Experimental Analysis for Semantic based Large Scale Service Composition using Deep Learning

Jackulin Sam Gini.A, A.Chandrasekar

Abstract: ***In Service Oriented Architecture (SOA) web services plays important role. Web services are web application components that can be published, found, and used on the Web. Also machine-to-machine communication over a network can be achieved through web services. Cloud computing and distributed computing brings lot of web services into WWW. Web service composition is the process of combing two or more web services to together to satisfy the user requirements. Tremendous increase in the number of services and the complexity in user requirement specification make web service composition as challenging task. The automated service composition is a technique in which Web Service Composition can be done automatically with minimal or no human intervention. In this paper we propose a approach of web service composition methods for large scale environment by considering the QoS Parameters. We have used stacked autoencoders to learn features of web services. Recurrent Neural Network (RNN) leverages uses the learned features to predict the new composition. Experiment results show the efficiency and scalability. Use of deep learning algorithm in web service composition, leads to high success rate and less computational cost.***

Keywords: Web services, web service composition, semantic web, Deep Learning

I. INTRODUCTION

Distributed Systems can be defined as Hardware or Software components located at different locations connected through networks can communicate and coordinate by sending messages. Service Oriented Computing (SOA) is a computing model built over the features of Distributed computing. The basic building blocks of Service Oriented Computing (SOC) are Web services. Reusable, Composable, Autonomous, Statelessness and Discoverable are the most widely used principles of SOC. Web services are reusable, discoverable and modular units of applications. These units are deployed in the internet and are accessed from many web based applications or users. Web Services Description Language(WSDL)used to describe the methods, input/output parameters to access the methods and message structure web services. WSDL files are written in XML format. Universal Descriptors Discovery Integration (UDDI) is an XML-based registry.

For worldwide use of published web services should be registered in this UDDI. This helps the companies or users to discover the web services and to create interoperable transaction based applications. The web services of these Applications interact among themselves by sending request and response messages. Simple Object Access Protocol (SOAP) provides the standards for these communications.

Everything as a Service (Platform as a Service-PaaS, Software as a Service -SaaS and Infrastructure as a Service - IaaS) forms the basis for the emerging computing models Internet of Things (IoT), Big data which operates with clouds. These bring large number of web services into the internet. One of the great thing about the web services are heterogeneity which means services developed for different purposes developed using different technologies interact and work together. This leads to the service composition. So when the single service doesn't satisfy the user's or application's complex requirement, the composite service can be used. Figure 1 explains the web service composition scenario. The end user gives the i/p-c1,i/p-c2 and ip-c3 as input parameters for his requirement and expect op-cas output parameter. In real world there is no single web service which does this. So the atomic services S1, S2, S3 and S4 are combined to form a composite webservice.

Web service composition can be broadly classified in two categories, static or dynamic. In static compositions, the work flows are defined by the developers at the time of design. In dynamic web service composition, the work flow logics and the participant services are selected at the run time. This works without or less human intervention.

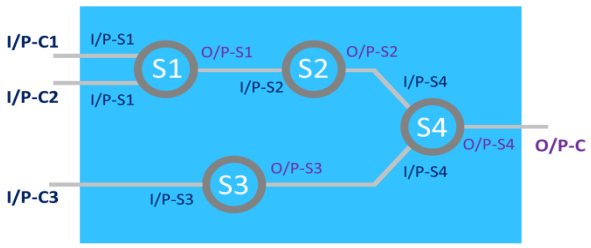

Fig. 1 Web Service Composition

When there exists multiple web services for the same functional requirements, then non functional properties are considered for service selection.

The following reasons makes the web service composition as a important research topic: Large number of web services available in the repository, Dynamically discovering the

web services using WSDL, Using Semantic methods to find the web services, Finding the optimal web

service using QoS parameters and work flow

Revised Manuscript Received on August 05, 2019

Jackulin Sam Gini.A, Research Scholar, Sathyabama Institute of Science and Technology, Chennai-600 119, India

A.Chandrasekar, Professor& Head, Department of CSE, St.Joseph's College of Engineering, Old Mamallapuram Road, Chennai-600 119, India

formation for the selected web services. Many research works are proposed based on Artificial Intelligence (AI), Fuzzy Logic, Process algebra, genetic algorithm, Graph based algorithm and Rule based planning.

In this paper we propose a semantic based automatic service composition using Deep learning algorithm. For learning the features of web services we are using the recent deep learning technique called, use Stacked Auto Encoders (SAE)[1]

II. RELATED WORK

In this section we review some of the works which deals with semantic web service composition. Yu Lie et al[2] proposed a web service composition method, time based learning algorithm for Web Service Composition(WSC). They have used POMDP for the dynamic WSC. They have measured the execution time. Comparison of execution time of various methods like Monte-Carlo method, TD-Learning algorithm and Q-learning methods of Reinforcement Learning was made. The function used in this paper that is based on time is applied with the use of a single agent system. At each time which service should be selected and which should be returned for integration is selected by the Time-Learning algorithm.

Hamza et al[3] proposes a approach for web service interactions using deep learning. They have used trained autoencoders for learning web service features. And deep forward neural network is used to predict the new compositions.

Qi Gu et al[4] proposes an approach for predicting a set of alternate services during composite development. Discourse parseris used to segment text description of mashups. The relationships between the obtained text segments are used with LSI (Latent Semantic Indexing) to model semantic relationships between services. The recommendation consists of a set of services whose relationships are consistent with the relationships among sentence blocks of a text description of a desired mashup. This approach is limited to comparing service text description. Leveraging other service features such as service categories, security information, and request/response formats improves recommendation precision. In addition, considering the history of service interactions may improve recommendation.

Lei Yu et al.[7] proposes a Q-learning algorithm for cloud environment to find the composite web service. In Q-Learning technique to learn the optimal policy they have used observed reward technique. State Transition function and reward function are not required in Q-Learning. This is the main advantage here. This method concentrates on predicting web services based on the uncertain QoS parameters. They have introduced the concept of supply chain management.

Joel et al.[9] proposes web service composition method using Markov Decision Process (MDP). It considered the QoSParameters for service composition. In real world QoSParameters are not provided accurately. In this work they have used multi agent concept. A software agent has all the web service detail. When the user request comes this agent call the Q-learning algorithm. Based on the QoS Parameters, atomic services are selected and using MDP the composite web service will be returned to the user.

Yang Yu et al.[10] proposed a service composition based on Partner First algorithm. This algorithm combines path planning with service composition, presented based on the social network. This algorithm has three stages look ahead search, candidate service filtering and path planning. Simulation experiments he conducted shows Partner First algorithm outperforms 10 times as the traditional composition methods.

We propose a service composition approach based on semantics using deep learning. For my knowledge this is the first work in this kind using deep learning for service composition.

III. SYSTEM IMPLEMENTATION

System Architecture

In our proposed work there are two phases, Training Phase and composition phase. The Figure 2 explains the architecture of the proposed work. From web service repository the available composite services and atomic services are fed as input. The WSDL files have the input parameters, output parameters, methods and messages of web services in XML format. This information is parsed using XML parser. The parsed information is given as input to the training phase.

Training phase use SAEs to learn the web service features. The features of web services are given as vectors are given as the output in this phase. In the second phase knowledge acquisition module and service composition modules are there. Here we use RNNs for knowledge gathering. When the user gives the query for his requirement, the possible service compositions are predicted based on the gained knowledge.

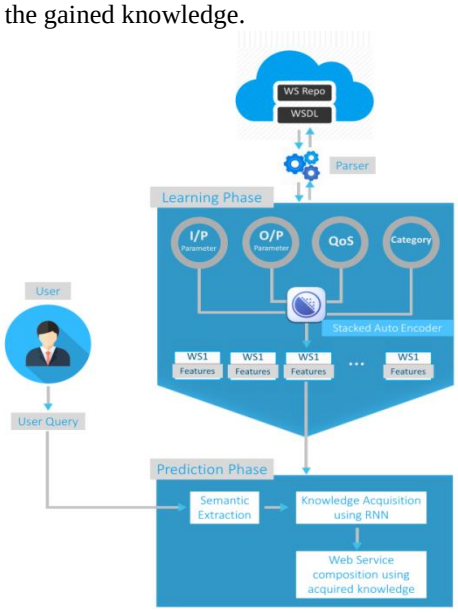

Fig. 2 System Architecture

Training Phase

In this phase to learn the features of web services we propose an unsupervised learning method. Here we use stacked auto encoders.

SAEs are the part of neural network. Using Back Propagation technique they recreate the input. An auto encoder has two parts, an encoder and a decoder. The encoder is used to read the input and changes it to a compressed representation. The decoder part will read the compressed representation and form the input from it.

Multiple encoders that are capable of compressing to smaller form can be stacked in sequence. The decoders can be stacked in opposite directions. This model forms the Stacked Autoencoders. In convolutional neural networks, convolution and pooling are the two operations that are done to capture hierarchical input structure. Similarly in SAEs the layers ordered in sequence captures the input structure.

Let I denote the input feature and O denote the learned output feature then Iis encoded to Oas ,

$$O = f(I) = A(w_1 I + b_1)$$

Where w_1 and b_1 are the weight and bias function respectively. A is the activation function Sigmoid. The encode feature O is decoded to I using a decoder function,

$$I^{:} = g(O) = A(w_2 O + b_2)$$

Where w_2 and b_2 denote the weight and bias respectively.

A sequence of words is used as input to train SAEs. Parameters of each auto encoder are learned independently using Greedy Wise layer[5]. To learn the parameters of stacked autoencoders at the same time we using the technique, Fine Tuning.[8]For fine tuning model we use back propagation and Stochastic gradient descent(SGD) algorithm.

Composition Phase

In this phase we propose a recurrent neural network (RNN). RNN is an Artificial Intelligence technique. RNN can be viewed as graph which consists of nodes and links.RNN has s single input layer, multiple hidden layers and an output layer. The nodes are RNN cells. All the cells in sequence layers perform the same operations.

The Features of web services from the training phase is passed as the input to the RNN. The Knowledge acquisition process gains knowledge about the various possibilities of composition.

IV. RESULTS AND DISCUSSION

Experiments and analysis

We aim to experiment the accuracy and scalability of the service composition. We conduct our experiments on wondows10 (64bit) system. We have used Google's Tensor flow and Keras tools for the implementing stacked autoencoders and RNN.

The composite services and atomic services are given as input. Atomic services from QWS [11] data set which contains around 2500 web services is used for the experiments. Our crawler program selects around 100 real time composite services. These composite web services are used as training data set for knowledge acquisition. Using acquired knowledge, the composite web services are dynamically combined according to the user requirement.

Performance Evaluation

We propose a deeplearning algorithm for large scale web service composition. After conducting experiments, we show this method provides composite web services based on QoS. We have compared the results of the proposed work with literature [8]. Success rate of compositions and computation cost are used to compare the effectiveness of our work.

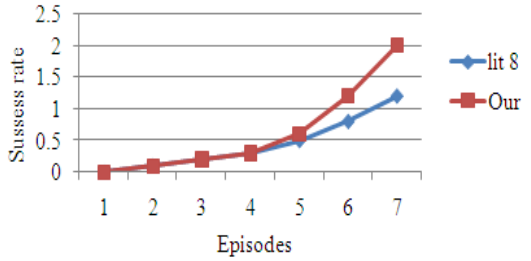

Fig. 3 Success Rate

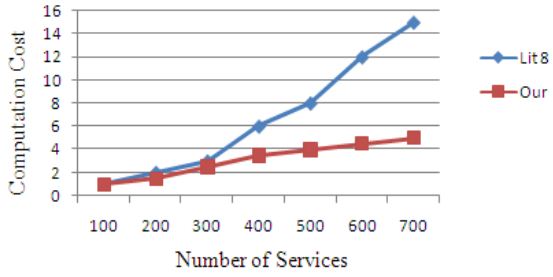

Fig. 4 Computation Cost

At the beginning policy is not optimal, but in the subsequent learning cycles the policy is constantly optimized and at one point it will converge. At the convergence point our policy can provide best service composition. Each learning cycle is called as Episode. Figure 3 shows the comparison of Success Rate. Figure 4 shows the comparison of Computation Cost in seconds. Success Rate can be defined as the ratio of number of times the request for the composition is succeeded to the total number of attempts. Computation cost is measured in terms of computation time. Our work proves that use of deep learning algorithm in service composition greatly improves the computation cost and success rate.

One of the limitation of our work is for simplicity we doesn't consider non functional requirements. In future our approach can be extended, to select the services based on the QoS Parameters.

REFERENCES

1. Alex Krizhevsky and Geoffrey E. Hinton. 2011. Using very deep autoencoders for content-based image retrieval. In ESANN 2011, 19th European Symposium on Artificial Neural Networks, Bruges, Belgium, April 27-29, 2011, Proceedings.
2. Yu Lei, Zhou Jiantao, Wei Fengqi, Gao Yongqiang and Yang Bo, "Web Service Composition Based on Reinforcement Learning", 2015

IEEE International Conferenceon Web Services, DOI 10.1109.

3. Hamza Labbaci, BrahimMedjahed, Faisal Binzagr, YoucefAklouf "A Deep Learning Approach forWeb Service Interactions",Proceedings of the International Conference on Web Intelligence, Pages 848-854
4. Qi Gu, Jian Cao, and Qianyang Peng. 2016. Service Package Recommendation for Mashup Creation via Mashup Textual Description Mining. In IEEE International Conference on Web Services, ICWS 2016, San Francisco, CA, USA, June 27 - July 2,2016. 452–459.
5. YoshuaBengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2006. Greedy Layer-Wise Training of Deep Networks. In Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December4-7, 2006. 153–160.
6. DumitruErhan, Pierre-Antoine Manzagol, YoshuaBengio, SamyBengio, and Pascal Vincent. 2009. The Difficulty of Training Deep Architectures and the Effect of Unsupervised Pre-Training.. In AISTATS, Vol. 5. 153–160.
7. Lei Yu1, Wang Zhili1, MengLingli, Wang Jiang , Luoming Meng1, Qiu Xue-song1, "Adaptive Web Services Composition Using Q-learning in Cloud", 2013 IEEE Ninth World Congress on Services
8. Hongbing Wang(B), Mingzhu Gu1, Qi Yu, HuanhuanFei, JiajieLiand Yong Tao1, "Large-Scale and Adaptive Service CompositionUsing Deep Reinforcement", International Conference on Service-Oriented Computing ICSOC 2017: Service-Oriented Computing pp 383-391
9. Joel Christian, Mohammed Hussain Bohara, "Multi-agent Web Service Composition using Partially Observable Markov Decision Process", AICTC '16 Proceedings of the International Conference on Advances in Information Communication Technology & Computing Article No. 70.
10. Yang Yu, Jian Chen , Shangquan LinYing Wang, "A Dynamic QoS-Aware Logistics Service Composition Algorithm Based on Social Network"IEEE Transactions on Emerging Topics in Computing , volume 2,no.4,December 2014
11. Al-Masri, E., and Mahmooud, Q.H.: Investigating Web Services on the World Wide Web, 17th International Conference on World Wide Web (WWW), Beijing, April 2008, pp. ss795-804.