Utilizing Web services Networks for Web service Innovation

Shahab Mokarizadeh
Royal Institute of Technology
Stockholm, Sweden
shahabm@kth.se

Peep Küngas
University of Tartu
Tartu, Estonia
peep.kungas@ut.ee

Mihhail Matskin
Royal Institute of Technology
Stockholm, Sweden
misha@kth.se

***Abstract*—The increasing presence and adoption of Web services on the Web has promoted the significance of management of new service development for service developing sectors. The major challenge is that how to find missing but potentially valuable Web services to be developed. This problem can be divided into two sub-problems: finding missing Web services and measuring the added-value of the introduced services. This paper addresses a plausible solution to the first sub problem. Given a collection of Web services, we propose a framework for suggesting a set of candidate Web services that can be introduced to the collection. These suggested services are novel and do not present in the given collection. Our solution relies on the network structure of Web services for finding and recommending new Web services and utilizes the already observed properties of Web services networks for collective evaluation of the suggested services. The proposed solution is evaluated using 753 semantically annotated Web services. The experimental results shows that the proposed framework provides web service community with new network driven methods for finding and evaluation of new Web services.**

***Index Terms*—Web service; Web service Suggestion; Web service Network; Semantic Search**

I. Introduction

Web services technology has dramatically changed how online services are provisioned. Web services are increasingly designed as reusable components serving as catalysts for emerging service applications. The increasing presence and adoption of Web services motivated studying Web service ecosystem formation mechanisms [10], [27] as well as developing Web service recommendation and selection solutions for this ecosystem [28], [10]. These recommendation solutions aim to suggest the best choice(s) among plenty of alternatives to satisfy service consumers demands. Conversely, management of service development, in the presence of such huge quantity of Web services [10], becomes a prime challenge for service developing sectors [24]. One of the primary concerns is how to find missing but potentially valuable Web services that can be developed. This problem can be divided into two sub-problems: *finding missing Web services (and introducing them)* and *measuring the added-value of the introduced services*. While the former is related to innovating a Web service solution to a certain problem, the latter involves evaluation of the importance of the considered problem. This paper presents a framework addressing a plausible solution to the first sub-problem and leaves the second problem to our future work.

We propose a framework for finding a set of new (i.e. not already seen) Web service operations that can be introduced to a given collection of Web services. The proposed framework considers semantically annotated Web services and employs semantic similarity measures [1] and Web services networks metrics [19]. More specifically, we consider Web services networks constructed by linking Web services through matching input and output parameters of their operations. The main reason for using network metrics is that the Web services network structure allows us to envision the problem of identifying missing Web services as a link prediction problem where the predicted links are constituent components of missing Web services. Moreover, Web services networks exhibit certain network properties and share common metric values that can be utilized for evaluation purposes. While the meaningfulness of suggested Web service operation is determined based on their semantic similarity to existing ones, Web services network characteristics are used for collective evaluation of all recommended services. We evaluate the suggested solution using a collection of semantically annotated Web services.

The rest of this paper is organized as follows. Section II summarizes the fundamentals and requirements of the envisioned solution, while the suggested framework is outlined in Section III. Section IV describes our experimental settings and analyses the experimental results. Finally, Section V reviews related work, while conclusions and future work are presented in Section VI.

II. Preliminaries

In this section, first we discuss briefly what kinds of Web services can be introduced and then remark the kind which we are looking for in this work. Next, we present a unified model for building Web services networks. The latter allows utilization of network structure for Web service recommendation and collective evaluation of results as described in Section IV.

A. Web services Innovation

By a Web service, we mean kind of Web based service that brings a certain value to the end-user [5]. The interface of Web service abstracts of the service in terms of semantic representation of its input(s) and output(s). In the context of this

978-1-4799-5054-6/14 $31.00 © 2014 IEEE
DOI 10.1109/ICWS.2014.95

paper, *service innovation* phrase refers to creating a valuable service which is different than existing ones. Considering Web services ecosystem as an electronic market [17], intuitively a new service is emerging in this market in accordance with (but not limited to) one of following patterns:

1) The introduced service is conceptually similar to existing ones while it variates in QoS metrics (e.g. cost, trustworthiness) and/or the semantic description of Web service input and output parameters. For example, the currency exchange rate services offered by *Central Bank of Armenia*[1] and *Daenet GmbH*[2] mainly differ in the provided exchange rate and trustworthiness of service supplier (governmental vs. private sector) and the signature of exposed service operations. The lessons learned from research efforts on Web service similarity efforts, [6], can be exploited for innovating these sort of similar services.
2) The innovated service is functionally dependent on other services and offers an added value proposition on top of existing ones. An illustrative example is the price comparison service provided by *Price Runner*[3] which collects a price list for a given item from various Web shops by invoking the Web services offered by those Web shops. Assuming the presence of Web service interface (WSDL or REST), service composition and workflow generation practices [16] can be utilized to innovate this kind of services.
3) The innovated service enjoys a novel business model in a particular market. In other words, the suggested service are unforeseen, independent and semantically different than existing ones. *MapQuest Traffic API*, which provides real-time traffic information[4], is an example of such services.

It should be noted that the concept of the introduced service might already be familiar in some other application domains, but it could be still novel in its application within a particular market [4]. In this work, we are concentrating on innovating services based on the first pattern, where missing services are functionally similar to existing ones while exposing semantically different service signature.

B. Web services Networks

A Web services network, is a directed graph where nodes denote inputs and outputs of Web service operations while links (edges) represent Web service operations. The direction of a link is from inputs toward outputs. Hence, identifying missing Web services can be seen as a link prediction problem in a network. Therefore, Web services network construction is the fundamental part of the suggested solution. In the following we formalize the Web services network models and formally describe the problem of identifying missing Web services.

We envision *entity attributes* as the smallest pieces of Web services descriptions, which can be used to build Web services networks [14]. An entity attribute refers to an atomic unit of information about an entity, for example the address of a supplier or the salary of an employee. In the context of a semantic Web services, an entity attribute corresponds to the semantic class, tagged by *model reference* tag in SAWSDL[5] presentation, annotating an XML element with no child elements (i.e. leaf elements) or an XML attribute that appears in the schema of one of the messages produced or consumed by a Web service.

We write $d^{in}(op)$ and $d^{out}(op)$ to denote the set of entity attributes in inputs and outputs of service operation op. Each Web service operation op can be abstracted in terms of its input(s) and output(s): $op : d^{in} \rightarrow d^{out}$. The relationship $\rightarrow$ implies that invocation of operation op requires the presence of all inputs, d^{in}, and as a result of execution the output set d^{out} is produced. Analogously we write $d^{in}(ws)$ and $d^{out}(ws)$ to denote the set of entity attributes in inputs and outputs of all service operations of Web service ws.

We refer to the set of all entity attributes of ws as its *attributes* and it is defined as $atts(ws) = \{d \in \mathcal{A} \mid \exists op : d^{in} \rightarrow d^{out}, op \in ws, d \in (d^{in} \vee d^{out})\}$ where $\mathcal{A}$ is the set of all possible attributes. Given a federation (collection) of Web services, $\mathcal{F}$, the set of attributes of collection of Web services is the union of the set of attributes of its contained Web services, i.e. $atts(\mathcal{F}) = \cup_{ws \in \mathcal{F}} atts(ws)$.

Among the suggested presentations of Web services network models, we opted to *semantic network* variation similar to [19], [13]. A semantic network is constructed by introducing an directed edge from attributes of input parameters to attributes of output parameters of each Web service operation. Formally:

Definition 1 (Semantic Network)**.** *A semantic network is a classic (1-mode) loop-free directed graph $\mathcal{G} = \{N, E\}$ where E and N represent respectively a set of edges and a set of nodes in the graph. Set N consists of entity attribute nodes and set E is defined as:*

$$E = \bigcup_{d \in Atts(\mathcal{F})} (\{(d_i, d_j) \mid \exists ws \in \mathcal{F}, \exists op \in ws, d_i \in input(op) \wedge d_j \in output(op)\})$$

In above formula, it can be seen $N \subseteq atts(\mathcal{F})$ and $E \subseteq atts(\mathcal{F}) \times atts(\mathcal{F})$.

Based on the above network model, the Web service innovation problem can be formulated as follows. We are given a collection (federation) of semantic Web services $\mathcal{F}$ and respective domain ontology $\mathcal{O}$. Lets assume that the given Web services are transformed to a semantic network $\mathcal{G} = \{N, E\}$ as described in definition 1. The goal is to suggest a set of possible (meaningful) links that can be added to this network. In other terms, the objective is to build function

[1]http://api.cba.am/exchangerates.asmx?wsdl

[2]http://www.currencyserver.de/webservice/currencyserverwebservice.asmx?WSDL

[3]Price Runner http://www.pricerunner.se/

[4]MapQuest Traffic API: http://www.mapquestapi.com/traffic

[5]http://www.w3.org/TR/sawsdl/

$f : Atts(\mathcal{F}) \times Atts(\mathcal{F}) \rightarrow \{0, 1\}$.

III. Framework

The suggested framework for finding missing Web service operations in the given collection of annotated Web services consists of the following stages:

1) Constructing the Web services network. The semantically annotated Web services are transformed into semantic network structure where nodes are semantic concepts annotating input and output parameters of Web service operations and edges (links) denote Web service operations;
2) Collecting candidate links. We collect the set of links that do not present in the constructed network structure. These candidate links are constituent components of eventual suggested Web service operations;
3) Filtering candidate links. Since not all of the candidate links necessarily denote a meaningful association between two concepts, the nonmeaningful ones need to be identified and then removed. The meaningfulness of a link is determined based on its semantic similarity to existing links in the network. As we need to ensure diversity in the final results, semantically redundant links are identified and removed;
4) Link expansion. The concepts in two sides of a link are expanded to accommodate semantically relevant concepts. As a result, each link associates two group of concepts and in this way an expanded link resembles a Web service operation;
5) Collective evaluation. In the last step, we utilize previously observed properties of Web services networks for collective evaluation of the expanded links. To this end, we add these links to the original network and then measure certain properties of the resulting network. It is expected that the resulting network demonstrate the same characteristics as previously observed in Web services networks. In other words they are complex networks as they exhibit *scale free* [2], *small-world* properties [26] and *negative correlation degree on nodes* [22].

In the rest of this section, we describe in detail each of the above steps.

A. Constructing the Web services Network

The input for construction of Web services networks, as considered in this paper, is a collection of semantically annotated service interfaces encoded as SAWSDL where $modelReference$ tag is exploited to denote the mapping of annotated elements to respective semantic classes in a domain ontology. Using this and the network formation model presented in definition (1) we construct the respective semantic network and utilize it in the next stages.

As an illustrative example of annotated Web services, consider Figure 1*(a)*. Each box represents a Web service, where the nodes are Web service input or output parameters and links denote Webs service operations where the direction is from inputs toward outputs. Accordingly, there exist three annotated Web services (WS_1, WS_2, and WS_3), each of which consists of one operation (OP_1, OP_2, and OP_3 respectively). While P_1 - P_8 are expressing WSDL part-names or XML schema leaf element-names of input/output parameters of their respective operations, the semantic classes annotating these elements are symbolized by S_1 - S_7. In the process of constructing Web services network, the input parameters P_1, P_4 and P_7 are replaced by semantic classes S_1, S_3, S_7 respectively while S_2, S_4, S_5 and S_7 substitute the output parameters P_2, P_3, P_5, P_6 and P_8 in the similar manner. Figure 1*(b)* shows the resulting network after this stage of transformation. Next, the results of match-making process are exploited to unify the nodes representing matched input and output elements. This potentially results in emergence of new nodes with unified concept labels. Every emerging semantic node also inherits the incoming and outgoing edges from their parent nodes. Let's consider set $\{\langle S_2, S_3\rangle, \langle S_5, S_6\rangle\}$ as the only possible matching cases from the previous example. As a result of unification, nodes $S_{2,3}$ and $S_{5,6}$ are created. Finally, redundant edges are eliminated such that there will be only one edge connecting two nodes. Figure 1*(c)* illustrates the final semantic network structure.

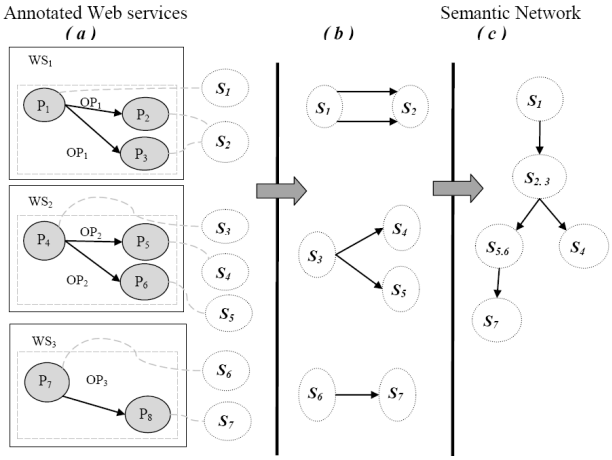

Fig. 1. Transformation of Annotated Web services to Semantic Network

B. Collecting Candidate Links

The set of candidate links can initially be seen as the collection of all pairs of concepts which are not connected directly to each other (we ignore the direction of association). For example, in case of Figure 1.(c), the initial candidate list includes the following pairs: $\{\langle S_1, S_{5,6}\rangle, \langle S_1, S_7\rangle, \langle S_1, S_4\rangle, \langle S_{2,3}, S_7\rangle, \langle S_{5,6}, S_4\rangle, \langle S_4, S_7\rangle\}$. As we are aiming for diversity, the pairs of concepts having indirect associations should to be removed. The indirect association implies that the concepts can be likely associated using service composition practices. Thus pairs of $\{\langle S_1, S_{5,6}\rangle, \langle S_1, S_7\rangle, \langle S_1, S_4\rangle, \langle S_{2,3}, S_7\rangle\}$ should be removed. In terms of graph theory practices this involves finding

reachable nodes from each other. A node S_j is reachable from a node S_i if there is a path that begins at S_i and ends at S_j. The set of pairs of reachable nodes can be identified using graph theory methods [25].

Let's assume that adjacency matrix of semantic network $\mathcal{G}$ is presented by matrix G. Each non-zero entry $g_{i,j}$ in this matrix denotes that there is a path with length one from node i to node j. Hence a path from node i to node j with maximum length K exists if the corresponding entry $m_{i,j}$, in matrix $\mathcal{M}$, is not zero where $\mathcal{M} = G \bigcup G^2 \bigcup .... \bigcup G^K$ and G^K denotes K times self multiplication of matrix G. A non zero entry $m_{i,j}$ implies that there is a Web service operation (individually or obtained by composition with others) which takes d_i as input and provides d_j as output. Similarly each zero entry $m_{p,q}$ reveals that a Web service operation taking d_p as input and providing d_q as output is missing and we are interested in collecting these zero entries. As we are looking for totally novel Web services, we do not distinguish between association from d_p to d_q and vice versa. Hence, we are only taking into account entries where both of $m_{p,q}$ and $m_{q,p}$ are zero. From now, we refer to a missing link by $\langle \hat{d}_p \rightarrow \hat{d}_q \rangle$ to distinguish it from an existing one denoted by $\langle d_i \rightarrow d_j \rangle$.

Finding all reachable pair of nodes requires $D-1$ times matrix multiplication where D is the diameter (the longest of all the calculated shortest paths) of the network. However, this method is computationally expensive for large Web services networks. Hence we utilize intrinsic properties of Web services networks to limit the number of times matrices are multiplied. Several research efforts in Web services networks [19], [13] have shown that networks constructed from real-world Web services, are exhibiting small-world properties [26]. In small-world networks although most of nodes are not neighbors, they can be reached from every other in a small number of hops (in maximum six hops), lets say $\mathcal{K}$ where $\mathcal{K} \ll D$. Thus, we only perform matrix multiplication $\mathcal{K}$ times. We refer to $\mathcal{M}$ as *small-world matrix* as it accumulates all paths in small-world distance in the Web services network. Returning back to network presented in Figure 1.(c), the output of this stage is $\{\langle S_{5,6}, S_4 \rangle, \langle S_4, S_7 \rangle\}$. Obviously, not all of these collected links are necessarily meaningful. In the next step, we measure the meaningfulness of candidate links.

C. Filtering Candidate Links

We assume a candidate link is meaningful if it is semantically similar to existence one(s) according to available local knowledge. The envisioned local knowledge includes the set of all concepts used for annotation denoted by $\mathcal{A}$ accommodated in the reference ontology $\mathcal{O}$, and the small-world matrix $\mathcal{M}$. For semantic similarity finding, we utilize the ontology operations borrowed from Andreasen and Bulskov's work [1]:

- *c-generalization*: we use $c \gg$ to denote any more general concepts to c;
- *Least Upper Bound (LUB)*: we use $LUB(c_1, c_2)$ to denote lowest common ancestor (i.e. conceptual sum) of c_1 and c_2;
- *Reachability*: operation $reachable(c_1, c_2)$ should evaluate to true provided that if c_2 is reachable from (is generalization of) c_1. In other words, $reachable(c_1, c_2)$ true if $c_2 \in c_1 \gg$

The utilized semantic similarity finding algorithm is presented in Algorithm 1. Accordingly, the algorithm examines each non zero entry $m_{i,j}$ in small-world matrix $\mathcal{M}$ to see if concepts $\hat{d}_p, \hat{d}_q$ in the given candidate link $\langle \hat{d}_p \rightarrow \hat{d}_q \rangle$ maintain an acceptable semantic distance (as an indicator of semantic similarity) to respective concepts in an existing link $\langle d_i \rightarrow d_j \rangle$ in the network. Measuring the semantic distance is advocated to $getSemanticDistance$ method outlined in Algorithm 2. Basically, the distance is regraded as the length of the path to traverse from $\hat{d}_p$ (or $\hat{d}_q$) to reach the closest common ancestor with d_i (or d_j respectively) in reference ontology $\mathcal{O}$. The distance is measured as discrete values, $\{1, 2, ...\}$, where 1 denotes that the compared concepts have a direct common parent. In the context of our example network, the similarity finding algorithm compares the given candidate pair $\langle S_{5,6}, S_4 \rangle$ (and $\langle S_4, S_7 \rangle$) with each pairs of $\langle S_{1,2}, S_{2,3} \rangle$, $\langle S_{2,3}, S_{5,6} \rangle$, $\langle S_{5,6}, S_7 \rangle$, $\langle S_{2,3}, S_4 \rangle$.

Starving for serendipitous (diversity), we remove the cases where the concepts in the given link is generalization or specialization of respective concepts in an existing link. To find out whether two concepts are generalization (or specialization) of each other we use *Reachability* operation and we return $+\infty$ when reachable concepts are identified. This large value ensures that specialized or generalized candidate links do not meet any given threshold the, thus they are removed. The output of this step is a set of candidate links which are determined to be meaningful and diverse while maintaining certain semantic similarity to existing links.

```
Input: ⟨d̂_p → d̂_q⟩, M, thr
 n ← M.size
for i to n do
    for j to n do
        if M_{i,j} ≠ 0 then
            dist_p ← getSemanticDistance(d_i, d̂_p)
            dist_q ← getSemanticDistance(d_j, d̂_q)
            if dist_p <= thr OR dist_q <= thr then
                return (dist_p + dist_q) /2
            end
        end
    end
end
 return 0
```

Algorithm 1: Semantic Similarity Finding

D. Link Expansion

The goal of this step is to extend the candidate links to resemble Web service operations. So far, each link only includes one concept as input (i.e. as a source node) and another concept as output (i.e. as a target node). However,

Input: $d, \hat{d}$
if $reachable(d, \hat{d})$ *OR* $reachable(\hat{d}, d)$ **then**
  return $+\infty$
else
  $lub \leftarrow LeastUpperBound(d, \hat{d})$
  return $distance(lub, \hat{d})$
end

Algorithm 2: Measuring Semantic Distance

Web service operations generally take more than one concept as input or output. So we expand each concept to a group of semantically related concepts. For instance, concept *City* is expanded to accommodate *Street* and *Zip Code* concepts as they are determined to be related to a same topic.

This group of semantically related concepts are discovered using an unsupervised agglomerative clustering technique presented in our previous work [20]. The underlying clustering principle is based on the intuition that terms which convey similar semantic often co-occur together in a same neighborhood [3]. In our case, the envisioned neighborhood is the input and output fragment of Web service operations. The input for the leveraged clustering method are a set of semantic classes used for annotations, $atts(\mathcal{F})$, and the respective association rules between these concepts.

The utilized clustering algorithm is a refinement of bottom-up basic agglomerative hierarchical algorithms [12] and it starts with all the data points as a separate cluster. Each step of the algorithm involves evaluation of merging of two clusters that are the most close according to a merging criteria. The output of clustering method is a set of clusters, $\mathcal{L}$, where each cluster is a set of semantically related concepts. In other words, $\mathcal{L} = \{L_i | L_i \subset atts(\mathcal{F}) \wedge L_i \bigcap L_j = \varnothing \wedge L_1 \bigcup L_2 \bigcup .... \bigcup L_n = atts(\mathcal{F})\}$. Due to the nature of the utilized clustering algorithm, the clusters are disjoint. Assuming $d_p \in L_p$ and $d_q \in L_q$, then candidate link $\langle \hat{d_p} \rightarrow \hat{d_q} \rangle$ is expanded to $\langle L_p \rightarrow L_q \rangle$.

IV. Experimental Results

A. Experimental Settings

We implemented the proposed framework in Java, iGraph[6] package of R and Pellet[7] ontology reasoner. The proposed solution is evaluated using SAWSDL-TC collection[8] containing 753 semantically annotated WSDLs[9](in SAWSDL format). This collection is frequently used in Web service community for benchmarking purpose [15]. The advantage of this dataset is that the Web services are already annotated and they represent real-world Web services.

Using this dataset, we created a semantic network and then collected the missing links. The candidate links are then filtered based on their semantic distance to existing Web services. We used two distance thresholds where threshold 1 considers only direct siblings (concepts with immediate common parents), and threshold 2 includes both direct and indirect siblings (concepts with not immediate common ancestors that can be reached in maximum two steps). We refer to the semantic network constructed using the actual dataset as Original network.

Accordingly, the Original network accommodates 255 nodes and 2230 links. This means that the size of initial set of candidate links (i.e. the initial quantity of missing links) is 33930. By applying semantic based filtering with thresholds 1 and 2, the size of candidate links is diminished to 2348 and 3113 cases respectively. The number of concepts annotating basic elements of inputs and outputs of Web service in SAWSDL collection accounts for 261 unique cases and this set is used as input to our clustering algorithm. The clustering algorithm groups these concepts into 140 disjoint clusters where each cluster accommodates a group of maximum four semantically relevant concepts. These clusters are then used in the link expansion stage and in this way an unforeseen collection of Web service operations is discovered. In the next section, we assess the meaningfulness of introduced links collectively using network metrics.

B. Collective Evaluation of Suggested Links

A major problem in link prediction is that a prior probability of a link is typically quite small due to sparsity of information in the examined dataset, as reported by Rattigan and Jensen [23]. This causes difficulty in quantifying the level of confidence in the predictions. Similarly, our work also suffers from weak confidence measurement for the predicted links. A suggested solution to improve the quality of prediction is to perform collective link prediction and evaluation [8]. Based on these findings we considered collective evaluation of predicted links in the Web services network. Our collective evaluation method is derived from already observed properties of Web services networks.

It is shown by several research work [19], [13] that semantic networks constructed from real-world annotated Web services are complex networks as they exhibit *scale free* [2], *small-world* properties [26] and *negative correlation degree on nodes* [22]. Hence, it is expected that semantic networks constructed by augmenting expanded candidate links with Original network demonstrate the same characteristics as previously observed in semantic networks made using real-world Web services. In this way, we evaluate the collective meaningfulness of newly introduced Web service operations. The semantic networks obtained by augmenting Original network with the candidate links obtained by thresholds 1 and 2 are referred as *Original+Thr.1* and *Original+Thr.2* respectively. The characteristics of all three networks are presented in Table.I and Table.II.

Our hypothesis is that networks of *Original+Thr.1* and *Original+Thr.2* expose the same network characteristics as observed in real-world network. In other words, they are

[6]http://igraph.sourceforge.net/

[7]http://clarkparsia.com/pellet/

[8]SemWebCentral.org. Retrieved at December 2011, from http://projects.semwebcentral.org/projects/sawsdl-tc/

[9]The collection originally contains about 1080 WSDLs and we omitted re-sampled Web services for the sake of accuracy.

small-world and scale-free networks and they exhibit negative correlation degree on nodes. Similar to many studies [26] on the small-world networks, the analysis is restricted to a giant component in the networks (i.e. the maximal connected sub-graph of the network).

Small-worldness: According to Watts and Storgatz [26], small world networks are networks with the following characteristics: exhibiting small average shortest path length, and exposing high clustering coefficient. These properties are measured by the average shortest path and clustering coefficient metrics, which are denoted by L and C symbols respectively in Table I. In the interest of verifying small-world characteristic, the computed metrics in the target networks are compared with those estimated from similar random network generated based on Erdos & Renyi (ER) model [7] (with same number of nodes and edges appearing in the actual network). The computed average shortest path and average clustering coefficient metrics for the random network are denoted by L_{Random} and C_{Random} symbols while those for the actual semantic network are denoted by L_{Actual} and C_{Actual} symbols respectively. If a network exposes the small world properties, then it is expected that $L_{Random} \lesssim L_{Actual}$ (i.e. average shortest path is almost equal or slightly larger than of a random network) and $C_{Actual} \gg C_{Random}$ (i.e. average clustering coefficient is much larger than that of a random network) [26]. In order to explore the extent to which the small-world topology changes with parameter variation, we exploit a measurement of *smallworldness*, shown by S_{index}, proposed by Humphries and Gurney [11] which is defined as:

$$\gamma = \frac{C_{Actual}}{C_{Random}}, \lambda = \frac{L_{Actual}}{L_{Random}},\ S_{index} = \frac{\gamma}{\lambda} \qquad (1)$$

According to Watts and Strogatz [26], in order to meet small world criteria given above, the network model should fulfill the following conditions: $\gamma \gg 1$, $\lambda > 1$, and $S_{index} > 1$. The S_{index} scales linearly with the size of vertexes of the network. Montoya and Sole [21] suggest that holding only the second condition for small networks (200-3000 vertexes) is sufficient to demonstrate small-world properties. We use S_{index} metric to compare networks constructed using different matching scheme and dataset with respect to their small-world properties.

It can be seen in Table I that networks *Original+Thr.1* and *Original+Thr.2* are exhibiting small world conditions, because $L_{Random} \lesssim L_{Actual}$ and $C_{Actual} \gg C_{Random}$ are holding, similar to *Original* network.

Scale-freeness: Emergence of power-law degree distribution in a network, as a prominent sign of scale free networks, implies that few nodes are highly connected, whereas majority of nodes have a low degree of connectivity (i.e. existence of hub nodes)[2]. Expecting to observe the same pattern, we examine outgoing edge distribution in all our networks. The results for all categories of the networks are fitted to a power law function (in log-log plot) where x represents the outgoing degree and y denotes the frequency of nodes with the same outgoing edge degree. The exponents of power-law function (i.e. α) for all networks are summarized in the first column of Table II.

TABLE I
SMALL-WORLD PROPERTIES OF EXAMINED NETWORKS. **L**: AVERAGE PATH LENGTH, **C**: CLUSTERING COEFFICIENT,S_{index}: INDEX OF SMALL-WORLDNESS.

Network		L	C	S_{index}
Original	**Actual**	**2.2832**	**0.3942**	**6.8451**
	Random-ER	2.7725	0.0699	
Original+Thr.1	**Actual**	**2.8166**	**0.392**	**5.3828**
	Random-ER	2.7246	0.0704	
Original+Thr.2	**Actual**	**2.5776**	**0.3683**	**3.7656**
	Random-ER	2.4937	0.0946	

TABLE II
NETWORK PROPERTIES. **N**: NUMBER OF NODES IN THE BIGGEST CONNECTED COMPONENT OF THE NETWORK, **E**: NUMBER OF EDGES IN THE NETWORK, **P**: POWER-LAW DEGREE EXPONENT, **D**: DEGREE CORRELATION

Network	**N**	**E**	**P**	**D**
Original	255	2230	1.39	-0.1072
Original+Thr.1	258	2348	1.38	-0.1032
Original+Thr.2	260	3113	1.31	-0.1666

Plotting the network in log-log scale shows that networks of *Original+Thr.1* and *Original+Thr.2* are presenting near power-law like distribution, similar to *Original* network, with the exponent ranging from 1.31 and 1.38 respectively. Thus, both of these networks exhibit scale-free properties.

Correlation degree on nodes: A network is said to have positive correlation on its degree, if nodes with high number of connection tend to be connected with other nodes which also have many links. Alternatively, if the preference is to attach to those having small quantity of connection, then it is said to have negative correlation on degrees. While negative correlation on degrees is common in technological and biological networks, for example in Internet, WWW, protein interactions networks and Web services networks [19], positive correlation is mainly observed in social networks (e.g. network of actors) [22]. It can be seen in Table II, that all networks of *Original+Thr.1* and *Original+Thr.2* are exposing negative correlation on their node degrees similar to *Original* network.

The results of above analysis all together supports the validity of our hypothesis. This verifies that introducing the expanded candidate links, as constituting components of new Web services, to the given collection of Web services preserves intrinsic nature of Web services networks. While at the same time, their semantic similarity to existing Web services supports (to some extend) their meaningfulness.

Discussion

The implicit assumption in measuring semantic distance is that the evaluated concepts are organized in a same ontology structure. However, this assumption is not always holding and Web services, even those belonging to the same application domain, in reality can be semantically annotated

using different ontologies depending on the requirement and availability of resources. In our experiment, we manually integrated all small domain ontologies used for annotation of SAWSDL-TC collection with a SUMO[10] upper level ontology. The resulting ontology accommodates of 4716 concepts, 954 object properties and 10 data properties. In this way, one unified reference ontology structure is obtained and utilized for semantic distance computation. We acknowledge that any possible bias introduced by the ontology engineer in integrating the ontologies could have negative effect on the validity of our results. However this issue does not effect the validity of the underlying method and the proposed framework.

V. Related Work

From one perspective, this research work is quite relevant to link prediction and generative graph models in link mining research area [8]. The research in link prediction can be roughly divided into two major groups: 1) either some links are observed and the goal is to predict unobserved links or 2) there is a temporal factor where given some links at time t we need to predict the emerging links at time $t + 1$. Since our utilized dataset does not carry the time stamp of Web service creation time, we could not employ temporal based link prediction techniques. Some other approaches take advantage of structural properties of the network to make prediction [18]. Basically, they devise a link prediction model using different graph proximity measures.

The recent rapid increase in the quantity of services on the Web has motivated several research efforts to study the Web the structure and evolution of Web services in general and interaction between services in Web services ecosystem in particular [9], [10], [27]. Weiss and Sari [27] used a birth-death process to study the evolution of the diversity in a mashup ecosystem. They reported that diversity of the mashup ecosystem is increasing with time, however, not monotonically. Huan et al. [10] propose a network based methodology to study the structural and temporal characteristics of service ecosystem and to predict its future behavior. They constructed Web services interaction networks and utilized link prediction method to predict which composite services will likely emerge in the future. They treat future service interactions as missing links. The prediction is performed based on historical data on Web services usage and their presence in composition with other services. In a quite similar manner, Grzech el al. [9] considered Web services networks to model the interaction patterns between Web services during composition and execution. The ultimate objective is to predict and evaluate the future service interaction and usage patterns. To this end, they employed techniques from dynamic network structure prediction practices and trained their systems by gathering real-world data form a Web service platform. It can be seen that similar to [9], [10], [27] we also rely on Web services networks and devise a network driven methodologies for Web service predictions. However, none of above mentioned works targets finding unforeseen services in a cold-start manner where no training dataset is available. Hence, our work is quite novel from this perspective.

[10]Suggested Upper Merged Ontology, http://www.ontologyportal.org/

VI. Conclusion and Future Work

In this paper we proposed an framework for recommending a novel set of Web service operations to a given collection of semantic Web services. The solution relies on the network structure of Web services for finding and recommending new Web services. While the meaningfulness of newly introduced Web service operation is determined based on its semantic similarity to existing ones, Web services network characteristics are used for collective evaluation of all recommended services. The experimental results demonstrated that the networks resulted from these recommended web services exhibit same properties as observed in real-world Web services networks. The proposed framework provides web service community with new network driven methods for recommending and evaluation of novel Web services.

The future work extends the current framework in two directions. In the first direction, we extend novel web service discovery step by employing We service composition practices. The goal is suggest development of new Web services that can be used in composition with other existing services where the composite Web service provides a new solution for a non-trivial demand. The next direction involves devising methods for measuring the added-value of suggested Web services using both QoS fo Web service and trustworthiness of service provider.

References

[1] T. Andreasen and H. Bulskov. Conceptual querying through ontologies. *Fuzzy Sets Syst.*, 160(15):2159–2172, Aug. 2009.

[2] A. L. Barabasi and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, Oct. 1999.

[3] C. Burgess, K. Livesay, and K. Lund. Explorations in Context Space: Words, Sentences, Discourse. *Discourse Processes*, 25(2/3):211–257, 1998.

[4] P. Den Hertog. Knowledge-intensive business services as co-producers of innovation. *International Journal of Innovation Management*, 4(04):491–528, 2000.

[5] J. Domingue and D. Fensel. Towards a service web: Integrating the semantic web and service orientation. *IEEE Intelligent Systems*, Jan. 2008.

[6] X. Dong, A. Halevy, J. Madhavan, E. Nemes, and J. Zhang. Similarity search for web services. In *VLDB '04*, volume 30, pages 372–383. VLDB Endowment, 2004.

[7] P. Erdos and A. Renyi. On the evolution of random graphs. *Publ. Math. Inst. Hungary. Acad. Sci.*, 5:17–61, 1960.

[8] L. Getoor and C. P. Diehl. Link mining: a survey. *SIGKDD Explor. Newsl.*, 7(2):3–12, 2005.

[9] A. Grzech, K. Juszczyszyn, P. Stelmach, and . Falas. Link prediction in dynamic networks of services emerging during deployment and execution of web services. ICCCI'12, pages 109–120. Springer-Verlag, 2012.

[10] K. Huang, Y. Fan, and W. Tan. Recommendation in an evolving service ecosystem based on network prediction. *IEEE Transactions on Automation Science and Engineering*, (99):1–15, 2014.

[11] M. D. Humphries and K. Gurney. Network 'Small-World-Ness': A Quantitative Method for Determining Canonical Network Equivalence. *PLoS ONE*, 3(4):e0002051+, 2008.

[12] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley and Sons, New York, 1990.

[13] H. Kil, S.-C. Oh, E. Elmacioglu, W. Nam, and D. Lee. Graph theoretic topological analysis of web service networks. *World Wide Web*, 12:321–343, 2009.
[14] P. Küngas and M. Dumas. Redundancy detection in service-oriented systems. In *WWW*, pages 581–590, 2010.
[15] U. Küster and B. König-Ries. Towards standard test collections for the empirical evaluation of semantic web service approaches. *International Journal of Semantic Computing*, 2(03):381–402, 2008.
[16] D. Leake and J. Kendall-Morwick. Towards case-based support for e-science workflow generation by mining provenance. ECCBR '08, pages 269–283. Springer-Verlag, 2008.
[17] C. Legner. Is there a market for web services? In *ICSOC 2007 Workshops*, pages 29–42. Springer-Verlag, 2009.
[18] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM '03*, pages 556–559. ACM, 2003.
[19] S. Mokarizadeh, P. Küngas, and M. Matskin. Evaluation of a semi-automated semantic annotation approach for bootstrapping the analysis of large-scale web service networks. In *Web Intelligence*, pages 388–395, 2011.
[20] S. Mokarizadeh, P. Küngas, and M. Matskin. Ontology acquisition from web service descriptions. In *SAC'13*, pages 325–332. ACM, 2013.
[21] J. M. Montoya and R. V. Solé. Small world patterns in food webs. *Journal of Theoretical Biology*, 214(3):405 – 412, 2002.
[22] M. E. J. Newman. Assortative Mixing in Networks. *Physical Review Letters*, 89:208701+, Oct. 2002.
[23] M. J. Rattigan and D. Jensen. The case for anomalous link discovery. *SIGKDD Explor. Newsl.*, 7(2):41–47, Dec. 2005.
[24] C. Riedl, T. Böhmann, J. M. Leimeister, and H. Krcmar. A framework for analysing service ecosystem capabilities to innovate. In *ECIS*, pages 2097–2108, 2009.
[25] M. van Steen. *Graph Theory and Complex Networks: An Introduction*. Maarten van Steen, April 2010.
[26] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–2, 1998.
[27] M. Weiss and S. Sari. Diversity of the mashup ecosystem. In J. Cordeiro and J. Filipe, editors, *WEBIST*, pages 106–111. SciTePress, 2011.
[28] Z. Zheng, H. Ma, M. R. Lyu, and I. King. Qos-aware web service recommendation by collaborative filtering. *Services Computing, IEEE Transactions on*, 4(2):140–152, 2011.