Utilizing Web services Networks for Web service Innovation

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

Peep Künugas  
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 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 novel Web services.

Keywords—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], [18], [10]. These recommendation solutions aim to suggest the best choice(s) among plenty of alternatives to satisfy service consumers demands. Conversely, management of new service development, in the presence of such huge quantity of Web services [10], becomes a prime challenge for service developing sectors as well [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 new services) 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] for suggesting missing Web services. 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 its 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 in SAWSDL\(^1\) format.

The rest of this paper is organized as follows. While Section II summarizes the fundamentals and requirements of the envisioned solution, we outlines the suggested framework and the underlying procedures 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, we describe the requirements and main preparation steps which are needed to find the candidate services that can be introduced to the given collection of Web services. First we discuss briefly what kinds of Web services can be introduced and remark the kind which we are aiming for in this work. Next, we present a unified model for building Web services networks. The latter allows utilization

\(^1\)http://www.w3.org/TR/sawSDL/
of network structure for Web service recommendation and collective evaluation of results as described in Section IV.

A. Web services Innovation

Finding missing, but a valuable Web service, in the given collection of services is analogous to innovating a service which is different than existing ones. Considering a Web services repository as an electronic market [16], 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 others and it varies in quality of service details such as quality, price, service provider affiliation, etc. For instance, there are plenty of services on the Web providing currency exchange rates. In principal, they all provide a similar service in terms of input and output semantic. For example, the currency exchange rate services offered by Central Bank of Armenia\(^2\) and Daenet GmbH\(^3\) mainly differ in the provided exchange rate (i.e. price), service provider affiliation (governmental vs. private sector). 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\(^4\) which collects a price list for a given item from various Web shops. Assuming the presence of Web service interface (WSDL or REST) of existing Web based businesses, service composition and mashup practices 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 is unforeseen, independent and semantically different than existing ones. MapQuest Traffic API,\(^5\) which provides real-time traffic information, 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 [5]. In this work, we are concentrating on innovating services based on the first pattern, i.e. based on their semantic similarity to the existing ones.

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 operation. The direction of a link is from input node toward output node. 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 below we formalize the Web services network models and formally describe the problem of identifying missing Web services.

We envision entity attributes 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, like 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 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 \(atts(ws) = \{d \in A \mid \exists op : d^{in} \rightarrow d^{out}, op \in ws, d \in (d^{in} \lor d^{out})\}\) where \(A\) is the set of all possible attributes. Given a federation (collection) of Web services, \(F\), the set of attributes of collection of Web services is the union of the set of attributes of its contained Web services, i.e. \(atts(F) = \cup_{ws \in F} atts(ws)\).

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 \(G = \{N, E\}\) 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:

\[
E = \bigcup_{d \in atts(F)} \{(d_i, d_j) \mid \exists ws \in F, \exists op \in ws, d_i \in input(op) \land d_j \in output(op)\}
\]

In above formula, it can be seen \(Natts(F)\) and \(E \subseteq atts(F) \times atts(F)\).

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 \( F \), respective domain ontology \( O \). Let’s assume that the given Web services are transformed to a semantic network \( G = (N, E) \) where the nodes are semantic classes, \( N \subseteq \text{Atts}(F) \), and the links \( E \subseteq \text{Atts}(F) \times \text{Atts}(F) \) represent Web service operations. Now, the goal is to predict the possible (meaningful) links that can be added to this network. In other terms, the objective is to build function \( f : \text{Atts}(F) \times \text{Atts}(F) \rightarrow \{0, 1\} \).

III. FRAMEWORK

In this section we present a framework for finding missing Web services in the given collection of semantically annotated Web services. The framework 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 non meaningful ones need to be identified and then removed. The meaningfulness of link is determined based on its semantic similarity to existing links in the network. We also need to ensure diversity in the final results; hence, semantically redundant links are also identified and removed;
4) Link expansion. The concepts in two sides of a link are extended 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 filtered 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 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). 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 \( P_1 - P_5 \) 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 \( P_1 \) and \( P_4 \) are replaced by semantic classes \( S_1 \) and \( S_3 \) respectively while \( S_2 \) and \( S_4 \) substitute the output parameters \( P_2, P_3 \) and \( P_5 \) 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.

B. Collecting Candidate Links

The set of candidate links can be seen initially 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. Transformation of Annotated Web services to Semantic Network**

Annotated Web services

(a) \( WS_1 \)

(b) \( WS_2 \)

(c) \( WS_3 \)

Semantic Network

(a) \( S_1 \)

(b) \( S_2 \)

(c) \( S_3 \)
As we are aiming for diversity, the pairs of concepts having indirect associations need to be removed. The indirect association implies that the concepts can be likely associated using service composition practices. Thus pairs of \(\{\langle S_1, S_6 \rangle, \langle S_1, S_7 \rangle, \langle S_1, S_4 \rangle, \langle S_{2.3}, S_7 \rangle, \langle S_{5.6}, S_4 \rangle, \langle S_4, S_7 \rangle\}\) should be removed. In terms of graph theory practices this involves finding reachable nodes from each other [25]. A node \(S_i\) is reachable from a node \(S_j\) 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.

Let’s assume that adjacency matrix of semantic network \(G\) is presented by \(G\). Each non-zero entry \(g_{i,j}\) in this matrix denotes that there is a path with length one (i.e. an edge) 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 \(M\), is not zero where \(M = G \cup G^2 \ldots \cup G^K\) 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. Thus 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 \(\langle d_p \rightarrow d_q \rangle\) 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 \(K\) where \(K \ll D\). Thus, we only perform matrix multiplication \(K\) times. We refer to \(M\) 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 \(\{\langle S_{5.6}, S_4 \rangle, \langle S_1, S_7 \rangle\}\). 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 \(A\) accommodated in the reference ontology \(O\), and the small-world matrix \(M\). 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 \(M\) to see if concepts \(d_p, d_q\) in the given candidate link \(\langle d_p \rightarrow d_q \rangle\) 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 \(d_p\) (or \(d_q\)) to reach the closest common ancestor with \(d_i\) (or \(d_j\) respectively) in reference ontology \(O\). The distance is measured as discrete values, \(\{1, 2, \ldots\}\), where 1 denotes that the compared concepts have a direct common parent. In context of our example network, the similarity finding algorithm compares the given pair \(\langle S_{5.6}, S_4 \rangle\) (and \(\langle S_1, S_7 \rangle\)) 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 the threshold, thus they are eliminated. 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.

**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, Web service operations in reality usually take more than
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 \( \langle d_p \rightarrow d_q \rangle \) 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 reported to 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

```plaintext
Algorithm 1: Semantic Similarity Finding
Input: \( \langle d_p \rightarrow d_q \rangle, M, \text{thr} \)
\( n \leftarrow M\text{.size} \)
for \( i \) to \( n \) do
    for \( j \) to \( n \) do
        if \( M_{i,j} \neq 0 \) then
            dist\(_p\) \leftarrow \text{getSemanticDistance}(d_i, \hat{d}_p)
            dist\(_q\) \leftarrow \text{getSemanticDistance}(d_j, \hat{d}_q)
            if dist\(_p\) <= \text{thr} OR dist\(_q\) <= \text{thr} then
                return \((\text{dist}_p + \text{dist}_q) / 2\)
        end
    end
end
return 0

Algorithm 2: Measuring Semantic Distance
Input: \( d, \hat{d} \)
if \( \text{reachable}(d, \hat{d}) \) OR \( \text{reachable}(\hat{d}, d) \) then
    return +\(\infty\)
else
    lub \leftarrow \text{LeastUpperBound}(d, \hat{d})
    return \text{distance}(\text{lub, } \hat{d})
end
```

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 of Web service operations. The input for the leveraged clustering method is a set of semantic classes used for annotations, \( \text{atts}(\mathcal{F}) \), together with association rules [6] for each pair of concept labels.

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, \( \mathcal{L} = \{ L_i | L_i \subset \text{atts}(\mathcal{F}) \land L_i \bigcap L_j = \emptyset \land L_1 \bigcup L_2 \bigcup ... \bigcup L_n = \text{atts}(\mathcal{F}) \} \) where \( \text{atts}(\mathcal{F}) \) is the set of all semantic classes used for annotation in Web service collection \( \mathcal{F} \), as denoted earlier in Section III-A. Due to the

\(^6\)http://igraph.sourceforge.net/
\(^7\)http://clarkparsia.com/pellet/
\(^9\)The collection originally contains about 1080 WSDLs and we omitted re-sampled Web services for the sake of accuracy.
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], [4], [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]. Moreover, in our previous research work [19] we showed that these properties are also preserved in acceptable threshold of semantic annotation of inputs and outputs of Web service operations. 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. We refer to 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. In other words, they are 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 Strogatz [26], small world networks are networks with the following characteristics: 1) exhibiting small average shortest path length, and 2) exposing high clustering coefficient. These properties are measured by the average shortest path and clustering coefficient metrics, which are denoted by \( L \) and \( C \) respectively. If a network exposes the small world properties, then it is expected that \( L_{\text{Random}} \ll L_{\text{Actual}} \) (i.e. average shortest path is almost equal or slightly larger than of a random network) and \( C_{\text{Actual}} \gg C_{\text{Random}} \) (i.e. average clustering coefficient is much larger than that of a random network) [26].

\[
\gamma = \frac{C_{\text{Actual}}}{C_{\text{Random}}}, \quad \lambda = \frac{L_{\text{Actual}}}{L_{\text{Random}}}, \quad S_{\text{index}} = \frac{\gamma}{\lambda}
\]

According to Watts and Strogatz [26], in order to meet small world criteria given above, the network model should fulfill the following conditions: \( \gamma > 1 \), \( \lambda > 1 \), and \( S_{\text{index}} > 1 \). The \( S_{\text{index}} \) 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 \( S_{\text{index}} \) metric to compare networks constructed using different matching scheme and dataset with respect to their small-world properties.

The computed small world network metrics for both actual and counterpart random networks (marked by Actual and Random-ER suffixes) are shown under columns \( L \) and \( C \) of Table I. It can be seen that networks Original+Thr.1 and Original+Thr.2 are exhibiting small world conditions. In other words, \( L_{\text{Random}} \ll L_{\text{Actual}} \) and \( C_{\text{Actual}} \gg C_{\text{Random}} \) are holding, similar to Original network. Thus it can be concluded that both of these networks are small-world networks.

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. \( \alpha \)) for all networks are summarized in the first column of Table II.

<table>
<thead>
<tr>
<th>Network</th>
<th>( L )</th>
<th>( C )</th>
<th>( S_{\text{index}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>Actual</td>
<td>2.2832</td>
<td>0.3942</td>
</tr>
<tr>
<td></td>
<td>Random-ER</td>
<td>2.7725</td>
<td>0.0699</td>
</tr>
<tr>
<td>Original+Thr.1</td>
<td>Actual</td>
<td>2.8166</td>
<td>0.392</td>
</tr>
<tr>
<td></td>
<td>Random-ER</td>
<td>2.7246</td>
<td>0.0704</td>
</tr>
<tr>
<td>Original+Thr.2</td>
<td>Actual</td>
<td>2.5776</td>
<td>0.3683</td>
</tr>
<tr>
<td></td>
<td>Random-ER</td>
<td>2.4937</td>
<td>0.0946</td>
</tr>
</tbody>
</table>

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 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\textsuperscript{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.

\begin{table}[h]
\centering
\caption{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}
\begin{tabular}{lcccc}
\hline
Network & $N$ & $E$ & $P$ & $D$ \\
\hline
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 \\
\hline
\end{tabular}
\end{table}

VI. 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 [17]. 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.

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

\hspace{1cm}

\textsuperscript{10}Suggested Upper Merged Ontology, http://www.ontologyportal.org/
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.

Although there are many Web service recommendation methods, our work is among a small number of research efforts that aims for suggesting unforeseen Web services based on only available local knowledge.

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. The next direction mostly involves devising methods for measuring the added-value of suggested Web services using both QoS metrics of Web services and related semantic information.

REFERENCES