Association-based dynamic computation of reputation in web services

Pradeep K. Atrey*, M. Anwar Hossain and Abdulmotaleb El Saddik

Multimedia Communications Research Laboratory
University of Ottawa
800 King Edward
Ottawa, Ontario, K1N 6N5, Canada
E-mail: patrey@mcrlab.uottawa.ca
E-mail: anwar@mcrlab.uottawa.ca
E-mail: abed@mcrlab.uottawa.ca
*Corresponding author

Abstract: Web services are usually selected and composed based on their reputation. In general, the reputation of a web service is computed using the feedback provided by the user. However, the users' feedback bears many problems, including a low incentive for providing ratings and a bias towards positive or negative ratings. In this paper, we propose a method that overcomes the dependency on users' feedback for computing the reputation of web services. The proposed method dynamically computes the reputation of a web service based on its association with other web services. The association coefficient between any two web services is computed by utilising the statistics of how often they have been composed together. This factor is considered in our method to show the evolution of reputation over a period of time. The experimental results demonstrate the utility of the proposed method.

Keywords: web services; reputation evolution; association coefficient.


Biographical notes: Dr. Pradeep K. Atrey is a Post-Doctoral Research Fellow at the Multimedia Communications Research Laboratory, University of Ottawa. He obtained PhD in Computer Science from National University of Singapore in 2006. He was a Lecturer at the Delhi College of Engineering, University of Delhi and at the Deenbandhu Chhotu Ram University of Science and Technology, Murthal in India from 1992 to 2002. His research interests include Multimedia Surveillance and Security, Smart Environment, and Web. He has authored and co-authored more than 30 refereed journal and conference papers.

M. Anwar Hossain received the BEng. in Computer Science and Engineering from Khulna University, Bangladesh. After working a few years in industry, he has returned to academia and completed his MCS in Computer Science from the School of Information Technology and Engineering at the University of Ottawa, Canada, in 2005. He is currently pursuing his PhD from the same university. His research interests include service-oriented architecture, multimedia smart environment, quality of information and ambient intelligence.
1 Introduction

Web services are an integral part of the World Wide Web. They are created and published by the service providers, and are consumed by the service requesters/consumers (e.g., application developers) by invoking the operations they provide, Dustdar and Schreiner (2005). First the service requesters usually select web services for specific subtasks from a large pool, and then compose them in order to accomplish the overall task. For example, a web portal responsible for managing tours (overall task) can select individual web services for various subtasks such as for air ticket booking, hotel booking, currency conversion, and car rental, etc. For the selection of these web services for individual subtasks, various criterion are considered. The most common is the reputation of the service. A web service with a higher reputation is preferred over a service, which has a lower reputation in the composition, Derbas et al. (2005).

The reputation of a web service represents a general opinion about how good a service has been rated based on its various characteristics. In general, it is computed based on a service requesters’ feedback/rating, Liu et al. (2004). Most reputation systems use various mechanisms for aggregating these ratings in order to compute the reputation of the service provider Sherchan et al. (2006). The attributes contributing to the rating may be generic as well as domain specific Maximilien and Singh (2002). The aggregation mechanisms include weighted averaging, Bayesian methods, discrete trust models, belief models, fuzzy models, and flow models. Each of these has advantages and disadvantages. For details, the readers are advised to refer to Josang et al. (2007).

The determination of the reputation of a web service, based on a service user’s rating, bears many problems including the low incentive for providing ratings and the bias towards positive rating Sherchan et al. (2006). In general, the lack of reward or incentive restrict users to provide their feedback and comments. However, some works advocate for incentives in terms of money to the users for providing feedback. For instance, Jurca and Faltings (2006) addressed the issue of optimal payment that should be made to the users for providing honest reputation feedback. In another work, Miller et al. (2005) presented a peer-prediction method that takes advantage of the relevance between the reports of different raters, based on which honesty of the reports is evaluated and appropriate rewards for the raters are suggested accordingly. In spite of these methods, the payment based incentives for collecting reputation feedbacks has different practical constraints such as how to make the payment and the greediness of the users to earn more by providing numerous feedback with different online identities. Moreover, even with the payment, the ratings provided by the service users remain to be subjective in many cases.
Sometimes, the users provide negative feedback without even knowing much about the service or its usage. In a nutshell, the existing methods of reputation computation have a high degree of dependency on the service users’ feedback, which is not always available. This motivates us to develop a method that overcomes this dependency.

In this paper, we propose a method for the dynamic computation of the reputation of a web service. The proposed method was earlier introduced with the preliminary results in the context of multimedia service composition in Atrey et al. (2008). Here, we describe our method with a detailed explanation and the corresponding experimental results. In our method, the reputation of a web service is computed based on its association with other services. The measure of association between web services is computed based on how often they have been composed together. To illustrate the core idea of our method, we provide an example as follows. A company, ABC Pvt. Ltd., has a good reputation and a product branded as ‘ABC’. Now, a new company, XYZ Pvt. Ltd., is associated with it and sells its product with a brand name ‘XYZ-ABC’. By virtue of being associated with a reputed company, the new company XYZ Pvt. Ltd. will also start gaining a good reputation, and the reputation of ABC Pvt. Ltd. can also evolve positively over a period of time. Adopting the same analogy, in our method, if a ‘not-so-reputed’ web service is associated with a ‘well-reputed’ web service, its reputation will grow more than that of the well-reputed one. Although determining a web service as being not-so-reputed or well-reputed may be quite fuzzy, we use 0.50 as a threshold value to differentiate between them. The services that have a reputation greater than 0.50 are called ‘well-reputed’, and others are known as ‘not-so-reputed’.

The proposed method is different from existing methods in the following two aspects. First, our method does not have a dependency upon the user’s feedback, which is in contrast to the traditional feedback-based approaches. Second, it allows for the dynamic computation of the reputation of new web services, which are added/composed without having any prior information of their reputation.

The remainder of this paper is organised as follows. Section 2 describes related works. In Section 3, we formulate the problem of determining the reputation of a web service based on its association with other web services. We present our proposed method of dynamic reputation computation in Section 4. The results are provided in Section 5. Finally, Section 6 concludes the paper with a discussion on future work.

2 Related work

Reputation has always been a primary issue among the online business communities, Resnicks et al. (2000). The buyers choose sellers or products based on their reputation. Some of the existing reputation computation systems are eBay, Amazon, etc.

In the context of web services, the reputation of services or of the service providers has been used primarily by the web service composers for first selecting and then composing them for various online business operations (Liu et al., 2004). In addition to this conventional usage, the reputation has also been used as a criteria to preserve privacy in the web services (Rezgui et al., 2003).

Various models have been proposed for the online computation of the reputation of business parties involved in electronic transactions. These can be classified into two categories. In the first category, the trading partners or the agents compute, based on
the collaborative ratings, the reputation of each other or the reputation of a common service. We call the models in this category ‘Collaborative-feedback models’. The second category, which we call ‘Independent-feedback models’, is the one in which the reputation of a service is computed based on the independent feedback provided by the user of the service. Collaborative-feedback models include the works of Zacharia et al. (1999), Mui et al. (2002), and Sreenath and Singh (2004). The works of Maximilien and Singh (2002), Namim et al. (2005) and Sherchan et al. (2006) fall in the Independent-feedback models category. We describe the works in these categories in the following paragraphs.

In the Collaborative-feedback models category, Zacharia et al. (1999) presented a collaborative model for computing the reputation of the users in the online electronic marketplace. In this model, the reputation is computed based on the ratings provided by one user to the other user.

Sreenath and Singh (2004) presented an agent-based collaborative approach for computing the reputation of the web services. In this approach, the agents (e.g., the service users) cooperate to evaluate service providers by autonomously deciding how much weight should be given to each other’s recommendations. This method is useful when the collaborating agents have similar needs in terms of service attributes.

Mui et al. (2002) proposed a computational model based on the reinforcing relationship among trust, reputation and reciprocity. The model proposes a probabilistic mechanism for inference among trust, reputation and the level of reciprocity in a multi-agent environment (such as an electronic market). The rationale behind this model is that a higher reputation leads to a higher trust; trust reciprocates actions between two agents; and the reciprocate actions will lead to a higher reputation.

In Independent-feedback models, the feedback on the different service attributes are aggregated to obtain a single reputation score of a service provider. Some of the works in this category are described as follows.

Maximilien and Singh (2002) proposed a conceptual model for computing the reputation of web services. In this model, the reputation of a web service is computed based on its various generic attributes (e.g., service delivery time) as well as the domain-specific attributes (e.g., accuracy and cost of the service). The domain-specific attributes may matter to a specific service user, as different users might have other preferences in terms of attributes. For example, a user, for whom the cost is a more important attribute, may not care about the accuracy of the service provided, which might be a concern to the other user. In this model, the authors also considered the weight of the attributes relative to their domain and the user’s preferences, and the temporal characteristics of the attribute values such as the decay function (e.g., exponential or step) that can be used for decaying the value of an attribute. This model for reputation computation, due to its dependence on users’ feedback, suffers from several problems such as lack of incentives for leaving feedback, general bias towards a positive rating, possible deceptions and collusions, etc.

In another work, Namim et al. (2005) applied a secret sharing scheme for securing the reputation. The reputation is computed based on the feedback from different users and these users are provided shares of a secret that is known only to the reputation computation system. The users of high importance are given more shares. The reputation is considered secure and valid if the feedback from the users, who hold a particular number of shares, is obtained and the secret is reconstructed.
Recently, a fuzzy model proposed by Sherchan et al. (2006) computes the reputation of a web service by using not only the users’ feedback but also the validation of users’ rating behaviour. An analysis of users’ rating behaviour is performed to infer the rationale for ratings. The model analyses the objective performance measures to determine if the subjective view matches the objective view, and combines the two dimensions (objective and subjective) of reputation using a fuzzy approach. Though this model claims to overcome the problem of deception and collusion, it is still dependent on the users’ feedback.

In contrast to the above methods, in which the feedback (Independent or Collaborative) plays an important role in computing the reputation, our method for computing the reputation of web services does not require users’ feedback. In our method, the reputation of a web service is computed based on whether this service has been used or composed with a well-reputed or not-so-reputed service. Moreover, our proposed method allows for a dynamic addition and reputation computation of web services.

The necessity of computing the rank/reputation automatically without using the users’ feedback has also been emphasised in Cho et al. (2005) and Borodin et al. (2005). In these works, the authors have proposed automated mechanisms to rank the web pages based on the link structure of the web. However, for web services, the link structure is different. Web services are usually associated or composed for accomplishing a composite task while the websites can be linked freely as a reference to the content they present.

3 Problem formulation

We consider a typical scenario of provider-consumer in context of the web services, where a consumer (or a user) uses a website to accomplish a task. This task, in order to be accomplished, may require access to various web services. For example, if a user wishes to book a trip, he/she can go to a travel website, which can use different web services for different subtasks, such as for the air ticket, hotel, car rental, currency conversion, etc. Some of these web services may be well-reputed, and others may be not-so-reputed.

We formulate below the problem of determining the reputation of a web service:

- Let us have a web system that consists of web services that can be composed to accomplish a user task, and let \( W = \{ W_1, W_2, \ldots, W_n \} \) be the set of web services where \( n \) is the total number of services in the composition.

- Let \( R_{Wj}(t-1), 1 \leq j \leq n \) be the reputation level of a web service \( W_j \) at time instant \( t-1 \). Note that \( R_{Wj}(t-1) \in [0,1] \).

- Let \( \lambda_{ij}(t) \in (0,1), 1 \leq i,j \leq n \), be the Association Coefficient at time instant \( t \) between the web services \( W_i \) and \( W_j \). The association coefficient between two web services is computed based on whether they have been composed together or not. The details of the computational model of the association coefficient is provided in Section 4.1.

The objective is to update the reputation \( R_{Wj}(t) \) of all the participating web services \( W_j \), \( 1 \leq j \leq n \) after the composition is done at the time instant \( t \).
The reputation $R_w(t)$ of a web service $W_r$ at the time instant $t$ is updated as:

$$R_w(t) = f(R_w(t-1), R_r(t-1), \lambda_{r,j}(t)), \quad r = 1:n, r \neq j.$$

(1)

In Equation (1), the term $f$ is a function that uses the previous reputation of the web services and their degree of association in order to update the reputation. The detail is described in Section 4.2.

In a similar fashion, the reputation of the other web services $W_r, r = 1:n, r \neq j$ is also updated.

4 Proposed method

In this section, we describe our method that dynamically determines the reputation level of a web service based on its association with another web service. This process involves two steps: computing the association coefficient and evolving the reputation level. Section 4.1 describes how the association coefficient between two web services is computed. Next, in Section 4.2, we describe how this association coefficient is used in evolving the reputation of web services.

4.1 Association coefficient computation

The association coefficient between the two web services refers to the measure of their co-occurrence. It is computed based on how often they have been composed together. The more they are composed together, the greater the association coefficient is between them.

In the past, various forms of association coefficients have been used for diverse applications, such as, for term similarity in documents, Kuropka (2005), for texture similarity in images, Partio et al. (2004), for detection of molecular self-association in biological applications, Lee et al. (2003), etc. The various forms of the measures of association include the Cosine measure, Dice’s coefficient, and Jaccard coefficient, Chen et al. (2004). In the context of our problem, these three forms of association coefficients can be modelled as follows.

Let us have the following statistics about the usage and co-usage of the web services $W_i$ and $W_j$. Here, the co-usage refers to that the web services are used together. Let $T_i$ and $T_j$ be the number of times the web services $W_i$ and $W_j$ are used separately for accomplishing a task; and let $T_{i,j}$ be the number of times both the web services $W_i$ and $W_j$ have been used together for accomplishing a task. Using these statistics, the measure of association (i.e., the Jaccard’s coefficient) is computed as:

$$\lambda_{i,j} = \frac{T_{i,j}}{T_i + T_j - T_{i,j}}$$

(2)

where, $\lambda_{i,j} \in [0,1]$ is the association coefficient between web services $W_i$ and $W_j$.

Another measure of association, the Dice-measure, is calculated as:

$$\lambda'_{i,j} = \frac{2 \times T_{i,j}}{T_i + T_j}$$

(3)
and the Cosine-measure is computed as:
\[
\lambda_{i,j} = \frac{T_{\text{ij}}}{\sqrt{T_i \times T_j}}. \tag{4}
\]

The Cosine measure and the Dice’s coefficient have been widely used as a similarity measure between two entities, and the Jaccard coefficient characterises the association strengths of paired data, van Rijsbergen (1979). However, these methods of association coefficient computation suffer from the problem of initialisation. The terms \(T_i\), \(T_j\) and \(T_{\text{ij}}\) are initialised to 0. At the occurrence of the first composition, each of these terms become 1. With \(T_i = 1\), \(T_j = 1\) and \(T_{\text{ij}} = 1\), the association coefficient is computed to be 1 with only one composition, and it takes several compositions to stabilise them to a true value. Stabilising the value of the association coefficient for all the web service pairs could result in a big overhead.

To overcome the aforementioned problem with the existing methods, we propose a linear combination model for computing the association coefficient, which is as follows:
\[
\lambda_{i,j}(t) = \beta \times \lambda'_{i,j} + (1 - \beta) \times \lambda_{i,j}(t - 1). \tag{5}
\]

In Equation (5), \(\lambda_{i,j}(t)\) and \(\lambda_{i,j}(t - 1)\) are the association coefficients between the web services \(W_i\) and \(W_j\) at the time instances \(t\) and \(t - 1\), respectively. The term \(\lambda'_{i,j}\) represents the association coefficient, which is determined only for the current composition (i.e., at time instant \(t\)). The terms \(\beta\) and \(1 - \beta\) are used to assign weights to the current and the past association coefficients, respectively. The term \(\lambda'_{i,j}\) is determined as follows:
\[
\lambda'_{i,j} = \begin{cases} 
1 & \text{if a composition } (W_i, W_j) \text{ exists at time instant } t \\
0 & \text{otherwise}
\end{cases}
\]

In Equation (5), in absence of any prior information, we assume, for \(1 \leq i, j \leq n\), \(\lambda_{i,j}(0) = \epsilon\) (a positive infinitesimal).

Note that the association coefficient is one of the major parameters that influences the growth or decay of the reputation of the web service that is being composed. If the two services are composed together repeatedly over a period of time, their association coefficient will increase. Therefore, the reputation transfer to them from each other will also occur in the same proportion. In the next section, we describe how the association coefficient is used in reputation evolution.

### 4.2 Reputation evolution model

When a web service is composed with other web services, its reputation evolves based not only on the degree of its association with other services, but also on the reputation level of the other services with which it is associated. Therefore, the growth in the reputation of a web service should be proportional to these two factors.

Considering the above mentioned factors, we first propose in Section 4.2.1, an exponential model for updating the reputation of a web service \(W_i\) based on its association with web service \(W_j\). Next, in Section 4.2.2, we will show how this model can be iteratively used to update the reputation of the web service \(W_j\) based on its association with all the other web services in the composition.
4.2.1 Reputation evolution based on the association with one service

The reputation of a web service \( W_j \) based on its association with the web service \( W_i \) is updated by instantiating Equation (1) as follows:

\[
R_{ji}^t(t) = \frac{1}{Z_{ij}(t)} \times R_{ji}^t(t-1) \times e^{\alpha_{t}(ij)}.
\] (6)

In the above equation, \( R_{ji}^t(t-1) \) are the reputation value of the web service \( W_j \) at time instance \( t-1 \), and \( R_{ji}^t(t) \) is its updated reputation due to its association with web service \( W_i \) at the time instance \( t \). The term \( e^{\alpha_{t}(ij)} \) represents the growth factor for the reputation of \( W_j \) due to its association with the web service \( W_i \) at the time instant \( t \). The term \( Z_{ij}(t) \) is a normalisation factor to limit the reputation value within \([0,1]\), and is given as:

\[
Z_{ij}(t) = R_{ji}^t(t-1) \times e^{\alpha_{t}(ij)} + (1-R_{ji}^t(t-1)) \times e^{\alpha_{t}(ij)}.
\] (7)

The rate of growth is determined by the term \( \alpha_{t}(ij) \), which is given by:

\[
\alpha_{t}(ij) = R_{ji}^t(t-1) \times \Delta \lambda_{ij}(t) \times \gamma.
\] (8)

In Equation (8), \( \alpha_{t}(ij) \) consists of three terms, which are described as follows. The term \( R_{ji}^t(t-1) \) is the reputation of the web service \( W_i \) at time instant \( t-1 \). The term \( \Delta \lambda_{ij}(t) \) is the change in the association coefficient between the web services \( W_i(t) \) and \( W_j(t) \), at time instant \( t \). We compute \( \Delta \lambda_{ij}(t) \) as:

\[
\Delta \lambda_{ij}(t) = \lambda_{ij}(t) - \lambda_{ij}(t-1).
\] (9)

In this equation, the value of \( \Delta \lambda_{ij}(t) \) will be positive when the composition of the services \( W_i \) and \( W_j \) occurs (i.e., \( \lambda_{ij} = 1 \)). When \( \Delta \lambda_{ij}(t) \) is positive, the reputation of both the services \( W_i \) and \( W_j \) correspondingly increases. On the other hand, \( \Delta \lambda_{ij}(t) \) will be negative at the time instant \( t \) when there is a disappearance of an association between these two services (i.e., \( \lambda_{ij} = 0 \)). In this case, the reputation of both the services decreases. The rate of growth or decay is also controlled by \( \gamma \). The term \( \gamma \) can hold either of two values (\( r_{\text{growth}} \) and \( r_{\text{decay}} \)) based on whether the change in the association coefficient is positive or negative. The presence of an association would result in a positive change in the association coefficient, and vice versa. The term \( \gamma \) is determined as follows:

\[
\gamma = \begin{cases} 
  r_{\text{growth}} & \text{if } \Delta \lambda_{ij}(t) \geq 0 \\
  r_{\text{decay}} & \text{else } \Delta \lambda_{ij}(t) < 0.
\end{cases}
\]

Note that the rate of change of the reputation is also proportional to the reputation value of the reference web service (e.g., a well-reputed web service). This satisfies the intuition that a not-so-reputed web service can become well-reputed sooner when it is frequently composed (i.e., association coefficient is high) with a highly reputed web service, compared to when it is seldomly composed (i.e., association coefficient is high) with a less reputed web service.

The aforementioned reputation evolution model is based on heuristics. In choosing the heuristics solution, we have resorted to Oommen and Rueda (2005), where, the authors have shown that the quality of a heuristic algorithm is determined by the accuracy of the heuristic function it uses. The heuristics function used in our algorithm provides a reasonable performance, as will be shown in Section 5.
The behaviour of the aforementioned reputation evolution model is depicted in Figure 1. In this simulation, the value of $\gamma$ is taken as one. The plots shown in Figure 1 correspond to different reputation values of the web service $W_i$. The figure shows that the reputation level of a not-so-reputed web service $W_j$ increases sharply when the change in the value of the association coefficient $\lambda_{ij}$ becomes more than 0.50. For example, it can be seen here that the rise in the reputation of the web service $W_j$ is sharper when composed with the web service $W_i$ with a reputation of 0.80, compared to that when composed with the web service $W_i$ with a reputation of 0.20.

**Figure 1** Evolution of the reputation of a not-so-reputed web service $W_j$ with respect to the association coefficient $\lambda_{ij}$, when composed with a well-reputed web service $W_i$ of varying reputation levels (0 to 1) (see online version for colours)

### 4.2.2 Reputation updation based on the association with multiple services

The model described in the previous section shows the reputation evolution of the web service $W_j$ based on its association with web service $W_i$. However, in web service composition, several services are composed together to accomplish a composite task. Assuming that $n$ web services ($W_1, W_2, ..., W_n$) are composed together (as has been stated in Section 3), the reputation of a web service $W_j$ is updated due to the fact that it has been composed with other $n-1$ web services. The iterative updation of the reputation is performed as follows. We begin with the evolution of the reputation of web service $W_j$ based on its association with the web service $W_i$, by using $i = 1$ in the Equation (6), as shown below:
Next, the reputation of the web service $W_j$ is updated iteratively for the remaining services in the composition by using the following equation:

$$R'_w(t) = \frac{1}{Z_{r,j}(t)} \times R'_w(t-1) \times e^{\alpha_{r,j}(t)}.$$  \hspace{1cm} (10)

Note that the Equation (10) is an iterative form of the Equation (6). The final reputation $R'_w(t)$ of the web service $W_j$ at the time instant $t$ is considered as $R'_w(t)$, i.e., when $r = n$ in the Equation (10).

The reputation of all the other web services in the composition is also computed using the steps explained above in Section 4.2.1 and Section 4.2.2.

5 Experimental results

We demonstrate the utility of our association based reputation computation method by presenting results in a test case scenario. The test case scenario is designed as follows. A user performs an online operation for booking a trip. In order to accomplish this task, the user needs to perform four subtasks – air ticket booking, car rental, currency conversion, and hotel reservation.

We developed a web portal consisting of 16 web services. A snapshot of the web portal is shown in Figure 2. These web services are divided into four different groups, with each group having four web services corresponding to a subtask. As shown in Table 1, the web services are denoted as follows: $W_1$ to $W_4$ for the air ticket booking, $W_5$ to $W_8$ for the car rental, $W_9$ to $W_{12}$ for the currency conversion, and $W_{13}$ to $W_{16}$ for the hotel reservation.

In our experiments, the well-reputed web service for each subtask is identified based on the feedback collected from 50 different users. We adopted this strategy due to the fact that we intend to compare the reputation computed using the proposed method with the feedback based approach. Note that, for initialisation, the identification of well-reputed web services can also be performed using alternative approaches such as being based on prior information (e.g., popularity). In absence of any prior information or the difficulties in collecting users’ feedback, the initial reputation can be assumed to be a very low value. Over a period of time, by applying our proposed method, a web service can become well-reputed by virtue of being frequently associated with the other services.

Also note that, although the identification of a well-reputed service is not mandatory for the proposed model to work, we used it for demonstrating the case when a not-so-reputed web service is associated with a well-reputed web service and how their reputation evolves. We have also examined the other cases of association where two not-so-reputed web services are composed together.

Table 1 shows the initial reputation of all 16 web services. Note that the web services with an unknown reputation are treated as not-so-reputed, and are assigned a very low value ($\varepsilon = 0.01$) as their initial reputation. The four well-reputed web services along with the other 12 not-so-reputed web services are provided in our web portal for selection and composition.
In the web portal (Figure 2), the users were required to select a service for each of the subtasks, and compose them together for accomplishing the overall task of booking a trip. The volunteers were requested to use this web portal to select and compose these services for booking their trip through our web portal. They were asked to select one service from each of the four groups for accomplishing the four subtasks described earlier. Over a period of two weeks, 120 volunteers participated in the experiment.
Table 1  Initial reputation of 16 web services

<table>
<thead>
<tr>
<th>Subtask</th>
<th>Web service</th>
<th>Initial reputation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air ticket booking</td>
<td>W₁</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>W₂</td>
<td>0.61 (Well-reputed)</td>
</tr>
<tr>
<td></td>
<td>W₃</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>W₄</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>W₅</td>
<td>0.56 (Well-reputed)</td>
</tr>
<tr>
<td>Car rental</td>
<td>W₆</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>W₇</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>W₈</td>
<td>Unknown</td>
</tr>
<tr>
<td>Currency conversion</td>
<td>W₉</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>W₁₀</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>W₁₁</td>
<td>0.73 (Well-reputed)</td>
</tr>
<tr>
<td></td>
<td>W₁₂</td>
<td>Unknown</td>
</tr>
<tr>
<td>Hotel reservation</td>
<td>W₁₃</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>W₁₄</td>
<td>0.59 (Well-reputed)</td>
</tr>
<tr>
<td></td>
<td>W₁₅</td>
<td>Unknown</td>
</tr>
<tr>
<td></td>
<td>W₁₆</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

We employed our proposed method to first compute the association coefficient between different pairs of 16 web services. The association coefficient is further used to dynamically determine the reputation of all 16 web services (not-so-reputed as well as well-reputed). The values of different factors (ε, β, r\textsubscript{growth} and r\textsubscript{decay}) used in our experiment are provided in Table 2. The results and analysis are provided in the following two subsections.

Table 2  The values of different factors in our experiment

<table>
<thead>
<tr>
<th>Factor</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε</td>
<td>0.01</td>
</tr>
<tr>
<td>β</td>
<td>0.10</td>
</tr>
<tr>
<td>r\textsubscript{growth}</td>
<td>1.00</td>
</tr>
<tr>
<td>r\textsubscript{decay}</td>
<td>0.20</td>
</tr>
</tbody>
</table>

5.1  Association coefficient computation results

As we have divided the services into four groups of four web services each and the user has to choose one service from each group, there would be $\binom{4}{1} \times \binom{4}{1} \times \binom{4}{1} \times \binom{4}{1}$ distinct pairs in total. The association coefficients $\lambda_{ij}$, $1 \leq i, j \leq 16, i \neq j$ are depicted in Figure 3. In the figure, x and y axes represent the web services $W_i$ and $W_j$, respectively; and the z-axis shows the association coefficient $\lambda_{ij}$ between the pairs ($W_i, W_j$). We observe from
Figure 3 that some of the pairs of web services have higher association coefficients than others. For example, the pairs \((W_1, W_8), (W_5, W_6), (W_2, W_{11}), (W_1, W_{13}),\) and \((W_1, W_{14})\) have association coefficients 0.3625, 0.2923, 0.2516, 0.2452, and 0.2256, respectively. Note that, the association coefficients shown in the figure are up to 45 compositions.

Figure 3 Association coefficient between all the pairs of web services at the end of the 45th composition (see online version for colours)

To show how the association coefficient in a specific pair of web services evolves over time, we provide the results of the pair \((W_1, W_8)\) in Figure 4. In Figure 4, a bar perpendicular to the x-axis represents the occurrence of the composition \((W_1, W_8)\).

The absence of a bar on the x-axis denotes that the current composition does not possess the cooccurrence of the web services \(W_1\) and \(W_8\). As can be seen in Figure 4, the association coefficient \(\lambda_{1,8}\) goes up when a pair \((W_1, W_8)\) occurs \((i.e.,\) shown by a bar), otherwise it begins to go down. For example, with the occurrence of three consecutive compositions of \((W_1, W_8)\), the association coefficient \(\lambda_{1,8}\) goes up to 0.4263 after the 46th composition. This validates our association coefficient computation model.

5.2 Reputation evolution results

As described earlier, 120 compositions were performed over a period of two weeks. The reputation of all 16 web services was dynamically computed over this period. In parallel we also requested volunteers to provide ratings for these 16 web services based on their past experience. The reputation was computed using our association-based method and was compared with the reputation computed using the feedback-based method. This comparison for the 16 web services is shown in Figure 5.
Figure 4  Evolution of the association coefficient between web services W1 and W8 over the first 60 compositions (see online version for colours)

Figure 5  Reputation evolution of 16 web services (see online version for colours)

Note:  Association-based method (solid lines) versus feedback-based method (dotted lines).
Figure 5 shows 16 graphs for 16 web services. In each graph, solid and dotted lines depict the association-based method and the feedback-based method, respectively. Note that the four web services ($W_2$, $W_5$, $W_{11}$ and $W_{14}$) out of the 16 web services are pre-assumed to be well-reputed. The graphs show how the reputation of the web services evolve over the 120 compositions.

From the graphs shown in Figure 5, we made the following observations. The reputation computed using the association-based method is quite comparable with the reputation computed using the feedback-based method. A comparison of the average of the reputation obtained based on both the methods is also provided in Table 3. In general, the reputation of not-so-reputed web services is slightly higher in the feedback-based method compared to that in the association-based method.

<table>
<thead>
<tr>
<th>Web service</th>
<th>Association-based method</th>
<th>Feedback-based method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_1$</td>
<td>0.1238</td>
<td>0.1835</td>
</tr>
<tr>
<td>$W_2$</td>
<td>0.8608</td>
<td>0.6898</td>
</tr>
<tr>
<td>$W_3$</td>
<td>0.0485</td>
<td>0.1331</td>
</tr>
<tr>
<td>$W_4$</td>
<td>0.0356</td>
<td>0.0823</td>
</tr>
<tr>
<td>$W_5$</td>
<td>0.8704</td>
<td>0.6860</td>
</tr>
<tr>
<td>$W_6$</td>
<td>0.0691</td>
<td>0.1289</td>
</tr>
<tr>
<td>$W_7$</td>
<td>0.0701</td>
<td>0.1165</td>
</tr>
<tr>
<td>$W_8$</td>
<td>0.0877</td>
<td>0.1385</td>
</tr>
<tr>
<td>$W_9$</td>
<td>0.0681</td>
<td>0.0985</td>
</tr>
<tr>
<td>$W_{10}$</td>
<td>0.0713</td>
<td>0.1165</td>
</tr>
<tr>
<td>$W_{11}$</td>
<td>0.8560</td>
<td>0.8298</td>
</tr>
<tr>
<td>$W_{12}$</td>
<td>0.0332</td>
<td>0.0914</td>
</tr>
<tr>
<td>$W_{13}$</td>
<td>0.1153</td>
<td>0.1379</td>
</tr>
<tr>
<td>$W_{14}$</td>
<td>0.8364</td>
<td>0.7009</td>
</tr>
<tr>
<td>$W_{15}$</td>
<td>0.0622</td>
<td>0.1530</td>
</tr>
<tr>
<td>$W_{16}$</td>
<td>0.1031</td>
<td>0.1227</td>
</tr>
</tbody>
</table>

On the other hand, the reputation of well-reputed web services is slightly higher in the association-based method. This is because the reputation of well-reputed web services increases significantly when they are associated with another well-reputed web service. For example, as can be seen in the graphs of $W_2$ and $W_5$ in Figure 5, these two web services are paired together in the 2nd composition, which increases the reputation of both these services.

We also studied the following specific cases of reputation evolution:

1. How the reputation of a well-reputed web service evolves when it is composed (or associated) with another well-reputed web service.
How the reputation of a well-reputed web service evolves when it is composed with a not-so-reputed web service.

How the reputation of a not-so-reputed web service evolves when it is composed with a well-reputed web service.

How the reputation of a not-so-reputed web service evolves when it is composed with a not-so-reputed web service.

Each of these cases are analysed from the aspect of an increase or decrease in the association coefficient between any two web services. The results of these four cases are shown in Table 4. In this table, we make the following observations:

- The reputation of a web service (well-reputed or not-so-reputed) increases when the change in the association coefficient is positive, and decreases when it is negative. For example, as shown in Table 4 (case 4), the reputation $R_{W_i}$ of web service $W_i$ increases from 0.0976 to 0.0991 when it is composed (with an associated coefficient $\lambda_{1,8} = 0.0455$) with a not-so-reputed web service $W_8$ having a reputation 0.1721. However, when the associated coefficient $\lambda_{1,8}$ between them is $-0.0817$, the reputation of web service $W_i$ decreases from 0.1426 to 0.1418 when composed with web service $W_8$ of reputation 0.1982.

- We also observe that, when a well-reputed service is composed with another well-reputed web service, its reputation increases more sharply compared to when it is composed with a not-so-reputed web service. For instance, as can be seen in Table 4, the reputation $R_{W_j}$ of a well-reputed web service $W_j$ increases sharply from 0.5600 to 0.5897 (case 1) when it is associated with another well-reputed web service $W_5$, compared to its increase from 0.8831 to 0.8865 (case 2) when it is associated with a not-so-reputed web service, $W_{16}$.

Table 4  Reputation evolution in four specific cases

<table>
<thead>
<tr>
<th>Case</th>
<th>Description</th>
<th>$R_{W_i}$</th>
<th>$\Delta \lambda_{ij}$</th>
<th>$R_{W_i}$ before composition</th>
<th>$R_{W_j}$ after composition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Well-reputed ($W_2$) with Well-reputed ($W_5$)</td>
<td>0.6100</td>
<td>0.0998</td>
<td>0.5600</td>
<td>0.5897</td>
</tr>
<tr>
<td></td>
<td>$R_{W_i} = 0.8238$</td>
<td>$\Delta \lambda_{2,5} = -0.0592$</td>
<td>$R_{W_i} = 0.8206$</td>
<td>$R_{W_j} = 0.8177$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Well-reputed ($W_2$) with Not-so-reputed ($W_{16}$)</td>
<td>0.2871</td>
<td>0.0668</td>
<td>0.8831</td>
<td>0.8865</td>
</tr>
<tr>
<td></td>
<td>$R_{W_i} = 0.2912$</td>
<td>$\Delta \lambda_{2,16} = -0.0280$</td>
<td>$R_{W_i} = 0.8831$</td>
<td>$R_{W_j} = 0.8824$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Not-so-reputed ($W_{16}$) with Well-reputed ($W_2$)</td>
<td>0.8831</td>
<td>0.0668</td>
<td>0.2871</td>
<td>0.3118</td>
</tr>
<tr>
<td></td>
<td>$R_{W_i} = 0.8831$</td>
<td>$\Delta \lambda_{16,2} = -0.0280$</td>
<td>$R_{W_i} = 0.2912$</td>
<td>$R_{W_j} = 0.2891$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Not-so-reputed ($W_1$) with Not-so-reputed ($W_8$)</td>
<td>0.1721</td>
<td>0.0455</td>
<td>0.0976</td>
<td>0.0991</td>
</tr>
<tr>
<td></td>
<td>$R_{W_i} = 0.1982$</td>
<td>$\Delta \lambda_{1,8} = -0.0817$</td>
<td>$R_{W_i} = 0.1426$</td>
<td>$R_{W_j} = 0.1418$</td>
<td></td>
</tr>
</tbody>
</table>
To observe how the reputation of a specific pair evolved over a period of our experiment, we analyse the case of a specific pair \((W_2, W_5)\). Figure 6 shows how the reputation of the web service \(W_2\) evolves due to its 13 compositions with the web service \(W_5\). Note that, both \(W_2\) and \(W_5\) are well-reputed web services. In Figure 6, the plot represented by the dotted line shows the association coefficient \(\lambda_{2,5}\) between these two web services, and the (solid) line segments show how the reputation \(R_{w_2}\) of web service \(W_2\) increases from the start point to the end point of the line segment. The square symbol denotes the reputation of the well-reputed web service \(W_5\) at its different composition instances.

**Figure 6** The reputation evolution of the well-reputed web service \(W_2\) due to its association with the other well-reputed web service \(W_5\) (see online version for colours)

It can be seen from Figure 6 that the reputation of \(W_2\) increases at the composition instances when the association coefficient grows, and it decreases otherwise. For example, the association coefficient decreases from the 5th composition to the 6th composition (labelled by the symbol \(A\)), and its effect can be seen in the reputation reduction on the line segment joining the 6th and 7th composition instances (symbol \(B\)), though this diminution is quite small.

Also note that, in Figure 6, there is a gap in the solid lines. This is because, the reputation \(R_{w_2}\) of web service \(W_2\) also evolves based on its association with the web services other than \(W_5\). For example, the reputation of \(W_2\) increases from 0.59 to 0.68 due its association with the other services before its 2nd composition with \(W_5\).
To summarise, the results suggest that the proposed association-based method provides a mechanism to determine the reputation of the web services. The association-based method not only provides reputation values comparable to what we achieve using the feedback-based method, but it also overcomes the dependency on users’ feedback. Moreover, the proposed method can also accommodate newly created web services without any prior information of their reputation.

5.3 Discussion

In this section, we discuss overhead and constraints of the proposed method. These can be summarised as follows:

- The proposed method has an overhead of maintaining the dynamics of the compositions and the association coefficient among web services, however this overhead is comparable to maintaining the feedback of the users. Moreover, the overhead in our method can be reduced by computing the association coefficient between two web services only when their composition occurs.

- As the proposed method takes into account the initial reputation of the services, their evolved reputation will be somewhat affected by the initial values. Therefore, if there is no good way of assessing the initial reputation of the services, it is recommended to assign a very low value as their initial reputation. In this case, the reputation evolution of all the services will be slower and the impact of having unfair initial reputation will be minimised.

- In our experiments, we have assumed that the data about composition is collected through a centralised unit. However, in order to apply this method in a distributed setting where the data about the composition would be collected by different entities, the issues like how to share the distributed composition data need to be resolved. This can be further explored in future.

6 Conclusions

We have presented a method for the dynamic computation of reputation of web services based on their association amongst themselves. The association between any two web services is computed based on how often they are composed together. The experimental results have shown that the proposed method overcomes the dependency of users’ feedback, and it also dynamically determines the reputation values of web services. The reputation computation results obtained using the two methods, the proposed association-based method and the traditional feedback-based method, are quite comparable. Future work would be to explore how the proposed reputation evolution method can be used in different architectural settings and applications.
References


