Logic and Event Based Semantic Relationship Evolution in Service Semantic Link Network

Yu Yu[1], Anping Zhao[2]

[1] School of Public Affairs, Chongqing University, China
[2] School of Computer and Information Science, Chongqing Normal University, China
yuyu@cqu.edu.cn, apzhao@cqnu.edu.cn

Abstract

To facilitate reasoning and predicting service evolution relationships with incomplete, partial or uncertain knowledge for supporting an intelligent service application and to preserve the relationship consistency in the whole service networks during Web service evolution processes over time. An approach was presented for event-based relationships evolution detection and reasoning among web services by combining first-order logic and probabilistic graphical models in a single representation. The Web service relationship evolution model was constructed by using available probability information and knowledge based on the related event and their inherent service relationship dependencies. And S-MLN was taken as a logical framework for Web service relationships evolution reasoning with uncertainty to discover and predict evolutionary service relationship classification. The events and evolutionary measures of the research were found to be helpful for evaluating Web service evolution relationships. And it is effective to evaluate the quality of service set classification by S-MLN based evolution relationships prediction. The study identifies the theoretical foundations of web service evolution discovery in the context of SLN. This study, based on an established theoretical foundation, will help the research community to gain a deeper understanding of the dynamic relationship in the semantic context of the web services network.

Keywords: Evolutionary event, Semantic relationship, Reasoning, Link prediction

1 Introduction

In the Service Oriented Architecture (SOA) understanding and coping with changes is challenging because of the distributed and dynamic nature of services. On one hand, the problem is quite complex and challenging due to the fundamentally distributed nature of SOA systems, whose constituent parts may across organizations and beyond the domain of any individual entity control. On the other hand, the overall SOA systems need to be aware of only the changes that impact the interface specifications; any changes to the service implementations that do not impact their interfaces are completely transparent to the overall system. Hence, the growing quantity of the produced Web services requires a new generation of model and tools to exploit this evolution to reduce costs and optimize the quality of utility. To represent Web service relationship network, we use a semantic data model Service Semantic Networks (S-SLN) to define semantic structure among Web services [1]. Its nodes can be any Web services. Its edges can be any semantic relationships. A S-SLN is a directed graph, denoted as Ss(ServiceSet, LinkSet), where Ss is the name of the S-SLN, ServiceSet is a set of Web services, and LinkSet is a set of semantic links in form of $S \xrightarrow{\alpha} S'$, where S, $S' \in$ ServiceSet, and α is a semantic factor representing a semantic relationship between S and S'. S-SLN is the underlying semantic model for effectively implementing automatic Web service search and composition by a relationship dependency network which connects services with different types of relationships. It allows us to specify types of the inherent service relation dependencies to describe the behaviors of Web services and focuses on collaboration, availability and navigation. The maintenance and evolution of SSLN as systems comprising multiple service-based applications are becoming a growing issue. As the service inventory grows, the services are more often reused. As the result, the more consequences a service change can cause on related applications in the S-SLN. To preserve the consistency in the whole S-SLN, evolutionary processes have to consider all relationships to the changing configuration item.

To this end, automatic discovery of evolution is of utmost importance, several steps have to be taken.

- Define a semantic model for relationship representation allowing a uniform description of all identified relationship variations applied in the realization of Web service based applications.
- Design a model capable of supporting the evolution, prediction, representation of an appropriate set of

*Corresponding Author: Anping Zhao; E-mail: apzhao@cqnu.edu.cn
DOI: 10.3966/160792642018011901020

relationships relevant for S-SLNs according to their dynamic needs.

- Understand the characteristics of services residing on the different abstraction layers between business, as well as the possible evolutionary relationships between them to provide a basis for a consistency check of the dynamic Web service information.

This paper focuses on how to discover relationship's evolution among Web services based on the Event in S-SLN, which combining several dependency representation methods of graph including undirected graph structure Markov network and directed graph structure. We propose a methodology to detect evolution Web service relationships by evolution event, which can help to identify Web service relationship evolution on the basis of their inherent service relationship dependencies.

The main contributions of this paper are as follows:

- We employ the model which based on the evolution events forWeb service relationship evolution in the S-SLN.
- Development of a Markov logic based approach for Web service evolution relationship prediction.
- We conduct preliminary experiments and performance studies, which verify the feasibility and effectiveness of our methods.

The rest of this paper is organized as follows: In section 2 we review related work, Section 3 introduces the problem definition and preliminaries related to our work. Section 4 gives the learning process of event based evolution model. And section 5 describes the approach to Web service evolution relationship reasoning based on the graphical model. We show the effectiveness of the presented approach by experimental result in section 6. Finally, we summarize our conclusions and the future research in section 7.

2 Related Work

The work on relationships in service-based applications found in the literature differs according to the purpose of relationship assessment and the considered set of relationships. Regarding the purpose for relationships, existing approaches separate in two general groups: providing support for relationship prediction to business-specific purposes or for Web service evolution to IT-specific purposes.

2.1 Evolution of Web Service Networks

The most recent work on service relationship management for maintenance and evolution of service networks has been developed [2]. Authors present a framework for collection, validation, and representation of service relationship information. To preserve the consistency in the whole service network, maintenance and evolution processes have to consider all relations to the changing configuration item. Fokaefs et al. analyzed the evolution of web services using a tool called VTracker which is based on the tree edit distance algorithm to calculate the minimum edit distance between two trees [3]. To better understand the evolution of web services and to facilitate the systematic and automatic maintenance of web-service systems, they introduce a specialized differencing method for web services [4]. The paper is concerned with the dynamic evolutionary analysis and quantitative measurement of primary factors that cause service inconsistency in service-oriented distributed simulation applications. A novel dynamic evolution model extended hierarchical service-finite state automata (EHS-FSA) is constructed based on finite state automata (FSA), which formally depict overall changing processes of service consistency states. And also the service consistency evolution algorithms (SCEAs) based on EHS-FSA are developed to quantitatively assess these impact factors [5]. In the work of Min Song et al. [6], the temporal logic of the semantic relation inner component and between components is expressed through semantic relation protocol items, the semantic relation model, the semantic relation matrix and semantic relation link matrix of the IWA are built when we detect and analyze the evolution characteristic. Romano and Pinzger propose a tool called WSDLD iff to extract fine-grained changes from subsequent versions of a web service interface defined in WSDL [7]. In contrast to existing approaches, WSDLDiff takes into account the syntax of WSDL and extracts the WSDL elements affected by the changes and the types of changes. From a perspective of social network, Maamar et al. describe how service engineers can capitalize on Web services' interactions, which is to build social networks for service discovery [8]. Recently, Wuhui Chen et al. propose connecting the isolated service islands into a global social service network to enhance the services' sociability on a global scale [9]. All of these works aims at improve the utility of Web services from different perspective of relationship and network.

2.2 Reasoning with Uncertainty of Web Service Network

In the last years, some efforts have been made in representing and reasoning with uncertainty of Web service network in the Semantic Web, there is a complete overview about the subject in paper [10]. These works are mainly focused on how to extend the logics behind Semantic Web languages to the probabilistic/possibilistic/fuzzy logics. In particular, some studies focused on the semantic relations reasoning with uncertainty. The performance and reliability of overlay services rely on the underlying overlay network's ability to effectively diagnose and recover from faults such as link failures and overlay node outages. They identify a set of potential faulty components based on shared end-user observed negative symptoms. Then, each potential faulty component is evaluated to quantify its fault likelihood and the corresponding evaluation uncertainty [11]. The paper introduces a fuzzy logic based approach which extends the classic Description Logic with Zadeh semantics to deal with uncertain knowledge about concepts and roles as well as instances of concepts and roles. Uncertain knowledge representation and the reasoning algorithm for consistency checking of a fuzzy knowledge base are addressed in detail. This paper also discusses complexity issues of the reasoning problem [12]. Authors developed a mechanism to distribute the context reasoning problem into smaller parts in order to reduce the inference time and described a distributed peer-to peer agent architecture of context consumers and context providers. They explain how this inference sharing process works, partitioning the context information according to the interests of the agents, location and a certainty factor [13]. An approach was presented to reasoning with uncertainty information in the Semantic Web [14]. The authors have applied Markov Logic, which is able to reason with uncertainty information, to several Semantic Web ontologies, showing that it can be used in several applications. Aiming at the complicated semantic matching problem of service's input and output data items in the service composition process, a pre-computed relation matrix for Semantic Web Services is presented [15]. Based on the semantic relation reasoning method of concept in Domain Standard Ontology, the matrix based automatic service composition algorithm is proposed. In paper [16], a semantic web service modeling approach based on dynamic description logic is proposed. The logic describes dynamic aspects such as the states of the world and the changes of the state the services caused. By using the description logic reasoning, the algorithm for checking the web services reliability is represented. Optimizing task scheduling in a distributed heterogeneous computing environment plays a critical role in boosting Quality of Service (QoS). The paper considers four conflicting objectives to develop a comprehensive multi-objective optimization model for task scheduling and evaluate the model by applying two multi-objective evolutionary algorithms [17].

All the works on uncertainty and vagueness in the Web service network expressed by Semantic Web technology rely on the principle that the uncertainty or vagueness of the ontology is already asserted. To our knowledge, there is no work on extracting this information automatically from knowledge representations, such as the semantically structured Web service network. However, there are many semantic relationships information that are uncertain or vague by nature, but do not have any type of information denoting that fact. This fact leads to the need of developing efficient Web service relationship's evolution and reasoning mechanisms to process this uncertain information.

3 Problem Definition and Preliminaries

3.1 Problem Definition

The vision of the S-SLN is that of an intelligent network of Web services that computers can understand and reason. However, Web services relationships represented by semantic links in S-SLN, in most of the cases, are characterized by uncertainty. Just as we are often uncertain as to semantic association structure, we may not have a perfect idea of what semantically implies what. The uncertainty in S-SLN can be identified as follows:

(1) The context of Web service is uncertain. In some situations there is some type of semantic relationship between services, but this is not always possible (if the service cannot be executed on available time), so the service semantic relationships is uncertain because of the uncertain context of Web services.

(2) The web service semantic relationship is incomplete. In an open service environment, the initial service semantic relationships in S-SLN are incomplete. There are some implicit semantic relations between services which can be obtained by semantic relation reasoning during its evolution process.

(3) The semantic relationships evolution reasoning rules in S-SLN have the vagueness. There is only a limited knowledge about the service reasoning rules, being very difficult to exactly describe the service relations reasoning or predicting its future behavior.

It is always essential but difficult to reason with incomplete, partial or uncertain knowledge when using S-SLN to support an intelligent service application. How can evolution model and evolution relationship reasoning be done with this uncertainty information? This fact leads to the need of developing efficient Web service evolution model and evolution relationships, reasoning mechanisms to learn this information.

Graphical models (Markov networks) are an important subclass of statistical models that possess advantages that include clear semantics and a sound and widely accepted theoretical foundation (probability theory). Graphical models can be used to represent efficiently the joint probability distribution of a domain. They have been used in numerous application domains, ranging from discovering gene expression pathways in bioinformatics to computer vision. Naturally, statistical analysis is one of the widely adopted approaches; In our problem, we believe that the structure of S-SLN represents the conditional independence relationships among the Web services. S-SLN is directed acyclic graph in nature in which the nodes represent Web services (variables), the links (arcs) signify the existence of direct causal influences between the linked variables, and the strengths of these influences are expressed by forward conditional probabilities. Web service semantic communities are associated with some particular Web service nodes in S-SLN. The semantic community for a Web service node is active as long as that node is active, although the membership may change. Hence, the events that we can observe for Web service semantic community over time are different from cluster evolution.

Based on this, one promising approach to Web service relationship evolution and reasoning with uncertainty is event-based evolution model and Markov Logic based graph model. We use the critical events to measure different types of evolutionary behavior, in Markov logic there is no right and wrong world, there are multiple worlds with different degrees of probability. Markov Logic is based on first-order logic and probabilistic graphical models to deliver the probability of a given logic formula. This type of logic has been applied to several application domains and has shown to be robust and able to deal with uncertain knowledge. In our work, we are studying how we can predict Web service evolution relationship about uncertainty in SSLN.

A graph G of S-SLN is said to be evolving if its relationship between Web services vary over time. Let $G=(V,E)$ denote a temporally varying Web service relationship graph where V represents the total unique Web services and E the total relationships that exist among the Web services. We define a temporal snapshot $S_i=(V_i,E_i)$ of G to be an active relationships Web services and its relationships active in a particular time interval $[T_{si},T_{ei}]$, called the snapshot interval. We are interested in understanding how Semantic Community of Web service evolves over time, in particular identifying key changes that occur and discovering the motivations for these changes. For each snapshot S_i, there can be a set of $|N_i|$ Semantic Communities of the Web service represented as $N_i=\{N_i^1,N_i^2,...,N_i^{ki}\}$. To characterize the changes occurring in Semantic Communities of the Web service over time, we require the use of certain measures on critical evolution events. These events capture the behavior of Web services and its semantic community over time. Here, we consider Web service semantic community and make use of the importance and depth information to quantify key changes.

3.2 Preliminaries

First-order logic. First-order logic builds a more expressive language in the foundations of propositional logic. This new language is designed to create accurate real world models, characterized by a large number of objects, with relations between them. A first-order knowledge base (KB) is a set of sentences or formulas in first-order logic [18]. Formulas are constructed using four types of symbols:

- Constant symbols represent objects in the domain of interest.
- Variable symbols range over the objects in the domain.
- Function symbols represent mappings from tuples of objects to objects.
- Predicate symbols represent relations among objects in the domain or attributes of objects.

A term is any expression representing an object in the domain. It can be a constant, a variable, or a function applied to a tuple of terms. An atomic formula or atom is a predicate symbol applied to a tuple of terms. A ground term is a term containing no variables. A ground atom or ground predicate is an atomic formula all of whose arguments are ground terms. Formulas are recursively constructed from atomic formulas using logical connectives and quantifiers. A positive literal is an atomic formula; a negative literal is a negated atomic formula. A KB in clausal form is a conjunction of clauses, a clause being a disjunction of literals. Every KB can be converted to clausal form. A possible world or Herbrand interpretation assigns a truth value to each possible ground atom. In finite domains, first-order KBs can be propositioned by replacing each universally (existentially) quantified formula with a conjunction (disjunction) of all its groundings.

Markov network. Given a set of random variables X, a *Markov network* (or *Markov random field*) is an undirected graph whose node set is X and whose edge set enco.s conditional independence assertions [19]. It is composed of an undirected graph G and a set of potential functions ϕ_k. The graph has a node for each variable, and the model has a potential function for each clique in the graph. A potential function is a non-negative real-valued function of the state of the corresponding clique. The joint distribution represented by a Markov network is given by

$$P(X=x)=\frac{1}{Z}\prod_k \phi_k(x_{\{k\}}) \quad \textbf{(1)}$$

where $x_{\{k\}}$ is the state of the kth clique (i.e., the state of the variables that appear in that clique). Z, known as the *partition function*, is given by

$$Z=\sum_{x\in X}\prod_k \phi_k(x_{\{k\}}) \quad \textbf{(2)}$$

Markov networks are often conveniently represented as *log linear models*, with each clique potential replaced by an exponentiated weighted sum of features of the state, leading to

$$P(X=x)=\frac{1}{Z}\exp(\sum_j w_j f_j(x)) \quad \textbf{(3)}$$

Instead of potential functions, there are features $f_j(x)$, each one with an associated weight w_j. A feature is a real-valued function of a state. Each feature can represent more than one state of a clique, being this way a more compact representation than potential functions. If a feature is represented in each state of each clique, the log-linear model can be easily translated to the potential function model, and vice-versa. A Markov network of a domain is an undirected graph model that can be used (1) to represent the joint probability distribution of the domain, and (2) to represent the set of conditional independences present in the domain. Each node in the graph represents one of the random variables in the domain, while the absence of edges encodes conditional independence.

Markov logic network. Markov Logic Network (MLN) combines first-order logic and probabilistic graphical models Markov networks in the same representation [20]. Different to hard constraints of a first-order, the basic idea in MLNs is to soften these constraints: when a world violates one formula in the KB it is less probable, but not impossible. The fewer formulas a world violates, the more probable it is. This is done by attaching weights to first-order logic formulas: the higher the weight, the bigger is the difference between a world that satisfies the formula and one that does not, other things being equal. These sets of weighted formulas are called Markov Logic networks (MLNs). Given a set of constants of the domain and an interpretation, the groundings of the formulas in an MLN can generate a Markov network by adding a variable for each ground atom, an edge if two ground atoms appear in the same formula, and a feature for each grounded formula. The probability distribution of the network is defined as

$$P(X=x)=\frac{1}{z}\exp(\sum_{j}^{F} w_i n_i(x)) \quad \textbf{(4)}$$

$$=\frac{1}{z}\prod_i \phi_i(x_{\{i\}})^{n_i(x)} \quad \textbf{(5)}$$

Where F is the number of formulas in the MLN, $n_i(x)$ is the number of true groundings of F_i in the world x, and w_i is the weight of F_i. $x_{\{i\}}$ is the state (truth values) of the atoms appearing in F_i. As formula weights increase, an MLN increasingly resembles a purely logical KB, becoming equivalent to one in the limit of all infinite weights.

Markov logic brings the power of probabilistic modeling to first-order logic by attaching weights to logical formulas and viewing them as templates for features of Markov networks. This gives natural probabilistic semantics to uncertain or even inconsistent knowledge bases with minimal engineering effort. From the point of view of probability, MLNs provide a compact language to specify very large Markov networks, and the ability to flexibly and modularly incorporate a wide range of domain knowledge into them. From the point of view of first-order logic, MLNs add the ability to soundly handle uncertainty, tolerate imperfect and contradictory knowledge, and reduce brittleness. Many important tasks in statistical relational learning, like collective classification, link prediction, link-based clustering, social network modeling, and object identification, are naturally formulated as instances of MLN learning and inference.

4 Event-Based Evolution Model in S-SLN

4.1 Events for Web Service Relationship Evolution

We measure some Web service relationship evolution event base in the semantic community of Web service. Therefore, different from the graph-based community notion, a semantic community of Web service consists of two parts: structure and semantics. Semantic community can be defined from the structure and reasoning point of view as follows:

Definition 1. *Semantic community of Web service*

Semantic community of Web service SC=<ServiceSet,LinkSet>, where ServiceSet is the set of Web service; LinkSet is the set of service semantic link.

We treat each community as a multinomial distribution over Web service in S-SLN. Each Web service WS is associated with a conditional probability P (WS |SC) which measures the degree that WS belongs to the service community SC. The goal is therefore to find out the conditional probability of a Web service given each service community.

Based on this, we use some event concept presented in work [21] and to define several basic events for the semantic community of Web service.

Growth: This event captures the size of the semantic community of the Web service increasing over time.

$$Growth(N_i^k)=1 \, iff \left|V_i^k\right|<\left|V_{i+1}^k\right| \quad \textbf{(6)}$$

Growth in an S-SLN indicates that more Web service nodes are invested in the semantic community of a particular Web service node.

Shrinkage: This event signifies the reduction of the size of a Web service node's semantic community.

$$Shrinkage(N_i^k)=1 \, iff \left|V_i^k\right|>\left|V_{i+1}^k\right| \quad \textbf{(7)}$$

Shrinkage can be caused by either relationship deletions among Web services in the immediate semantic community of a Web service node or the deletion of an influential hub node in its semantic community.

Continuity: A Web service semantic community is said to continue if the members of the community do not change. Note that, this does not place any restrictions on the semantic relationship link structure among the Web service semantic community members.

It conveys the information that the Web service nodes invested in this particular semantic community remain unchanged.

$$Continuity(N_i^k) = 1 \, iff \left|V_i^k\right| = \left|V_{i+1}^k\right| \quad \textbf{(8)}$$

Since the evolution of an S-SLN graph does not uniformly affect all nodes, this event serves the purpose of determining the range of such changes within the graph. If a node's semantic community satisfies a Continuity event, it demonstrates that the changes occurring in the graph do not affect this particular Web service node in any way. This is a stability measure of the relationship of the Web service node.

Mutate: This event indicates major changes within the semantic community of a Web service node. If more than half of the members of a node's community are different over two successive snapshots, it indicates a significant change in the community and hence can be considered a Mutate event.

$$Mutate(N_i^k) = 1 \, iff \left|V_i^k \cap V_{i+1}^k\right| < 0.5 * \left|V_i^k\right| \quad \textbf{(9)}$$

Using this event, one can identify Web service nodes whose semantic community are affected severely by changes occurring in the graph over time. A node's sociability can be quantified based on the number of Mutate events, the node participates in.

K-Attraction: This event signifies positive change in the semantic community of a Web service node with k% of the nodes moving closer than before. Let $Dep(m)_i^k$ represent the depth (minimum distance from the Web service) of node m in the semanthe timecImmunity of k at time i.

$$S = \{m \in V_i^k \,\big|\, Dep(m)_i^k > Dep(m)_{i+1}^k\} \quad \textbf{(10)}$$

$$Att(N_i^k, k) = |S| \, iff \, (|S| > k * \left|V_i^k\right|) \quad \textbf{(11)}$$

If a Web service node experiences this event, it reflects positively on the influence of that particular Web service node. For example, some Web service nodes are closer to a particular Web service node than previously. This signifies an Attraction event on the part of the particular Web service node.

K-Repulsion: This event signifies the increase in distance between a Web service node and the members of its semantic community in the previous timestamp. If *k*% of the Web service nodes in the semantic community of a Web service node WS is farther away in the next timestamp, WS is considered to partake in this event.

$$S = \{m \in V_i^k \,\big|\, Dep(m)_i^k < Dep(m)_{i+1}^k\} \quad \textbf{(12)}$$

$$Rep(N_i^k, k) = |S| \, iff \, (|S| > k * \left|V_i^k\right|) \quad \textbf{(13)}$$

This event demonstrates a negative influence of the Web service node in question. It intrinsically represents the fact that the changes are occurring in the graph as a whole, have an adverse effect on the relations of this Web service node with its semantic communities. The events we describe above are no mutually exclusive. For instance, it is common for a semantic community to undergo a *Growth* event and an *Attraction* event at the same time. To find the events, we consider two snapshots of communities at a time.

4.2 Event-based Evolution Model

Event-based S-SLN network data consist of sets of events over time, each of which may involve multiple Web services. In traditional network analysis techniques, such as social network models, often aggregate the relational information from each event into a single static network. In contrast, we focus on the temporal nature of such Web service data, event data is inherently temporal, with a timestamp or fixed time interval associated with each event. In particular, we look at the problems of temporal Web service relationship evolution prediction and evolution relationship reasoning classification, and describe model based on data mining and machine learning techniques in this context.

From a data analysis and data mining perspective, there are a number of different questions that can be asked in this context, including questions about how the S-SLN evolve over time, the emergence of Web service communities, and so forth. In this subsection, we set the Web service evolution model related to event based Web service network data:

- Predicting future event co-participation of Web services: how likely is it that a given pair of Web services will co-participate in at least one event over some specific future time period?
- Relationship evolution: how does the relationship of each pair of Web services change over time in response to participation in a series of events?

We have argued that an event-driven model may be a more general and natural way to model Web service network evolution, We use an event-based framework for modeling Web service network evolution. We use the model found in [22], which satisfies all of the requirements stated above. We define the basic model as follows:

We denote the potential of participant Web service $v \in V$ in a graph $G = (V, E)$ of S-SLN at time t_i by $R_i(v)$, which takes on values in the interval (0, 1). $R_0(v) = \frac{1}{n}$, and in general $R_i(v)$ is recursively defined as

$$v \in P_i: \quad R_{i-1}(v) + \alpha_i \cdot \frac{\overline{R}_{i-1}(v)}{\sum_{d \in P_i} \overline{R}_{i-1}(d)} \quad \textbf{(14)}$$

$$v \notin P_i: \quad R_{i-1}(v) \cdot (\frac{\alpha_i}{T_{N_{i-1}}}) \qquad \textbf{(15)}$$

where P_i is the set of Web service participants of event e_i, α_i is the total amount of potential that the event e_i contributes to the participant set of Web service, $\overline{R}(d,t_i)$ is the additive inverse of d's potential, $1-R_i(d)$, and $T_{N_{i-1}}$ denotes the total amount of potential held by the non-participants of e_i, that is, $\sum_{d \notin P_i} R_{i-1}(d)$.

5 Relationship Evolution Reasoning

5.1 MLN-based Semantic Link Network

Similar to any other networks, an S-SLN can be represented as a directed graph with semantic relations which represents a dependency relation between two Web services. So Markov network can be adopted as the backbone framework of S-SLN. Service Markov logic network can be seen as a template to construct service Markov networks from given sets of Web services: each ground atom is a variable of service, logical connectives are the edges between variables of service, and each grounded formula is a feature. The resulting service Markov network gives a probability distribution over the possible worlds, being used to answer any probabilistic query and Web service relationship evolution reasoning about Web services in S-SLN.

Definition 2. *Service Markov Logic Network*

A Service Markov logic network S-MLN is a set of pairs $\{(F_i, w_i)\}$, where F_i is a formula in first-order logic and w_i is a real number. Together with a finite set of Web services $S = \{S_1, S_2, \ldots S_{|S|}\}$, it defines a Service Markov network MN_S as follows:

(1) MN_S contains one binary node for each possible grounding of each predicate appearing in S-MLN. The value of the node is 1 if the ground predicate is true, and 0 otherwise.

(2) MN_S contains one feature for each possible grounding of each formula F_i in S-MLN. The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is the w_i associated with F_i in S-MLN.

Therefore, to apply service Markov logic network in the S-SLN, two objects are needed: first-order formulas and their weights. Formulas can be acquired by interpreting the semantic relations (semantic links) among services in the S-SLN as sets of first-order formulas. Weights can be acquired by obtaining the probabilities of edges in the directed graph, we can interpret these probabilities as weights and perform reasoning. A service semantic link is an ordered relation between two Web services. It can be represented as a pointer with a type directed from one service (predecessor) to another service (successor). Types of semantic links can be defined according to the specific Web service application domain. Six types of first-order predicates for event semantic inks are defined as shown in Table 1. And the basic event semantic links evolution reasoning rules in S-SLN are defined as shown in Table 2. It is essential but difficult to reason with incomplete, partial or uncertain knowledge when using service semantic link network S-SLN to support an intelligent advanced service application, i.e., navigation on S-SLN. We desire that navigation in our approach is on-the-fly transformation of the underlying service network in such a way that users can focus on one or more services in the S-SLN, and immediately see a conceptual summary of their focus, in the form of transformed reduced service link network, in which unrelated service will be pruned, but not removed completely, and highly relevant services will be brought to the user's attention even if they were not explicitly linked to current user's focus.

Table 1. First-order predicates for semantic inks

Predicates	Explanation
Similar (S_i, S_j)	Service S_i similar to S_j
Refer (S_i, S_j)	Service S_i refer S_j
Orthogonal (S_i, S_j)	Service S_i orthogonal with S_j
Invocate (S_i, S_j)	Service S_i invocate S_j
Equal (S_i, S_j)	Service S_i equal to S_j
Member (S_i, S_j)	Service S_i and S_j has a membership

Table 2. Reasoning ruels

Rules
$\forall x,y,z: Similar(x,y) \wedge Similar(y,z) \Rightarrow Similar(x,z)$
$\forall x,y,z: refer(x,y) \wedge refer(y,z) \Rightarrow refer(x,z)$
$\forall x,y,z: refer(x,y) \wedge refer(x,z) \Rightarrow Invocate(y,z)$
$\forall x,y,z: Orthogonal(x,y) \wedge Orthogonal(y,z) \Rightarrow Orthogonal(x,z)$
$\forall x,y,z: refer(x,y) \wedge refer(z,y) \Rightarrow Similar(x,z)$
$\forall x,y,z: Equal(x,y) \wedge Equal(y,z) \Rightarrow Equal(x,z)$

The heuristic rules for connecting different types of links are presented. Any two links with the same type can be added by sequentially connecting their predecessors and successors. S-SLN follows probabilistic graph model semantics, and therefore the semantic link reasoning evolution rules of S-SLN can be interpreted as formulas in first-order logic. The main idea behind this interpretation is that relations correspond to unary predicates, roles to binary predicates, and individuals correspond to Web services. In our case, service semantic links evolution rules can be easily interpreted as first order formulas for learning

S-MLN.

There are two things that can be learned in an S-SLN: parameters (the weights) and structure. Both types are learned from example data. The weight learning solution can be achieved by analyzing one of the weight learning algorithms used in Markov logic. The discriminative weight learning algorithm (M. Richardson and P. Domingos, 2006) maximizes the conditional likelihood of some query predicates taking only in account the evidence atoms X and query atoms Y. This solution is more feasible for our work to construct E-SLN, since it needs fewer individuals and can be applied in any type of first-order logic formula. S-MLN structure learning can start from an empty network or from an existing KB. In our case, we use Bottom-Up structure learning algorithm [23] based on the defined semantic link evolution rules.

5.2 Probabilistic Evolution Reasoning

Evolution reasoning in an S-SLN is to derive the semantic relationship between two Web services by logical reasoning via a series of semantic relationships (links). A semantic link can be appended to an S-SLN once the exact meaning between two services can be derived through logical reasoning. However, the semantic relationships in S-SLN are characterized by uncertainty, and most of the times, this uncertainty comes from the uncertain context of service and incomplete semantic relationship. This uncertainty is usually represented by a probability. One way of dealing with uncertainty is through probability theory, where a degree of belief (i.e., a probability) is assigned to the knowledge in the domain. Probabilistic evolution reasoning is an area that tries to find efficient mechanisms to reason under uncertain knowledge expressed through probability theory. In our work, S-MLN provides a compact and expressive tool to deal with uncertainty and complexity, by joining concepts from probability theory and graph theory in the same representation.

Definition 3. *Probability query*

The probability query consists of two parts:

The *evidence*: a subset E of random variables in the model, and an instantiation e to these variables; The *query variables*: a subset *Y* of random variables in the network. Denoted as $P(Y|E=e)$, that is, the posterior probability distribution over the values y of *Y*, conditional on the fact that *E\,=\,e*. A basic inference task is finding the most probable state of the world given some evidence. This is known as MAP query.

Definition 4. *MAP query*

The MAP query also called most probable explanation (MPE), let, MAP is to find the most likely assignment to the variables in given, denoted as $MAP(W|e)=\arg\max_{\omega} P(\omega,e)$, where $\arg\max_{\omega}$ $P(\omega,e)$ represent s the value of ω for which $P(\omega,e)$ is maximal.

S-MLN can answer arbitrary queries of the form "What is the probability that formula F1 holds given that formula F2 does?" If F1 and F2 are two formulas in first-order logic, S is a finite set of Web service constants including any constants that appear in F1 or F2, and L is an S-MLN, then

$$P(F_1|F_2,L,S)=P(F_1|F_2,M_S) \quad \textbf{(16)}$$

$$=\frac{P(F_1\wedge F_2|M_S)}{P(F_2|M_S)} \quad \textbf{(17)}$$

$$=\frac{\sum_{x\in\chi_{F_1}\cap x\in\chi_{F_2}} P(X=x|M_S)}{\sum_{x\in\chi_{F_2}} P(X=x|M_S)} \quad \textbf{(18)}$$

where χ_{F_i} is the set of worlds where formula F_1 holds.

5.3 Evolution Relationship Prediction

The work described in the previous subsection describes event based and graph based modeling efforts that are focused on dynamic an uncertainty of Web service networks. As we stated before, for event-based networks it is of interest to directly take the temporal and sequential aspect of the data into account. Our approach is to treat it as a relationship-driven evolution prediction classification problem (in which "co-participating" in a service semantic community is one class, and "not co-participating" is the other). The methods used are primarily probabilistic classifiers, which assign a probability to each class conditioned on the values of a set of specified rules, whose nature may vary depending on the relationship set. We formally define this evolution relationship prediction problem as follows:

Evolution relationship prediction is the problem of predicting the existence of a relationship link between two objects based on the relations of the object with other objects. In an S-SLN, service semantic link evolution prediction problem can be described as follows:

Given a service semantic link network $S_N=(S,L)$, where $S=\{s_i\}_{i=1}^{n}$ is the Web service set, L is the semantic link set in S_N , service semantic link prediction problem is to predict the possibility of the semantic link l_{ij} between any services s_i and s_j , $l_{ij}\notin L$. Typically, the link prediction includes the existence prediction of semantic link and the classification prediction.

The main task in link-based classification is to predict the category of an object based on the relations of that object with other objects. In our case, we focus on the prediction of the service set category based on the service semantic link in an S-SLN. Service relationship evolution reasoning rules are used for chaining the relevant semantic links and obtaining the reasoning result from the chaining. Service relationship evolution classification prediction rules are used for obtaining the classification result based on reasoning. The reasoning rules of service semantic links in Table 1 are introduced as heuristic rules for supporting reasoning for link prediction. In order to obtain a good evolution reasoning result, the evolution reasoning mechanism should find the strongest link between the candidate links. So we need to obtain the weight of service set classification predication rules. The service set classification prediction rules and its corresponding first-order logic formulas are shown in Table 3.

Table 3. The classification prediction rules

Rules	Weigh
$\forall x, y : Similar(x, y) \Rightarrow Membership(x, y)$	1.99
$\forall x, y : Invocate(x, y) \Rightarrow Membership(x, y)$	0.82
$\forall x, y, z : refer(x, y) \wedge refer(x, z) \Rightarrow Membership(y, z)$	1.28
$\forall x, y, z : refer(x, y) \wedge refer(z, y) \Rightarrow Membership(x, z)$	0.29

These rules state that the class of a service is influenced by the semantic relations among other services. The weight of the rule was learned generatively in conjunction with the reasoning rules of the previous section. Inference and learning algorithms are similar to those used for standard MLN. In particular, the MC-SAT algorithm [20] can be applied for performing probabilistic queries in Web service relationship evolution reasoning.

6 Experimental Results and Analysis

6.1 Evolutionary Measures

Our measure of event evolution score considers the most common evolution components: sociability of Web service nodes in a community, popularity of Web service nodes are attracted to the node's community neighborhood. Thus, we attempt to compare event based Web service relationship evolution scoring functions, we can use the events described in the previous subsection to build evolutionary measures to signify key evolution patterns that occur over time.

Sociability: Sociability is a measure of how many different Web service nodes are affected by or cause effect to this particular node over time.

$$Soc(x) = \frac{\sum_{i=1}^{T} Mutate(N_i^x)}{|Act(x)|} \quad \mathbf{(19)}$$

where $Act(x)$ denotes the number of pairs of successive timestamps this particular Web service node is active in. It is calculated as the ratio of the number of timestamps its neighborhood changes drastically to the number of pairs of successive timestamps it is active in. The key institution behind this is that if two Web services have high sociability, and they have not yet collaborated (not been linked together), there is a high chance they will. This behavior can be captured by the number of Web services join and leave a semantic community that the Web service is involved in. We set the threshold of Sociability at 0.65 to predict the Web services which have scores greater than 0.65 and which have not been linked together in the past.

Popularity: Popularity is a measure of how many Web service nodes are attracted to the node's neighborhood. It can be described using the ¶-Attraction and ¶-Repulsion events.

$$Pop(x) = \frac{\sum_{i=1}^{T} Att(N_i^x, k) - Rep(N_i^x, k)}{|Act(x)|} \quad \mathbf{(20)}$$

If a Web service attracts several nodes over time and does not have high repulsion rates, it is considered popular. This is a measure of attraction of Web services to a semantic community over the course of a time interval. And this also is an influence measure of a community, which does not reflect the size of the service semantic community. However, larger communities have a higher propensity of attracting new Web service to contribute to their popularity.

The proposed framework was applied to represent the evolution relationships in the S-SLN and to describe evolution events over time of involving multiple Web service participants from an evolution analysis perspective. We used Andreas Hess data set Collection of categorized Web Services [24] for our experiment. The data set gathered a corpus of 424 Web Services from SALCentral.org, a Web Service index. The Web services were manually classified into taxonomy and were arranged the categories as a hierarchy. The directory structure serves as the label, i.e. a WSDL file in the communication\mail directory was classified as a "mail" web service, where "mail" is a subclass of "communication". However, we used only the 5 categories: Business (22), Communication (44), CountryInfo (62), Money (54), and News (30), 212 Web Services in our experiments. We manually created a service semantic link network composed by 212 Web services and randomly removed the classification information to 30% of the services.

We use two main components of evolutionary measures on the above dataset by adding and removing some relationship links between Web services manually. In the case of Andreas Hess dataset, popularity reflects a buzz around a particular Web service, as more Web service node and relationships are added to it. We computed the popularity for semantic communities of Web services (as $Att(N_i^x, k) - Rep(N_i^x, k)$) at different adding and removing operation. We identified evolution events and analyzed the evolution relationship.

Although the data set is small and the process of evolution is manually created. This result suggests that the events and evolutionary measures are helpful for evaluating Web service evolution relationships.

6.2 Evolutionary Relationship Prediction

The goal was to observe the framework's behavior under conditions like missing or incomplete Web service relationship and Web service classification. The result from the experiment was an evolution reasoning and prediction process comprising Web service evolution relationship reasoning with relationships classification prediction after the basic Web services were added to the S-SLN with no relations. Additionally, for each of the evolution reasoning concerning the missing relationships of the Web services discovered within the evolution process.

What our mainly focus is that whether the resulting S-SLN could represent the true dependency relationship between services accurately. So the accuracy of the relationship reasoning and prediction classification is evaluated experimentally. With the rules of Table 2 and Table 3, we used MC-SAT to predict the membership of the missing classification Web services based on the evolution event. The accuracy is defined as follows:

$$P_s = \frac{N_r}{N_s} \quad \textbf{(21)}$$

$$R_s = \frac{N_r}{Q_s} \quad \textbf{(22)}$$

$$F_{measure} = \frac{2 * P_s R_s}{P_s + R_s} \quad \textbf{(23)}$$

Where P_s represents the precision of relationship evolution classification, N_r is the number of correct classified Web services, N_s is the number of Web services for classification, R_s represent the precision of recall, Q_s is the number of all relevant Web services, $F_{measure}$ combines precision and recall.

The experimental results are shown in Figure 1 to evaluate the quality of service set classification by S-MLN based evolution relationships prediction. It can be seen that there is a reciprocal relationship between the precision and recall. The service class with the small number of Web services, has a high recall, but instead has a relatively lower precision, while the F1value of each class tends to be closer. And the dependency relations between the service class also affected the experimental results. It is also worth to note that Recall reports the lower performance. This may be due to the small dataset which does not benefit the accuracy of recall. Generally, above experimental results and analysis show our proposed method for event based Web service relationships evolution and reasoning prediction in S-SLN is efficient.

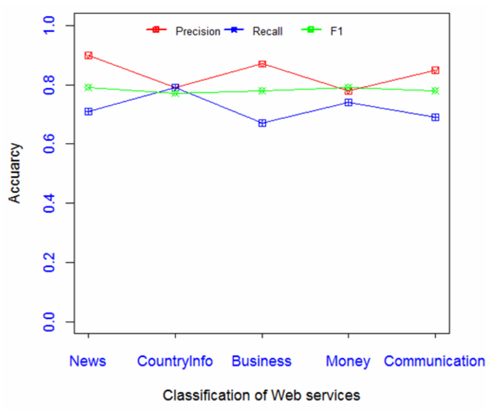

Figure 1. Relationship evolution classification result

7 Conclusion

Understanding and explicitly modeling relationships in S-SLNs are an essential prerequisite for controlling maintenance and evolution of Web services. In this paper, we proposed and explored the use of evolutionary event in the S - SLN and undirected graph structure Markov logic network, a unifying representation of logic and probability, to detect relationship evolution and to predict about uncertain service evolution relationship in the S-SLN. The main motivation is to facilitate S-SLN to reason and predict service evolution relationships with incomplete, partial or uncertain knowledge for supporting an intelligent service application. We have presented the method to construct Web service relationship evolution model by using available probability information and knowledge based on the related event and their inherent service relationship dependencies. And S-MLN was taken as a logical framework for Web service relationships evolution reasoning with uncertainty to discover and predict evolutionary service relationship classification.

Our framework makes assumptions that some Web service relationship evolution event can be measured in the semantic community of Web service. Therefore, the optimality of the semantic communities will have an impact on the efficacy of the results obtained. And semantic community can be deferent from the different structure and reasoning point of view. Semantically structured community discovery methods can be applied to obtain efficient and robust semantic communities for the logic and event-based relationship evolution framework, which is to show how the semantics of the functional properties and the graph structure of the inherent semantic association dependency between services and operations can be incorporated to identify Web services semantic community. While because of the complex interrelations between Web services, the relation between the services will dynamically vary from time to time. This issue is at least worth investigating and will be part of our future work.

Acknowledgments

This research was supported by the Fundamental Research Funds for the Central University [No.2017CDJSK01XK22], the Chongqing Research Program of Basic Research and Frontier Technology [No.cstc2017jcyjAX0106], the Pre-research Project of Chongqing Normal University [No.16XYY20], the Scientific and Technological Research Program of Chongqing Municipal Education Commission [No.KJ1600306].

References

[1] Y. Sun, R. Bie, X. Yu, S. Wang, Semantic Link Networks: Theory, Applications, and Future Trends, *Journal of Internet Technology*, Vol. 14 No. 3, pp. 365-378, May, 2013.

[2] A. Kabzeva, J. Götze, T. Cottermann, P. Müller, Service Relationships Management for Maintenance and Evolution of Service Networks, *ICSEA 2013: The Eighth International Conference on Software Engineering Advances*, Venice, Italy, 2013, pp. 201-207.

[3] M. Fokaefs, R. Mikhaiel, N. Tsantalis, E. Stroulia, A. Lau, An Empirical Study on Web Service Evolution, *2011 IEEE International Conference on Web Services*, Washington, DC, 2011, pp. 49-56.

[4] M. Fokaefs, E. Stroulia, WSDarwin: Studying the Evolution of Web Service Systems, *Advanced Web Services*, A. Bouguettaya, Z. Sheng, F. Daniel, eds., pp. 199-223, Springer, 2014.

[5] L. Fan, J. Tang, Y. Ling, B. Li, Dynamic and Quantitative Method of Analyzing Service Consistency Evolution Based on Extended Hierarchical Finite State Automata, *The Scientific World Journal*, Vol. 2014, http://www.hindawi.com/journal/tswj/2014/79327/

[6] M. Song, Z. Wei, G. Yin, J. Ni, H. Wang, D. Liu, Evolution-based Semantic Relationships in Internet-ware Architecture, *2012 Sixth International Conference on Internet Computing for Science and Engineering*, Henan, China, 2012, pp. 12-19.

[7] D. Romano, M. Pinzger, Analyzing the Evolution of Web Services using Fine-Grained Changes, *2012 IEEE 19th International Conference on Web Services (ICWS)*, Honolulu, HI, 2012, pp. 392-399.

[8] Z. Maamar, N. Faci, L. Wives, Y. Badr, P. Santos, J. P. M. de Oliveria, Using Social Networks for Web Services Discovery, *IEEE Internet Computing*, Vol. 15, No. 4, pp. 48-54, July-August, 2011.

[9] W. Chen, I. Paik, P. C. K. Hung, Constructing a Global Social Service Network for Better Quality of Web Service Discovery, *IEEE Transactions on Services Computing*, Vol. 8, No. 2, pp. 284-298, March, 2015.

[10] T. Lukasiewicz, Uncertainty in the Semantic Web, *Lecture Notes in Computer Science Volume 5785*, L. Godo, A. Pugliese, eds., Springer-Velrlag, 2009, pp. 2-11.

[11] Y. Tang, E. Al-Shaer, K. Joshi, Reasoning under Uncertainty for Overlay Fault Diagnosis, *IEEE Transactions on Network and Service Management*, Vol. 9, No. 1, pp. 34-47, January, 2012.

[12] J. Zhao, H. Boley, W. Du, A Fuzzy Logic Based Approach to Expressing and Reasoning with Uncertain Knowledge on the Semantic Web, *Studies in Computational Intelligence Volume 399*, K. Madani, C. A. Dourado, A. Rosa, J. Filipe, eds., pp. 167-181, Springer-Verlag, 2012.

[13] A. Almeida, D. López-de-Ipina, A Distributed Reasoning Engine Ecosystem for Semantic Context-Management in Smart Environments, *Sensors 2012*, Vol. 12, No. 8, September, 2012, pp. 10208-10227.

[14] P. Oliveira, P. Gomes, Learning and Reasoning about Uncertainty in the Semantic Web, *Proceeding(s) of the AICIT Symposium(Conference) on Internet Technologies*, Madurai, India, 2009, pp. 13-22.

[15] L. Hu, S. Ying, K. Zhao, Towards a Precomputed Relation Matrix for Semantic Web Service, *Communications in Computer and Information Science, Vol. 134*, R. Chen, ed., pp. 265-270, Springer-Verlag, 2011.

[16] G. Shen, Z. Huang, J. Yang, X. Zhu, Reasoning about Web Services with Dynamic Description Logics, *2009 WRI World Congress on Computer Science and Information Engineering*, Los Angeles, CA, 2009, pp. 106-110.

[17] F. Ramezani, J. Lu, J. Taheri, F. K. Hussain, Evolutionary Algorithm-based Multi-objective Task Scheduling Optimization Model in Cloud Environments, *World Wide Web*, Vol. 18, No. 6. pp. 31737-1757, November, 2015.

[18] M. R. Genesereth, N. J. Nilsson. *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann, 1987.

[19] Pearl, J., Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.

[20] M. Richardson, P. Domingos, Markov Logic Networks, Machine Learning, Vol. 62, No. 1-2, pp. 107-136, February, 2006.

[21] S. Asur, *A Framework for the Static and Dynamic Analysis of Interaction Graphs*, Ph. D. Thesis, Ohio State University, Columbus, OH, 2009.

[22] J. O'Madadhain, P. Smyth, EventRank: A Framework for Ranking Time-varying Networks, *Third International Workshop on Link Discovery (LinkKDDa¸r′05)*, Chicago, IL, 2005.

[23] L. Mihalkova, R. J. Mooney, Bottom-uP Learning of Markov Logic Network Structure, *Proeeedings of thwenty-Fourth Intemational Conference on Machine Learning*, Corvallis, OR, 2007, pp. 625-632.

[24] Andreas Hess, http://andreashessinfo/projects/annotator/ws2003.html.

Biographies

Yu Yu received her Ph.D. degree of Logics in 2010. She is a lecturer in the School of Public Affairs, Chongqing University, China. Her main research interests include Artificial Intelligence, Semantics and Language logic.

Anping Zhao received his Ph.D. degree of Artificial Intelligence in 2011. Now, he is working in the School of Computer and Information Science, Chongqing Normal University, China. His main research interests include Machine learning and Web mining.