from sources of Internet [7]. Service recommendations also play an important role in finding better services. It is explored in [9] to have negative-service aware recommendations to get rid of issues in the composition. When multiple services work together, it becomes a Service Based System (SBS). As explored in [10] BPEL process with self supervision are examples for SBS oriented applications. As mentioned earlier, situational web applications are dynamically created in [11] with SOA approach [21] and data-driven approach. There are many contributions towards investigating and composing web services from World Wide Web (WWW). They include [12], [13] which focused on large scale service networks, [14] where keyword search is employed, [16] with wishful search and top-k nearest search method in [20]. Monitoring of SBS after composition is the study focused in [15] while the concept of functionally diverse services is explored in [18] for robust service composition. Adaptive service composition is the study found in [19] while simulation approaches are studied in [22], [23] and [25]. The review of literature found that there is need for further research in terms of developing a framework that can help in discovering and composing SBS with better keyword set recommendations. This gap is filled in this paper. The following sections throw light on this.

3 Problem Definition

Service Based Systems (SBSs) play vital role in rendering services to end users. They can get all related services with a single application instead of moving to different individual applications. When SBS is constructed dynamically, there is the problem of service discovery and service composition. From the literature it is understood that the service composition needs search keywords. When the users are not equipped with good keywords, it results in SBS composition that does not provide accurate solution. Therefore it is important to suggest keywords also to end users so as to help them to provide ideal keywords for search based SBS composition. This is the challenging problem considered in this paper.

4 Proposed Framework

This section presents the proposed framework which is discussed with multiple phases in it. In the first phase, the framework focuses on taking a SBS dataset and predicts class labels for the same. It is based on the SVM classifier. SVM makes binary classification dividing the testing data into SBS with anti-patterns and SBS with no anti-patterns. The Figure 1 shows how the classification is made. However, just classification is not useful unless, users get best keyword sets for enhancing capability of discovering and creating or composing best SBS that provide integrated services to end users.

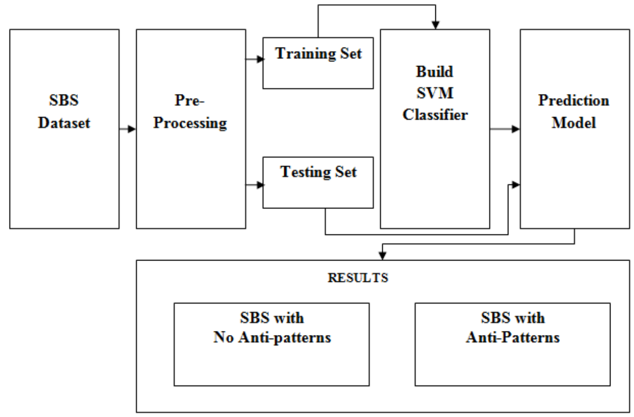

Fig. 1: Proposed framework for predicting SBS anti-patterns

As shown in Fig.1, it is observed that SBS dataset collected from [17] is subjected to pre-processing. The pre-processing step results in generation of two datasets. They are known as training and testing datasets. The training set is used to train the SVM classifier while the testing set is used to predict class labels. The results of SVM are a model that has gained or learned knowledge with the proposed supervised learning approach. With the knowledge model in hand, now it is able to predict SBS instances to be either SBS with anti-patterns or SB with no anti-patterns. Once the SBS with no-anti patterns are predicted, they are used for further research. The main problem identified in SBS generation is that users are not aware of the keywords that are used to discover highly related services to form SBS. To overcome this problem, we extended our framework as shown in Figure 2. The extension part of the framework is used to generate keyword sets that are highly useful to have best service discovery and composition of services. Before that, here is the functionality of SVM described. The dataset D is represented as in Eq. (1).

$$D = \{(\sum_{i=1}^{n} x_i, y_i | x_i \in R^p, y_i \in \{-1,1\})\} \quad (1)$$

The dataset D has number of rows denoted as n. Each row has different elements denoted as element $((x_i, y_i)$. In this context, the pair such as x_i and y_i are denoted as elements. Out of which a p dimensional feature vector is associated with x_i while set of class labels is associated with y_i. There needs to be a decision function that can determined the two classes. It is as in Eq. (2).

$$f(x) = sign(w^T.x + b) \quad (2)$$

Where 'w' denotes a normal vector while 'b' denotes the hyper plane's intercept. In other words, the position of hyper plane is represented by both w and b. The optimal separation is possible with the concept of hyper plane. The SVM classification problem can be mathematically specified as in Eq. (3).

$$\text{Minimize } \frac{1}{2} w^T.w \quad (3)$$

Subjected to

$$1 - y_i(w^T.x_i + b \le 0) \quad for\ any\ \ i = 1,2, \ldots n$$

Then it is possible to have an optimal hyper plane by considering the solution to problem aforementioned with linearly constrained quadratic equation for optimization. After solving the equation, the final solution is as in Eq. (4).

$$f(x) = \text{sign}\left(\sum_{i,i=1}^{n} \alpha_i y_i x_i x_i + b\right) \quad (4)$$

Where Lagrange multiplier is denoted as 'α_i'. This is the linear case approach. With respect to non-linear approach to classification, the SVM algorithm performs transformation of original data to get the same in higher dimension. Afterwards, the SVM seeks linear optimal solution by separating decision boundary or hyper plane in a new dimension. Thus the decision function is revised as in Eq. (5).

$$f(x) = \text{sign}\left(\sum_{i,i=1}^{n} \alpha_i y_i \emptyset(x_i).\emptyset(x_i) + b\right) \quad (5)$$

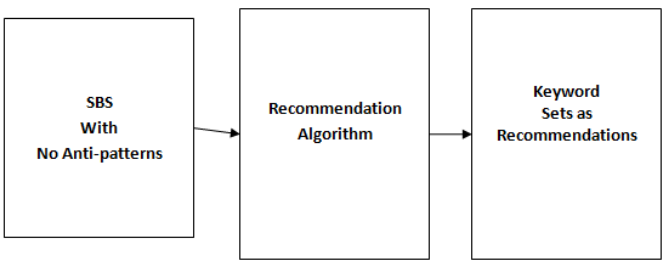

Fig.2: Search keyword extraction

Here there is involvement of non-linear mapping function denoted as $\emptyset$. SVM is used as classifier that predicts two class labels which as SBS with anti-patterns and SBS without anti-patterns. After prediction of class label SBM with no anti-patterns, the instances of this class are further used for recommendations with keyword sets. As presented in Fig.2, it is evident that the search keyword extraction procedure is illustrated. The SBS with no anti-patters is taken as input. In a setting of multiple users preferring related SBS, a recommendation algorithm is proposed based on collaborative filtering concept. The algorithm returns keyword sets as recommendations. The recommendation algorithm is described in Algorithm-1. In a multi-user multi-SBS setting the algorithm is based on collaborative filtering method.

Algorithm: Collaborative Filtering for SBS Recommendations

Inputs: Users U, SBSs S

Output: SBS Recommendations OS and keyword sets

1. Initialize a Users and SBSs matrix US
2. Initialize output SBS recommendations vector
3. Initialize keyword sets KS
4. For each user in U
5. For each sbs in S
6. Add rating value to US
7. End For
8. End For
9. Use cosine similarity for generating SBS to SBS similarity matrix
10. Filter US
11. For each item in US
12. Extract keywords from item
13. Add keyword set to KS
14. End For
15. Return KS

The algorithm performs collaborative filtering from multiple users and their SBS ratings. The algorithm considers SBSs chosen or preferred by previous users in order to make keyword set recommendations for best SBS generation. The results are then subjected to keywords extraction to have search keyword recommendations for end users to have search based SBS composition that meets quality criteria.

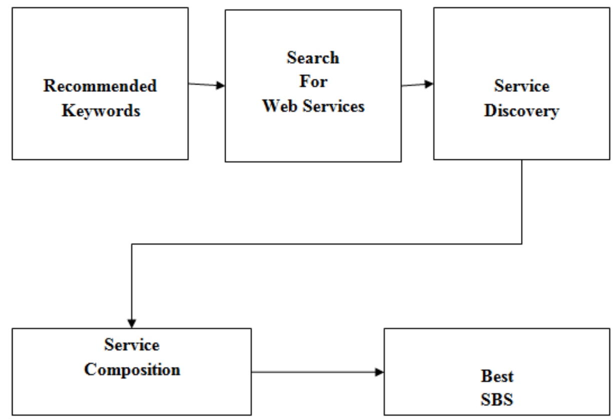

Fig. 3: Generation of best SBS

As presented in Figure 3, it is evident that the recommended keywords are used to searching web services online and discover them live. Then the service composition is made to have best SBS. The reason behind user recommendations for the keyword sets is that users are not aware of related keywords to have best search for SBS. Once the SBS are discovered and composed, their quality and integrity will be better than the keywords directly used by the users. The recommended keyword sets improve performance of making SBS that are very useful

5 Dataset and Experimental Processing

The proposed frameworks is realised with a prototype application having Graphical User Interface (GUI). The application is built using JDK and Eclipse IDE. It is used to demonstrate proof of the concept. The dataset obtained from [17] is employed to have SVM classification and then to have further recommendation based keyword set usage for service discovery and composition. The process of generating SBS is automated with graphical tool and the underlying algorithms for recommendations of keywords and the other framework mechanisms illustrated in Figure 1, Figure 2 and Figure 3. The implementation point of view and the flow of operations are presented in Figure 4.

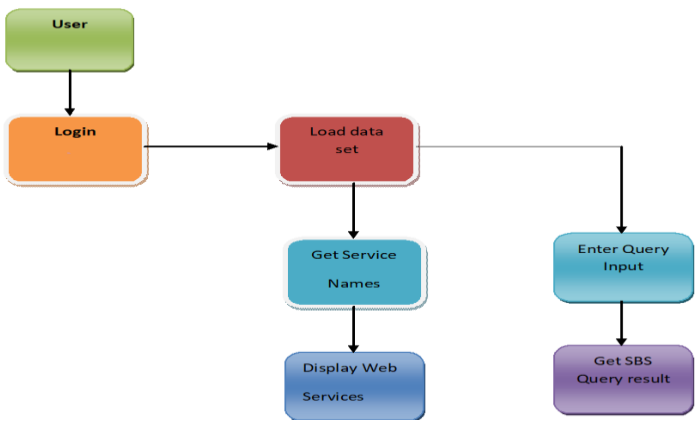

***Fig. 4** Flow of operations in involved in the proposed framework*

The flow of operations that are implemented in the proposed framework and realized in terms of different parts is provided. The application makes use of dataset related to SBS and it is loaded into a collection data structure before it is used for processing. The initial pre-processing generates training and testing datasets. After generating classification results by SVM classifier (as shown in Figure 1), there is service discovery and service composition procedures based on the recommended keyword sets.

5.1 Datasets Used

SBS datasets that are having mix of anti-patterns and without anti-patterns are used as input to the proposed framework. The dataset is collected from [17]. Fig. 5 presents some part of the dataset used for experiments initially. The dataset contains SBS of different categories like travel, weather, shipping, service, finance and so on.

Financial (94)
http://developer.ebay.com/webservices/finding/latest/FindingService.wsdl
http://www.xignite.com/xMaster.asmx?WSDL
http://www.xignite.com/xIndexComponents.asmx?WSDL
…
Science (34)
http://www.ebi.ac.uk/webservices/whatizit/ws?wsdl
http://www.pharmaceutical-bioinformatics.de/prolific/soap/prolific.wsdl
http://mrs.cmbi.ru.nl/mrsws/search/wsdl
…
Shipping (38)
http://webservices.linjegods.no/PublicMethodes.asmx?WSDL
http://developer.stamps.com/developer/downloads/files/Stamps.com_SWSIM_v26.wsdl
http://ws.epostcode.com/uk/postcodeservices09.asmx?WSDL
…
Travel (65)
https://api.flightstats.com/flex/flightstatus/docs/v2/lts/soap/flightsNearService.wsdl
http://api.whl.travel/soap?act=wsdl
http://www.orchestration.com.br/TravelManagementService.svc?wsdl
…
Weather (42)
http://eil.cs.txstate.edu/ServiceXplorer/wsdl_files/service86.Specialist.wsdl
http://river.sdsc.edu/wateroneflow/SNOTEL/cuahsi_1_0.asmx?WSDL
http://eil.cs.txstate.edu/ServiceXplorer/wsdl_files/2446_Czech_Republic_WeatherWS.wsdl
…

***Fig. 5:** Dataset details*

As can be seen in Figure 4, it is evident that there are different SBS hat is used to have experiments. The SVM classifier used in the framework distinguished SBS anti-patterns from SBS with no anti-patterns. As shown in Fig.6, the prototype application has all features right from loading dataset to performing all activities. It supports SVM classification, dynamic discovery of web services and SBS generation based on the recommended relevant keyword sets

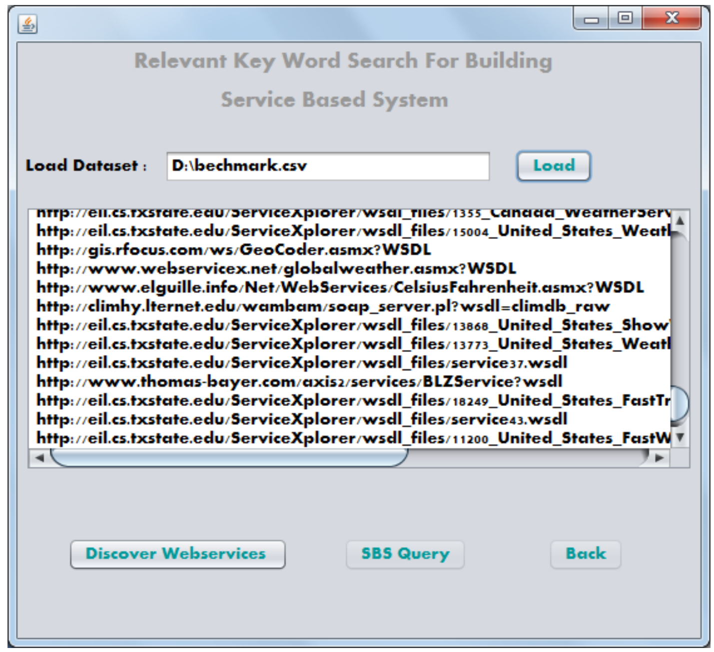

***Figure 6**: The prototype application used in the experiments*

6 Experimental Results

Experiments made with the proposed framework realized in the form of the prototype application are described here. The dataset obtained from [17] is used to evaluate the proposed framework. The performance of the application is measures in terms of computation time and the utility of the framework that generates SBS with service discovery and service composition based on generated keyword sets used for search. Actually collaborative filtering based algorithm proposed in this paper generates keyword set recommendations as the users are not aware of relevant keywords.

***Table 1**: Performance of the proposed system with different keyword distances*

Keyword Distance	Computation Time(ms)			
	KS3 Normal	KS3 Constraint	KS3 Optimal	Proposed
1	0	0	0	0
2	1000	1000	1000	800
3	1000	1000	7000	800
4	1000	1000	8000	800
5	1000	1000	10000	800
6	2000	2000	13000	1500
7	2000	2000	17000	1600
8	2500	2500	18000	2000
9	3000	4000	22000	2500
10	4000	5000	30000	3500

As can be seen in Table 1, the computation time is the main observation recorded for various keyword distances. The observations are against different methods like the proposed one and other state of the art approaches found in literature like KS3 optimal, KS3 constraint and KS3 normal.

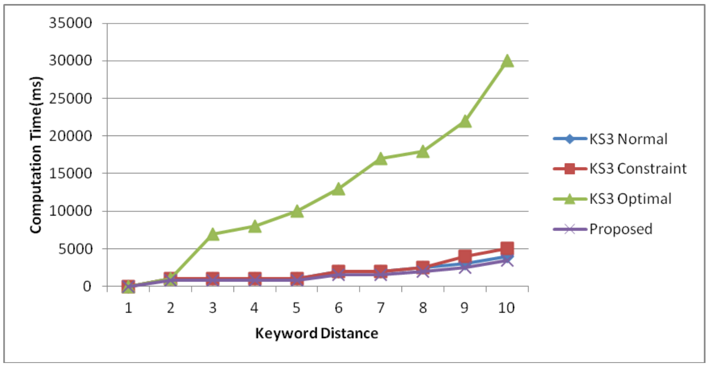

Figure 7: *Performance comparison against different keyword distances*

Table 2: *Performance of the proposed system with number of keywords in query*

Number of keywords in Query	Computation Time(ms)			
	KS3 Normal	KS3 Constraint	KS3 Optimal	Proposed
1	0	0	0	0
2	4000	4000	18000	3000
3	6000	12000	28000	5000
4	12000	25000	45000	10000

The results shown in Figure 7 reveal the performance differences between the state of the art approaches and the proposed framework. Keyword distances are presented in horizontal axis while the vertical axis shows the performance in terms of computational time. The results showed that the computational time is influenced by key word distances and the proposed approach outperforms other approaches. As presented in Table 2, the observations on computational time are recorded against different number of keywords and different state of the art methods besides the proposed method in this paper.

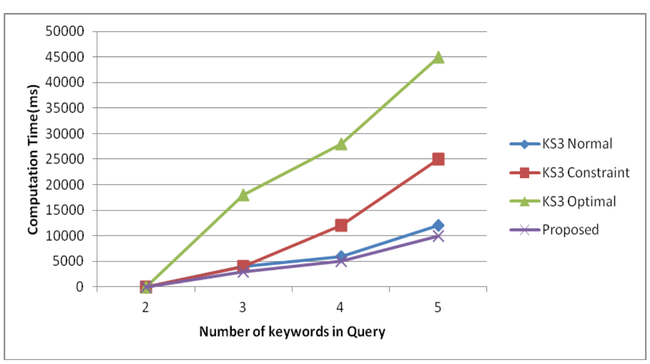

Fig 8. *Performance comparison against different number of keywords in query*

As shown in Figure 8, the results of the experiments reveal the performance of the proposed method over state of the art approaches. The number of keywords present in given query is shown in horizontal axis while the vertical axis shows the performance in terms of computational time. The number of keywords in the query has their impact on the performance and the proposed system is found outperforming other methods.

Number of Quality Constraints	Computation Time (ms)			
	KS3 Normal	KS3 Constraint	KS3 Optimal	Proposed
1	1800	2500	2500	2200
2	1700	2300	2400	2000
3	2000	2000	2100	1800
4	2000	1600	1700	1400
5	2200	1500	1600	1300
6	1800	900	900	800
7	2000	500	500	400
8	1900	0	0	0
9	2100	0	0	0
10	2000	0	0	0

Table 3: *Performance of the proposed system with different number of quality constraints*

Table 3 presents the quality constraints used in the proposed system that influences the performance of the system. The computational time for different approaches against different number of quality constraints is presented. As found in Figure 9, it is clear that the number of quality constraints do have their impact on the performance. The results also reveal that the proposed system outperforms other state of the art methods compared. The proposed system shows less computational time.

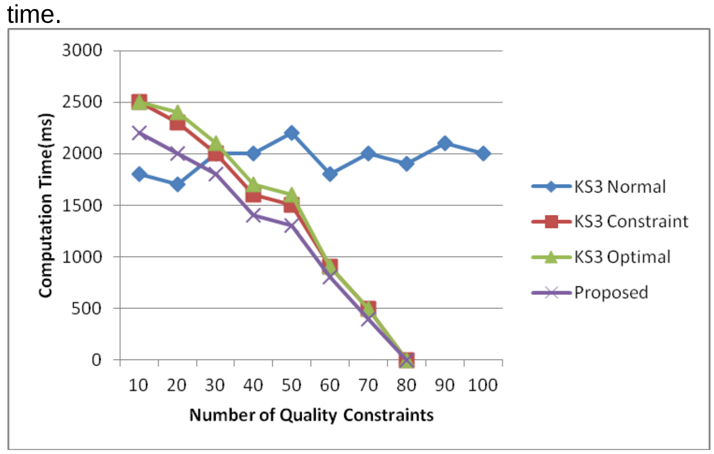

Fig 9: *Performance comparison against different number of quality constraints*

7 Conclusions and Future Scope

Building SBSs dynamically by discovering relevant services is very challenging phenomenon. The rationale behind this is that users who perform search based discovery do not know exactly what are the most relevant keywords related to desired SBS to be built. This has limited the capability of service discovery and composition. To overcome this problem, it is important to recommend keyword set recommendations to users who involve in building SBS dynamically. Towards this end, in this paper, we proposed a framework that focuses on finding key words from SBS with no anti-patterns. Since anti-patterns defeat the very purpose of the SBS, an SVM classifier is built to distinguish them from SBS with no anti-patterns. Afterwards, the framework makes use of SBS with no anti-patterns to generate relevant keyword sets as recommendations. Collaborative filtering based algorithm is proposed to achieve this. The algorithm takes different users and different rated SBSs for collaborative filtering. The recommended keyword sets based on the kind of SBS to be built is used to discover and compose services. The framework is realized with a prototype application built using JDK and Eclipse. The experimental results are encouraging as the proposed method outperforms other state of the art methods. In future, we try to use ensemble approach to investigate the process of building SBS dynamically. Enhancement of the framework with such method is an interesting direction for future work.

REFERENCES

[1] M. Alrifai, D. Skoutas, and T. Risse, "Selecting Skyline Services for QoS-based Web Service Composition," Proc of 19th International Conference on World Wide Web (WWW 2010), Raleigh, North Carolina, USA, pp. 11-20, 2010.

[2] D. Benslimane, S. Dustdar, and A. Sheth, "Services Mashups: The New Generation of Web Applications," IEEE Internet Computing, vol. 12, no. 5, pp. 13-15, 2008.

[3] V. Cardellini, E. Casalicchio, V. Grassi, S. Lannucci, F. Lo Presti, and R. Mirandola, "MOSES: A Framework for QoS Driven Runtime Adaptation of Service-Oriented Systems," IEEE Transactions on Software Engineering, vol. 38, no. 5, pp. 1138-1159, 2012.

[4] G. Cassar, P. Barnaghi, and K. Moessner, "Probabilistic Matchmaking Methods for Automated Service Discovery," IEEE Transactions on Services Computing (TSC), vol. 7, no. 4, pp. 654-666, 2014.

[5] K. Golenberg, B. Kimelfeld, and Y. Sagiv, "Keyword Proximity Search in Complex Data Graphs," Proc of 28th ACM SIGMOD International Conference on Management of Data (SIGMOD 2008), pp. 927-940, 2008.

[6] A. Klein, F. Ishikawa, and S. Honiden, "Towards Network-Aware Service Composition in the Cloud," Proc of 21st World Wide Web Conference (WWW 2012), Lyon, France, pp. 959-968, 2012.

[7] C. Carpineto and G. Romano, "A Survey of Automatic Query Expansion in Information Retrieval," ACM Computing Surveys, vol. 44, no. 1, pp. 1-50, 2012

[8] A. Brogi, S. Corfini, and R. Popescu, "Semantics-Based Composition-Oriented Discovery of Web Services," ACM Transactions on Internet Technology, vol. 8, no. 4, pp. 19:1-19:39, 2008.

[9] Y. Ni, Y. Fan, W. Tan, K. Huang, J. Bi, "NCSR: Negative-connection-aware service recommendation for large sparse service network", IEEE Trans. Autom. Sci. Eng., vol. 13, no. 2, pp. 579-590, Apr. 2016.

[10] L. Baresi and S. Guinea, "Self-Supervising BPEL Processes," IEEE Transactions on Software Engineering, vol. 37, no. 2, pp. 247-263, 2011.

[11] X. Liu, Y. Ma, G. Huang, J. Zhao, H. Mei, and Y. Liu, "Data-Driven Composition for Service-Oriented Situational Web Applications," IEEE Transactions on Services Computing (TSC), vol. 8, no. 1, pp. 2-16, 2015

[12] E. Al-Masri and Q. H. Mahmoud, "Investigating Web Services on the World Wide Web," Proc of 17th International Conference on World Wide Web (WWW 2008), Beijing, China, pp. 795-804, 2008.

[13] S.-C. Oh, D. Lee, and S. R. Kumara, "Effective Web Service Composition in Diverse and Large-Scale Service Networks," IEEE Transactions on Services Computing (TSC), vol. 1, no. 1, pp. 15-32, 2008.

[14] V. Hristidis and Y. Papakonstantinou, "DISCOVERY: Keyword Search in Relational Databases," Proc of 28th International Conference on Very Large Data Bases (VLDB 2002), Hong Kong, China, pp. 670-681, 2002.

[15] Q. He, J. Han, Y. Yang, H. Jin, J.-G. Schneider, and S. Versteeg, "Formulating Cost-Effective Monitoring Strategies for Services-based Systems," IEEE Transactions on Software Engineering, vol. 40, no. 5, pp. 461-482, 2014.

[16] A. V. Riabov, E. Boillet, M. D. Feblowitz, Z. Liu, and A. Ranganathan, "Wishful Search: Interactive Composition of Data Mashups," Proc of 17th International Conference on World Wide Web (WWW 2008), Beijing, China, pp. 775-784, 2008.

[17] Web services datasets. Retrieved from https://github.com/wsdream/wsdream-dataset. Accessed on 15 August 2018.

[18] F. Wagner, B. Klöpper, F. Ishikawa, and S. Honiden, "Towards Robust Service Compositions in the Context of Functionally Diverse Services," Proc of 21st International World Wide Web Conference (WWW 2012), Lyon, France, pp. 969-978, 2012.

[19] D. Ardagna and B. Pernici, "Adaptive Service Composition in Flexible Processes," IEEE Transactions on Software Engineering (TSE), vol. 33, no. 6, pp. 369-384, 2007.

[20] M. Jiang, A. W.-C. Fu, and R. C.-W. Wong, "Exact Top-k Nearest Keyword Search in Large Networks," Proc of 36th ACM SIGMOD International Conference on Management of Data (SIGMOD 2015), pp. 393-404, 2015.

[21] Erl, T. (2006). Service-oriented architecture concepts, technology and design. New York: Prentice Hall.

[22] Kim, S., Sarjoughian, H. S., & Elamvazuthi, V. (2009). DEVS-suite: a simulator for visual experimentation and behavior monitoring. In High performance computing & simulation symposium, proceedings of the spring simulation conference, 1–7 March, San Diego, CA.

[23] Hu, X., Zeigler, B. P., & Mittal, S. (2005). Variable structure in DEVS component-based modeling and simulation. Simulation, 81(2), 91–102.

[24] Lenz, G., & Moeller, T. (2003). NET: a complete development cycle. Reading: Addison-Wesley.

[25] Muqsith, M. A., Sarjoughian, H. S., Huang, D., & Yau, S. S. (2011). Simulating adaptive service-oriented software systems. Simulation, 87(11), 915–931.