Integrating User Invocation Data and Extended Semantics for Service Community Discovery

QI Yu and Jai Kang
College of Computing and Information Sciences
Rochester Institute of Technology

We present in this paper an integrated service discovery framework based on Non-negative Matrix Factorization (NMF). NMF provides an effective means to cluster high-dimensional sparse data with both high clustering accuracy and good interpretability of the clustering result. This makes NMF especially suitable for service community discovery by clustering the Web service description data. Nevertheless, as the standard service description language, WSDL, rarely offers rich service description, accurate discovery of service communities remains as a central challenge. The proposed integrated service discovery framework adapts and makes key extensions to NMF. It enables the amalgamation of NMF with other key evidences that may be helpful to further boost the clustering accuracy. In particular, we identify two important sources of information, users' service invocation data and extended semantics of service descriptions, and seamless integrate them into the proposed service community discovery framework. We apply the proposed framework to real-world Web services to demonstrate the effectiveness in service community discovery.

Keywords: Web service - Service Discovery - Service Community - Matrix Factorization

1. INTRODUCTION

Service computing is increasingly gaining popularity and importance for use in both industry and academia [Yu et al. 2008]. It offers an attractive computing paradigm, via which, key functionalities can be wrapped as Web services and deployed on the Web for easy access by the computing society and the general public. Recent statistics reveal that there are over 28,000 Web services offered by almost 8,000 distinct providers located in different parts of the world [1]. These numbers still keep increasing in a fast rate. While the ever increasing number of Web services holds significant promise, we are facing a risk of overloading the users with these services. As casual users are not equipped with the required technical skills, they can be easily overwhelmed by the large scale and heterogeneous services. Without the capability to effectively and efficiently organize, retrieve, and use these services, we can quickly reach the point of negative return.

Despite the abundance of various supporting technologies to facilitate the access to Web services, a meaningful organization that can effectively categorize the large and diverse Web service space is still missing from the current service computing technology stack. Discovery of communities that group together functionality similar services is a fundamental enabler for a set of key service computing tasks, including service discovery, service composition, and quality based service selection. As an example, discovery of a service with user desired functionality can be performed solely within the service communities that offer relevant functionality. This not only improves the accuracy of discovery but also significantly reduces the searching time because services from irrelevant communities are directly filtered out. Furthermore, grouping together relevant services into communities facilitates the discovery of potentially composable services. Service composition can be (semi-)automated in such a controlled environment to generate value-added composite services. Service communities are also a key building block that allows service users to select the "best" Web services and/or their combinations with respect to their expected quality. As competing Web services that offer "similar" functionalities will be

[1] http://webservices.seekda.com/

Author's address: Rochester Institute of Technology, 1 Lomb Memorial Drive, Rochester, NY 14623; email: {qi.yu, jai.kang}@rit.edu.

categorized into the same service communities, service users are provided with a one-stop shop to get the service with required functionality and the best desired quality.

When service communities are needed, most existing approaches either assume that the communities already exist or can be constructed in a "top-down" fashion. A top-down approach typically starts with a set of predefined template services and bootstraps the communities by grouping together the related template services. It then relies on the services to register to the corresponding service communities based on the similarity with the template services. A top-down strategy may only be applicable to a limited number of Web services (e.g., within an organization) or building a community by creating all its services from scratch. It demands a centralized control on the services that are involved in the communities. Nevertheless, when a large scale of Web services from an open environment (e.g., the Web) are considered, the top-down strategy presents key challenges. On the one hand, as Web services are expected to be *autonomous* (i.e., provided by independent service providers) and *a priori unknown*, it is infeasible to predefine the template services that match the functionalities of these services. On the other hand, it is also unreasonable to rely on the independent service providers to register their services with the predefined service communities.

Some research is underway to directly infer service communities from the Web service descriptions. As the standard Web service description language, WSDL primarily describes a service from the syntactic perspective and rarely provides rich service descriptions [Dong et al. 2004]. This hinders the direct application of traditional document clustering approaches. Some recent efforts have been devoted to break the limitations of WSDL for improving the accuracy of service search and community discovery. These approaches can be divided into two categories. The first category aims to fully exploit the information carried by the WSDL service descriptions [Dong et al. 2004; Elgazzar et al. 2010; Liu et al. 2009; Liu and Wong 2008]. For example, a key premise behind the Woogle Web service search engine is that terms that co-occur frequently tend to share the same concept [Dong et al. 2004]. Nevertheless, WSDL descriptions usually come with very limited number of terms. Hence, *semantically similar terms* (e.g., car and vehicle) will have a slim chance to co-occur in a WSDL corpus and thus be deemed as irrelevant. In contrast, the second category explores external information sources, such as WordNet, Wikipedia, and search engines, to extend WSDL with rich semantics [Liu et al. 2010; Bose et al. 2008]. However, the external semantic extensions may not fit into the context of the original services. For example, "apple" means different things for a computer hardware service and an online grocery store service. In this regard, the semantic extensions are useful only when they can be leveraged in the context of the original service.

We propose an integrated framework to discover service communities from diverse and large scale Web services. In order to attack the central challenges as highlighted above, the proposed framework exploits Non-negative Matrix Factorization (NMF) as a powerful tool for service community discovery. An application of NMF to discover service communities was first presented in a shortened form as a conference paper in [Yu 2011]. In this work, we present an integrated community discovery framework, enabling the amalgamation of NMF with other key evidences that may be helpful to further boost the discovery accuracy. In particular, we identify two important sources of information, user service invocation data and extended semantics of service descriptions, and seamless integrate them into the proposed service community discovery framework. The **key contributions** of the proposed framework are summarized as follows.

— **Community Discovery via NMF.** Service community discovery is to group together Web services with similar functionalities. As the functionalities of Web services are captured by the operations they offer, we construct an $m \times n$ matrix X, where the i-th row represents service $\mathbf{s}_i$, the j-th column represents operation $\mathbf{o}_j$, and the entry $\mathbf{X}(i, j)$ represents the association between $\mathbf{s}_i$ and $\mathbf{o}_j$. We exploit an augmented version of NMF, called Non-negative Matrix Tri-Factorization (NMTF), which factorizes matrix $\mathbf{X}$ into three low-rank non-negative matrices: a service cluster indicator matrix, an operation cluster indicator matrix, and a service-operation

association matrix. NMTF in essence simultaneously clusters both services and operations. In this way, NMTF not only leverages the WSDL service descriptions but also exploits the "duality" relationship between services and operations [Dhillon 2001; Yu and Rege 2010]. Duality signifies that service clustering is determined by the functionalities of services (i.e., the operations they offer) while operation clustering is determined by the co-occurrence of operations in functionally similar services. Simultaneously clustering services and operations enables the two clustering processes to guide each other so that the overall clustering accuracy can be improved. Furthermore, the non-negative constraint of NMTF yields a natural parts-based representation of the data as it only allows addictive combinations [Lee and Seung. 1999]. Thus, the clustering result from NMTF is more intuitive to interpret.

— **Semantic Extension Integration.** NMTF goes beyond the existing service and community discovery approaches by fully exploiting the information carried by the WSDL corpus, which includes not only the service descriptions but also the duality relationship between services and operations. Unfortunately, due to the limited descriptive capacity of WSDL, terms that share similar semantics may be regarded as irrelevant if they do not co-occur in a WSDL file. This will lead to poor community discovery performance. To attack this challenge, we compute the semantic extensions of the WSDL corpus by leveraging external information sources. We then integrate the semantic extensions into the NMTF process, where the original service descriptions are used to discover the service communities. The amalgamation of the semantic extensions and NMTF has the effect of fitting the extended semantics obtained from external sources into the context of the original services. This enables the proposed the framework to effectively leverage the semantic extensions to benefit service community discovery.

— **User Invocation Data Integration.** Inspired by the recent development in crowdsourcing research [Doan et al. 2011], we believe that users play a key role during the interactions with Web services. A valuable information repository that will benefit service community discovery is the user service invocation data. A key premise behind this idea is that services that are used by the same users tend come from the same communities. The premise is supported by two key observations:

  — First, the functionalities of services used by users are highly relevant to the type of work they want to perform with the services. For example, a biologist may frequently use Web services that provide easy access to the biological data (e.g., BioMOBY [Wilkinson and Links 2002]). In contrast, a user who travels a lot may often times use the travel related services, such as airline status services, map services, and Point-Of-Interest (POI) services.

  — Second, services that are used together by the same users commonly appear as component services of a composite service or service mashup. A composite service orchestrates functionality relevant services to provide a customized and value-added service.

  Hence, user invocation data contains useful information that can lead to accurate community structures. Leveraging the invocation data is expected to improve the community discovery accuracy.

— **Iterative Algorithm and Experimental Study.** We develop an iterative algorithm that can efficiently construct the communities based on a set of multiplicative update rules. We test our algorithm on a real-world Web service description dataset to assess the effectiveness of the proposed integrated service community discovery framework.

The remainder of the paper is organized as follows. We present a generic framework based on NMTF for service community discovery in Section 2. We identify two key external information sources and present the strategy to integrate them into the proposed community discovery framework 3. We develop an efficient iterative algorithm based upon the proposed framework in Section 4. We evaluate the effectiveness of the proposed service community discovery framework via real-world Web services in Section 5. We give an overview of related work in Section 6 and conclude in Section 7.

Table I. Notations

Notation	Description
$\mathcal{S}, \mathcal{O}$	sets of services and operations
$\mathbf{s}_i, \mathbf{o}_j$	the i^{th} service and j^{th} operation
$W_{\mathbf{s}_i}$	the WSDL description of service $\mathbf{s}_i$
$E(W_{\mathbf{s}_i})$	the semantic extension of $W_{\mathbf{s}_i}$
$\hat{s}_p$, $\hat{o}_q$	the p^{th} service community and q^{th} operation community
$\mathbf{X}, \mathbf{S}, \mathbf{R}, \mathbf{O}$	matrices
$\mathbf{X}^T$	the transpose of matrix $\mathbf{X}$
$\mathbf{X}(i,j)$	the element at the i^{th} row and j^{th} column of matrix $\mathbf{X}$
$\mathbf{X}(i,:)$	the i^{th} row of matrix $\mathbf{X}$
$\mathbf{X}(:,j)$	the j^{th} column of matrix $\mathbf{X}$

2. FRAMEWORK FOR SERVICE COMMUNITY DISCOVERY

Service community discovery aims to group together Web services that provide similar functionalities. Since the functionality of a Web service is reflected by its operations, it is desirable to evaluate the similarity between services based on the operations they offer. We consider two types of objects in a Web service space: services $\mathcal{S} = \{\mathbf{s}_1, ..., \mathbf{s}_m\}$ and operations $\mathcal{O} = \{\mathbf{o}_1, ..., \mathbf{o}_n\}$. The association (or similarity) between a service s and an operation $\mathbf{o}$ is denoted by a scalar value $x(\mathbf{s}, \mathbf{o})$. Thus, we can use a m-by-n two dimensional matrix $\mathbf{X}$ to denote the association between each pair of service and operation if we map the row indices into $\mathcal{S}$ and the column indices into $\mathcal{O}$. Each entry $\mathbf{X}(i,j) \in \mathbf{X}$ denotes the association between service $\mathbf{s}_i$ and operation $\mathbf{o}_j$. We refer to the matrix $\mathbf{X}$ as the service-operation contingency matrix. Once matrix $\mathbf{X}$ is constructed, the similarity between services $\mathbf{s}_i$ and $\mathbf{s}_j$ can be computed as the dot-product of the i^{th} and j^{th} row vectors of $\mathbf{X}$:

$$sim(\mathbf{s}_i, \mathbf{s}_j) = \mathbf{X}(i,:) \cdot \mathbf{X}(j,:) \tag{1}$$

To complete the construction of matrix $\mathbf{X}$, we also need to compute the association between each pair of service and operation. This can be achieved by representing both services and operations as N-dimensional term vectors, where N is the number of distinct terms in the WSDL corpus. More specifically, if the k^{th} term appears in the description of service $\mathbf{s}_i$ (or the signature of operation $\mathbf{o}_j$), the corresponding entry in the term vector will be set as the frequency of this term [2]. Otherwise, the corresponding entry is set to 0. Hence, the association between service $\mathbf{s}_i$ and operation $\mathbf{o}_j$ can be computed as the dot-product of their term vectors. Table I lists the notations that are used throughout this paper.

2.1 Community Discovery via NMTF

In this section, we propose to use a Non-negative Matrix Tri-Factorization (NMTF) process to discovery service communities based on the service-operation contingency matrix $\mathbf{X}$ constructed above. In particular, NMTF factorizes $\mathbf{X}$ into three low-rank matrices, i.e.,

$$\mathbf{X} \approx \mathbf{SRO}^T \tag{2}$$

where $\mathbf{S} \in \mathbb{R}^{m \times k}$ is the cluster indicator matrix for clustering services (i.e., rows of $\mathbf{X}$), $\mathbf{O} \in \mathbb{R}^{n \times l}$ is the cluster indicator matrix for clustering operations (i.e., columns of $\mathbf{X}$), $\mathbf{R} \in \mathbb{R}^{k \times l}$ is the cluster association matrix that captures the association between service clusters and operation clusters. NMTF in essence simultaneously clusters $\mathcal{S}$ into k disjoint service communities and $\mathcal{O}$ into l disjoint operation communities. In this way, it effectively exploits the *duality* between services and operations to improve the overall community discovery accuracy.

To further demonstrate how NMTF works, we use a collection of real-world WSDL files obtained from [Klusch et al. 2006]. This dataset consists of over 450 services from 7 different

[2]Other values, such as the TFIDF score [Baeza-Yates and Ribeiro-Neto 1999], can also be used.

$$\begin{pmatrix} 81 & 3 & 22 & 0 & 0 \\ 3 & 68 & 30 & 0 & 4 \\ 22 & 30 & 71 & 0 & 4 \\ 0 & 0 & 0 & 42 & 22 \\ 0 & 6 & 6 & 54 & 257 \end{pmatrix}_{\mathbf{X}} \approx \begin{pmatrix} 0.3069 & 0.0000 \\ 0.2878 & 0.0042 \\ 0.3834 & 0.0017 \\ 0.0000 & 0.0824 \\ 0.0000 & 0.7045 \end{pmatrix}_{\mathbf{S}} \begin{pmatrix} 307.7633 & 8.4288 \\ 10.9841 & 612.4139 \end{pmatrix}_{\mathbf{R}} \begin{pmatrix} 0.3418 & 0.0000 \\ 0.3206 & 0.0064 \\ 0.4274 & 0.0029 \\ 0.0000 & 0.1347 \\ 0.0000 & 0.5936 \end{pmatrix}^T_{\mathbf{O}} \tag{3}$$

domains. For a clear illustration, we select 5 services, where three of them are from the eduction domain and two are from the medical domain. Each service offers one operation and thus there are altogether five operations. Through some preprocessing of the WSDL files (refer to Section 5 for details), we identify 33 distinct terms. Hence, all the services and operations can be represented as 33-dimensional vectors. Then, we construct a 5×5 contingency matrix $\mathbf{X}$ where each row represents a service and each column represents an operation. Applying NMFT on $\mathbf{X}$, we obtain the following result in Equation (3). It is easy to tell that the first three rows of $\mathbf{X}$, which represent three education services, are grouped into the first service community $\hat{s}_1$ (because $\mathbf{S}(i,1) > \mathbf{S}(i,2)$, where $i \in \{1,2,3\}$). The last two rows, representing two medical services are grouped into the second service community $\hat{s}_2$ (because $\mathbf{S}(i,1) < \mathbf{S}(i,2)$, where $i \in \{4,5\}$). Similarly, columns 1, 2, and 3, which represent three operations from the education domain are grouped into the first operation community $\hat{o}_1$ and the fourth and fifth operations are grouped into the second operation community $\hat{o}_2$.

2.2 Result Interpretation

Under NMTF, a row vector $\mathbf{X}(i,:) \in \mathbf{X}$, which corresponds to the i^{th} service in the service space, can be represented as follows:

$$\mathbf{X}(i,:) \approx \sum_{p=1}^{k} \mathbf{S}(i,p)\mathbf{V}(p,:) \tag{4}$$

where $\mathbf{V} = \mathbf{R}\mathbf{O}^T$. Each entry $\mathbf{V}(p,j)$ captures the association of operation $\mathbf{o}_j$ with service community $\hat{s}_p$. $\mathbf{V}(p,:)$, a row vector of $\mathbf{V}$, captures the association of service community $\hat{s}_p$ with all operations. In this regard, $\mathbf{V}(p,:)$ can be regarded as the centroid vector of service community $\hat{s}_p$. Recall that NMTF enforces a non-negative constraints on matrices $\mathbf{S}, \mathbf{R}, \mathbf{O}$. In addition, $\mathbf{S}$ is the cluster indicator matrix with $\mathbf{S}(i,p) \in \mathbf{S}$ representing the cluster membership of $\mathbf{s}_i$ in service community $\hat{s}_p$. Therefore, a service $\mathbf{X}(i,:)$ is essentially formulated as the *additive combination* of all the service community centroids weighted by the memberships of $\mathbf{s}_i$ in these communities.

2.3 Objective Function

NMTF aims to find three low-rank non-negative matrices to approximate the original service-operation contingency matrix $\mathbf{X}$. A good approximation requires that values in $\mathbf{S}\mathbf{R}\mathbf{O}^T$ be close to the original values in $\mathbf{X}$. Considering the non-negative constraints, it is equivalent to solve the following optimization problem:

$$\min_{\mathbf{S}\geqslant 0, \mathbf{R}\geqslant 0, \mathbf{O}\geqslant 0} J_0 = ||\mathbf{X} - \mathbf{S}\mathbf{R}\mathbf{O}^T||_F^2 \tag{5}$$

where $||\cdot||_F$ denotes Frobenius norm.

3. INTEGRATING EXTERNAL INFORMATION SOURCES

The NMTF process proposed in Section 2 aims to fully leverage the WSDL descriptions to discover service communities. Due to the autonomous nature of Web services, it is common that different WSDL files use distinct terms to describe similar functionalities (e.g., AirlineReservation and BookFlight). Existing document clustering techniques rely on the co-occurrence of terms to

identify semantically similar terms [Dong et al. 2004]. Unfortunately, most WSDL descriptions are generated from program source code written in certain programming languages. This implies that WSDL files rarely provide rich service descriptions. Due to the limited terms used in the WSDL descriptions, the semantically similar terms may have a low chance to co-occur in the WSDL corpus. To attack this challenge, we propose to explore two key evidences from external information sources: (i) user invocation data, and (ii) extended semantics of WSDL service descriptions.

3.1 User Invocation Data Integration

We present in this section how to integrate user invocation data into the proposed NMTF based service discovery framework. As motivated in Section 1, services that are accessed together by the same users tend to provide relevant functionalities. Hence, incorporating user invocation data is expected to improve the accuracy of service community discovery. Another important piece of information that can be derived from user invocation data is the irrelevance of the services' functionalities. For example, if two services have never been accessed together by any given user, it probably means that these two services are not relevant and should not be clustered into the same community. The irrelevance can also serve as a guidance on the community discovery process.

We now show how to integrate the relevant and irrelevant services information into the NMTF. As indicated in Eq (4), $\mathbf{V}(p,:)$ can be regarded as the centroid vector of service community $\hat{s}_p$. We also know that $\mathbf{S}$ is the cluster indicator matrix with $\mathbf{S}(i,p) \in \mathbf{S}$ representing the cluster membership of $\mathbf{s}_i$ in service community $\hat{s}_p$. If we enforce a hard cluster membership, we have $\mathbf{S}(i,p) = 1$ if a service, i.e., a row vector $\mathbf{X}(i,:) \in \hat{s}_p$, and $\mathbf{S}(i,p) = 0$ if otherwise. Based on this, the objective function in Eq (5) can be reformulated as:

$$\min \sum_{p} \sum_{\mathbf{X}(i,:) \in \hat{s}_p} \|\mathbf{X}(i,:) - \mathbf{V}(p,:)\|^2 \tag{6}$$

which is essentially the objective function of K-means clustering. Now consider two services $\mathbf{X}(i,:)$ and $\mathbf{X}(j,:)$, which co-occur frequently in the user invocation data. Denote $f[\mathbf{X}(i,:), \mathbf{X}(j,:)]$ as the frequency these two services co-occur and t as a predefined threshold value. This gives

$$[\mathbf{X}(i,:), \mathbf{X}(j,:)] \in \begin{cases} \mathcal{R} & \text{if } f[\mathbf{X}(i,:), \mathbf{X}(j,:)] \geqslant t \\ \mathcal{I} & \text{if } f[\mathbf{X}(i,:), \mathbf{X}(j,:)] = 0 \end{cases} \tag{7}$$

where $\mathcal{R}$ and $\mathcal{I}$ are sets of relevant and irrelevant services, respectively. We incorporate the relevant and irrelevant service information derived from the user invocation data as:

$$\min \left(\sum_{p} \sum_{\mathbf{X}(i,:) \in \hat{s}_p} \|\mathbf{X}(i,:) - \mathbf{V}(p,:)\|^2 \right.$$
$$\left. - \sum_{p} \sum_{\substack{[\mathbf{X}(i,:), \mathbf{X}(j,:)] \in \mathcal{R} \\ \mathbf{S}(i,p) = \mathbf{S}(j,p)}} \phi_{ij} + \sum_{p} \sum_{\substack{[\mathbf{X}(i,:), \mathbf{X}(j,:)] \in \mathcal{I} \\ \mathbf{S}(i,p) = \mathbf{S}(j,p)}} \phi_{ij} \right) \tag{8}$$

The basic rationale of the above objective function can be illustrated as follows. When two relevant services $\mathbf{X}(i,:)$ and $\mathbf{X}(j,:)$ are assigned to the same community, we deduct ϕ_{ij} from the objective function to award the consistent result [Kulis et al. 2005]. On the other hand, when two irrelevant services are clustered into the same community, we increase the objective function by ϕ_{ij} to panelize the inconsistent result. Following [Wang et al. 2008], we define

$$\mathbf{\Phi}(i,j) = \begin{cases} -\phi_{ij}, & \text{if } [\mathbf{X}(i,:), \mathbf{X}(j,:)] \in \mathcal{R} \\ \phi_{ij}, & \text{if } [\mathbf{X}(i,:), \mathbf{X}(j,:)] \in \mathcal{I} \\ 0, & \text{if otherwise} \end{cases} \tag{9}$$

Recall that when a hard cluster membership is enforced, $\mathbf{S}(i,p)$ takes a binary value (i.e., 1 or 0). This transforms Eq (8) into:

$$\min \left(\sum_{p} \sum_{i} \mathbf{S}(i,p) \left\| \mathbf{X}(i,:) - \mathbf{V}(p,:) \right\|^2 + \sum_{p} \sum_{i,j} \mathbf{S}(i,p)\mathbf{S}(j,p)\mathbf{\Phi}(i,j) \right) \tag{10}$$

Formulating Eq (10) into a matrix format gives

$$\min_{\mathbf{S}\geqslant 0, \mathbf{R}\geqslant 0, \mathbf{O}\geqslant 0} J_{User} = ||\mathbf{X} - \mathbf{S}\mathbf{R}\mathbf{O}^T||_F^2 + \lambda_1 \mathrm{Tr}(\mathbf{S}^T\mathbf{\Phi}\mathbf{S}) \tag{11}$$

where λ_1 is a regularization parameter. As enforcing binary values on $\mathbf{S}$ makes the optimization problem in Eq (11) unsolvable [Wang et al. 2008], we relax the constraint by only enforcing an non-negative constraint on $\mathbf{S}$.

3.2 Extended Semantics Integration

In this section, we present how to explore external information sources to extend WSDL descriptions with rich semantics. We then exploit these extended semantics to improve the accuracy of service community discovery. Some recent efforts have been devoted to leverage semantic extensions of the WSDL files to improve service discovery [Liu et al. 2010; Bose et al. 2008]. In these approaches, the semantic extensions are directly used to match users' queries or compute the semantic distances between terms. However, as motivated in Section 1, using external sources may lead to semantic extensions that are irrelevant to the original services. Using irrelevant semantics to match users' queries or compute the similarity between terms will negatively affect the service discovery accuracy.

We propose to integrate the semantic extensions of the WSDL corpus into the NMTF process, in which the original services are clustered to discover the service communities. The amalgamation of the semantic extensions and NMTF places the extended semantics into the context of the original services to improve community discovery accuracy.

3.2.1 *Computing the Semantic Extensions of the WSDL Corpus.* A number of external information sources, such as WordNet and Wikipedia, may be used to compute the semantic extensions of the WSDL corpus. However, as most WSDL descriptions originate from program source code, a lot of terms may not be proper English words. For example, the concatenation of a number of words is typically used to describe the names of operations (e.g., GeocodeByZip). Abbreviations are also commonly used in the parameters of the operations (e.g., temp for temperature). This significantly limits the effectiveness of traditional lexical references, such as WordNet, which do not include WSDL terms that are not proper English words.

One useful and powerful information source that we plan to leverage is the large volume of documents on the Web. This also allows us to exploit web search engines to effectively process the irregular and misspelled terms, which are quite common in WSDL files. We follow a procedure, which is similar to the one proposed in [Sahami and Heilman 2006] to compute the semantic extensions of the WSDL corpus:

(1) Preprocess each WSDL file ($W_{\mathbf{s}_i}$) in the corpus to identify the *functional* terms (refer to Section 5 for the details of WSDL file preprocessing). A functional term describes the functionality provided by a service.

(2) Submit each functional term $t \in W_{\mathbf{s}_i}$ to a search engine and retrieve the top-k documents, $d_1, ..., d_k$.

(3) Rank the terms in documents, $d_1, ..., d_k$ based on their TFIDF scores and select the top-r terms.

(4) The semantic extension of $W_{\mathbf{s}_i}$ is a vector $E(W_{\mathbf{s}_i})$, which consists of the TFIDF scores of the selected top-r terms.

3.2.2 *Extended Semantics Integration.* We propose a *graph based approach* to achieve semantic extension integration. The first step is to construct a semantic similarity graph, $G = (V, E)$, which captures the semantic similarity between different services. Each vertex v_i represents the semantic extension of a service $\mathbf{s}_i$. Two vertices are connected if the similarity $\mathbf{W}(i, j)$ between services $\mathbf{s}_i$ and $\mathbf{s}_j$ is larger than a certain threshold. The edge is weighted by $\mathbf{W}(i, j)$, which is obtained via the dot-product between $E(W_{\mathbf{s}_i})$ and $E(W_{\mathbf{s}_j})$. Based on the semantic similarity graph, the underlying rationale of semantic extension integration can be specified as follows.

Rationale: If two services $\mathbf{s}_i$ and $\mathbf{s}_j$ share similar semantic descriptions (i.e., they have a large edge weight $\mathbf{W}(i, j)$ in the similarity graph), they are expected to provide similar functionalities. Hence, their corresponding cluster memberships (e.g., $\mathbf{S}(i, p)$ and $\mathbf{S}(j, p)$) are expected to be similar. ■

Therefore, $\mathbf{W}(i, j)(\mathbf{S}(i, p) - \mathbf{S}(j, p))^2$ is expected to be small for all i, j. This is equivalent to say that

$$\mathcal{R}_p = \frac{1}{2} \sum_{i,j=1}^{m} \mathbf{W}(i, j)(\mathbf{S}(i, p) - \mathbf{S}(j, p))^2$$

is small. If all k service communities are considered and through some algebra, we have

$$\mathcal{R} = \sum_{p=1}^{k} \mathcal{R}_p = \mathrm{Tr}(\mathbf{S}^T \mathbf{L} \mathbf{S}) \tag{12}$$

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{13}$$

$$\mathbf{D}(i, i) = \sum_{j} \mathbf{W}(i, j) \tag{14}$$

where $\mathbf{L}$ is the graph Laplacian of the semantic similarity graph and $\mathbf{D}$ is the degree matrix.

To integrate the semantic extensions with the NMTF process, we incorporate $\mathcal{R}$ as a regularizer into the original objective function specified in Equation (5). Thus, service community discovery with semantic extensions can be formulated as the following optimization problem:

$$\min_{\mathbf{S} \geqslant 0, \mathbf{R} \geqslant 0, \mathbf{O} \geqslant 0} J_{Semantic} = ||\mathbf{X} - \mathbf{S}\mathbf{R}\mathbf{O}^T||_F^2 + \lambda_2 \mathrm{Tr}(\mathbf{S}^T \mathbf{L} \mathbf{S}) \tag{15}$$

where λ_2 is the regularization parameter.

3.3 The Overall Objective Function

The above two sections describe how to incorporate user invocation data and extended semantics of the service descriptions into the proposed NTMF based community discovery framework. We now integrate everything together into a single objective function, which gives:

$$\min_{\mathbf{S} \geqslant 0, \mathbf{R} \geqslant 0, \mathbf{O} \geqslant 0} J_{Overall} = ||\mathbf{X} - \mathbf{S}\mathbf{R}\mathbf{O}^T||_F^2 + \lambda_1 \mathrm{Tr}(\mathbf{S}^T \mathbf{\Phi} \mathbf{S}) + \lambda_2 \mathrm{Tr}(\mathbf{S}^T \mathbf{L} \mathbf{S}) \tag{16}$$

More generally, we can define an objective function that allows to integrate information from other external information sources:

$$\min_{\mathbf{S} \geqslant 0, \mathbf{R} \geqslant 0, \mathbf{O} \geqslant 0} J = ||\mathbf{X} - \mathbf{S}\mathbf{R}\mathbf{O}^T||_F^2 + \sum_{r} \lambda_r \mathrm{Tr}(\mathbf{S}^T \mathbf{E}_r \mathbf{S}) \tag{17}$$

where $\mathbf{E}_r$ is the encoding matrix of the information from the r-th external information source and λ_r is the corresponding regularization parameter. It is straightforward to see that Eq (16) is a special case of Eq (17), where user invocation data and extended semantics of the service descriptions are considered, and we have $\mathbf{E}_1 = \mathbf{\Phi}$ and $\mathbf{E}_2 = \mathbf{L}$.

4. ALGORITHM DERIVATION

Since the objective function is not convex in $\mathbf{S}, \mathbf{R}$ and $\mathbf{O}$ together, we adopt an iterative algorithm to find a local minimum of the optimization problem in Equation (17). In each iteration, two matrices in $\mathbf{S}, \mathbf{R}$ and $\mathbf{O}$ are fixed and the third one is updated using the update rules introduced in what follows.

We introduce three Lagrange multipliers $\mathbf{\Theta}_S \in \mathbb{R}^{m\times k}, \mathbf{\Theta}_R \in \mathbb{R}^{k\times l}$, and $\mathbf{\Theta}_O \in \mathbb{R}^{n\times l}$ to handle the non-negative constraints of $\mathbf{S}, \mathbf{R}$ and $\mathbf{O}$. Thus, the Lagrangian function is

$$\begin{aligned}\mathcal{L} &= ||\mathbf{X} - \mathbf{SRO}^T||_F^2 + \sum_r \lambda_r \mathrm{Tr}(\mathbf{S}^T\mathbf{E}_r\mathbf{S}) + \mathrm{Tr}(\mathbf{\Theta}_S\mathbf{S}^T) \\ &+ \mathrm{Tr}(\mathbf{\Theta}_R\mathbf{R}^T) + \mathrm{Tr}(\mathbf{\Theta}_O\mathbf{O}^T) \\ &= \mathrm{Tr}(\mathbf{X}^T\mathbf{X} - 2\mathbf{S}^T\mathbf{XOR}^T + \mathbf{S}^T\mathbf{SRO}^T\mathbf{OR}^T) \\ &+ \sum_r \lambda_r \mathrm{Tr}(\mathbf{S}^T\mathbf{E}_r\mathbf{S}) + \mathrm{Tr}(\mathbf{\Theta}_S\mathbf{S}^T) + \mathrm{Tr}(\mathbf{\Theta}_R\mathbf{R}^T) + \mathrm{Tr}(\mathbf{\Theta}_O\mathbf{O}^T) \end{aligned} \tag{18}$$

The partial derivative of $\mathcal{L}$ with respect to $\mathbf{S}, \mathbf{R}$ and $\mathbf{O}$ are:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{S}} = -2\mathbf{XOR}^T + 2\mathbf{SRO}^T\mathbf{OR}^T + 2\sum_r \lambda_r \mathbf{E}_r\mathbf{S} + \mathbf{\Theta}_S \tag{19}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{O}} = -2\mathbf{X}^T\mathbf{SR} + 2\mathbf{OR}^T\mathbf{S}^T\mathbf{SR} + \mathbf{\Theta}_O \tag{20}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{R}} = -2\mathbf{S}^T\mathbf{XO} + 2\mathbf{S}^T\mathbf{SRO}^T\mathbf{O} + \mathbf{\Theta}_R \tag{21}$$

Using the KKT complementarity conditions for the non-negativity $\mathbf{\Theta}_S(i,p)\mathbf{S}(i,p) = 0$, $\mathbf{\Theta}_O(j,q)\mathbf{O}(j,q) = 0$, and $\mathbf{\Theta}_R(p,q)\mathbf{R}(p,q) = 0$, we have

$$\left(-2\mathbf{XOR}^T + 2\mathbf{SRO}^T\mathbf{OR}^T + 2\sum_r \lambda_r \mathbf{E}_r\mathbf{S}\right)(i,p)\mathbf{S}(i,p) = 0 \tag{22}$$

$$\left(-2\mathbf{X}^T\mathbf{SR} + 2\mathbf{OR}^T\mathbf{S}^T\mathbf{SR}\right)(j,q)\mathbf{O}(j,q) = 0 \tag{23}$$

$$\left(2\mathbf{S}^T\mathbf{XO} + 2\mathbf{S}^T\mathbf{SRO}^T\mathbf{O}\right)(p,q)\mathbf{R}(p,q) = 0 \tag{24}$$

Since $\mathbf{E}_r$ may contain both positive and negative terms, we define $\mathbf{E}_r^+ = \frac{|\mathbf{E}_r|+\mathbf{E}_r}{2}$ and $\mathbf{E}_r^- = \frac{|\mathbf{E}_r|-\mathbf{E}_r}{2}$. This gives $\mathbf{E}_r = \mathbf{E}_r^+ - \mathbf{E}_r^-$. These equations lead to the following update rules:

$$\mathbf{S}(i,p) \leftarrow \mathbf{S}(i,p)\sqrt{\frac{\left(\mathbf{SRO}^T + \sum_r \lambda_r \mathbf{E}_r^+\mathbf{S}\right)(i,p)}{\left(\sum_r \lambda_r \mathbf{E}_r^-\mathbf{S} + \mathbf{SRO}^T\mathbf{OR}^T\right)(i,p)}} \tag{25}$$

$$\mathbf{O}(j,q) \leftarrow \mathbf{O}(j,q)\sqrt{\frac{(\mathbf{X}^T\mathbf{SR})(j,q)}{\left(\mathbf{OR}^T\mathbf{S}^T\mathbf{SR}\right)(j,q)}} \tag{26}$$

$$\mathbf{R}(p,q) \leftarrow \mathbf{R}(p,q)\sqrt{\frac{(\mathbf{S}^T\mathbf{XO})(p,q)}{(\mathbf{S}^T\mathbf{SRO}^T\mathbf{O})(p,q)}} \tag{27}$$

The algorithm iteratively applies these three multiplicative rules to update the matrices until the algorithm converges or a specified maximum iteration number is reached. The details are summarized in Algorithm 1.

Algorithm 1 The Iterative Algorithm

Require: $\mathbf{X}$, $\mathbf{E}_r$. number of service clusters k, number of operation clusters l, and regularization parameters λ_r

Ensure: Service cluster indicator matrix $\mathbf{S}$

1: Initialize matrices $\mathbf{S}, \mathbf{R}, \mathbf{O}$.
2: **while** algorithm not converge **and** iteration $\leqslant$ max_iteration **do**
3: $\mathbf{S}(i,p) \leftarrow \mathbf{S}(i,p)\sqrt{\frac{\left(\mathbf{SRO}^T+\sum_r \lambda_r \mathbf{E}_r^+\mathbf{S}\right)(i,p)}{\left(\sum_r \lambda_r \mathbf{E}_r^-\mathbf{S}+\mathbf{SRO}^T\mathbf{OR}^T\right)(i,p)}}$
4: $\mathbf{O}(j,q) \leftarrow \mathbf{O}(j,q)\sqrt{\frac{(\mathbf{X}^T\mathbf{SR})(j,q)}{(\mathbf{OR}^T\mathbf{S}^T\mathbf{SR})(j,q)}}$
5: $\mathbf{R}(p,q) \leftarrow \mathbf{R}(p,q)\sqrt{\frac{(\mathbf{S}^T\mathbf{XO})(p,q)}{(\mathbf{S}^T\mathbf{SRO}^T\mathbf{O})(p,q)}}$
6: **end while**

5. EMPIRICAL STUDY

We conduct a set of experiments to assess the effectiveness of the proposed service community discovery framework. The experiments are performed based upon a real-world WSDL corpus obtained from [Klusch et al. 2006]. The WSDL corpus consists of over 450 services from 7 different application domains. Table II lists the number of services from each domain.

5.1 Data Preprocessing

We preprocess the WSDL corpus before applying the proposed service community discovery algorithm. The purpose of WSDL preprocessing aims to identify the *functional terms*, which describe the functionalities of the services. We follow a procedure which is similar to the one adopted in [Yu and Rege 2010]. More specifically, preprocessing consists of four steps: *extraction, tokenization, stopword removal*, and *stemming*: (1) Extraction extracts the key components of a WSDL file including types, messages, operations, port types, binding, and port using path expressions. (2) Tokenization is to decompose the concatenated terms into simple terms (e.g., from AirlineReservation to Airline and Reservation). (3) Stopword removal removes the non-functional terms, which include not only the regular stopwords but also the WSDL specific stopwords, such as *url, host, http, ftp, soap, binding, type, get, set, request, response*, etc. (4) Stemming reduces different forms of a term into a common root form. After the functional terms are identified through preprocessing, we follow the procedure described in Section 2 to construct the service-operation contingency matrix $\mathbf{X}$.

As there is currently no sizable real-world user invocation data available, we exploit the following process to create the encoding matrix. We start with a matrix $\mathbf{\Phi} \in \mathbb{R}^{m\times m}$ with all zero entries. We then randomly pick two services, $\mathbf{s}_a$ and $\mathbf{s}_b$. If these two services are from the same service domain, we set $\mathbf{\Phi}(a,b) = \mathbf{\Phi}(b,a) = -1$. On the other hand, if the two services are from different domains, we set $\mathbf{\Phi}(a,b) = \mathbf{\Phi}(b,a) = 1$. We use a parameter α to control the total pairs of relevant and irrelevant services that can be derived from the user invocation data. More specifically, we define

$$\alpha = \frac{|\mathcal{R}| + |\mathcal{I}|}{m^2/2}$$

which represents the percentage of the total number of relevant and irrelevant pairs of services over the total number of pair-wise relationships among all services.

5.2 Evaluation Metrics

The performance is assessed by comparing the community membership assigned by the proposed community discovery framework and the service domains provided by the WSDL corpus. The

Table II. Domains of Web Services

Domain	#Service	Abbreviation
Communication	42	Comm
Education	139	Educ
Economy	83	Econ
Food	23	Food
Medical	45	Medi
Travel	90	Trav
Weapon	30	Weap

accuracy metric that is used to evaluate the performance of community discovery is defined as follows.

AC **metric:** For a given service $\mathbf{s}_i$, assume that its assigned community membership is z_i and its domain label is y_i based on the WSDL corpus. The *AC* metric is defined as follows:

$$AC = \frac{\sum_{i=1}^{m} \delta(z_i, map(y_i))}{m} \tag{28}$$

where m is the total number of Web services in the WSDL corpus. $\delta(x, y)$ is the delta function that equals to one if $x = y$ and equals to zero if otherwise. $map(y_i)$ is the permutation mapping function that maps each assigned community membership to the equivalent domain label from the WSDL corpus. The Kuhn-Munkres algorithm is used to find the best mapping [Lovasz 1986].

5.3 Experiment Design and Parameter Setting

We also implement two well-know clustering algorithms to compare with the proposed service community discovery framework. These algorithms are Singular Value Decomposition (SVD) based Co-clustering algorithm and k-means algorithm. The SVD based co-clustering algorithm leverages the duality between services and operations and has been demonstrated to be effective in clustering WSDL service descriptions [Yu and Rege 2010]. We apply this algorithm to the service-operation contingency matrix to generate service communities. The k-means algorithm is applied to the semantic extensions of the WSDL corpus. The semantic extension of a WSDL file W_{s_i} is represented as a vector $E(W_{s_i})$, which consists of the TFIDF scores of the top-r terms returned by a web search engine. Refer to Section 3.2.2 for details about how to compute the semantic extension of a WSDL file. In addition, we also solely apply NMTF to the service-operation contingency matrix to generate service communities.

We plan to achieve the following objectives through the comparisons with the approaches described above:

— The comparison with the SVD based co-clustering algorithm aims to justify the effectiveness of integrating external semantic information into the service community discovery process.

— The comparison with k-means clustering on the semantic extensions of the WSDL corpus aims to demonstrate that placing the extended semantics into the context of the original service can better leverage the semantics to benefit service community discovery.

The notations and descriptions of all algorithms under comparison are given in Table III. The regularization factor λ is set to 10. We perform k-means clustering to initialize matrices $\mathbf{S}$ and $\mathbf{O}$. $\mathbf{R}$ is initialized as $\mathbf{S}^T\mathbf{X}\mathbf{O}$ [Ding et al. 2006]. We run each algorithm 50 times and the average *AC* is reported.

5.4 Performance Comparison

Table III compares the *AC* performance of six different algorithms. In this set of experiments, we set $\alpha = 0.5\%$, $\lambda_1 = 50$, and $\lambda_2 = 10$. NMTFUS generates the best result over all the

Table III. Performance Comparison

Notation	Description	Parameter values	AC (%)
NMTFUS	NMTF + User + Semantics	$\lambda_1 = 50, \lambda_2 = 10$	55.5
NMTFU	NMTF + User	$\lambda_1 = 50, \lambda_2 = 0$	55.2
NMTFS	NMTF + Semantics	$\lambda_1 = 0, \lambda_2 = 10$	54.9
NMTF	NMTF	$\lambda_1 = 0, \lambda_2 = 0$	52.5
SVCCo	SVD Co-clustering	-	45.5
SK-means	Semantic k-means	-	45.0

algorithms. The accuracy of NMTFU is only slightly worse than that of NMTFUS. NMTFS is the third best algorithm. The performance advantage of the first three algorithms over NMTF demonstrates that the integration of key external evidences indeed help improve the community discovery accuracy. Furthermore, NMTF outperforms the other two algorithms, which justifies the choice of NMTF as the baseline algorithm in the proposed service discovery framework. It is also worth to note that semantic k-means reports the lowest accuracy. This also confirms that using semantic extensions without considering the context of the original services does not necessarily benefit community discovery.

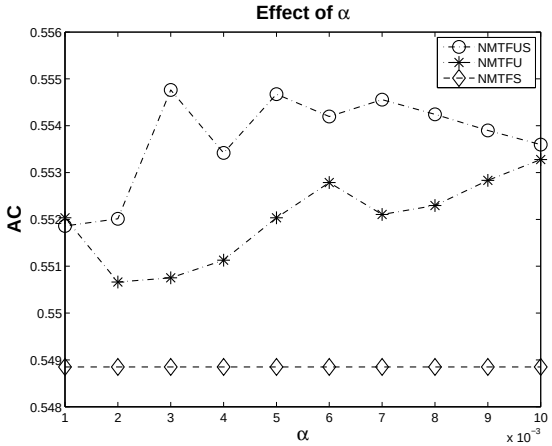


Figure. 1. Effect of User Invocation Data

Figure 1 investigates the effect of user invocation data by varying parameter α from 0.1% to 1%. First, NMTFU outperforms NMTFS, which demonstrates that user invocation data is more effective than the extended semantics in improving the accuracy of community discovery. The integration of both user invocation data and extended semantics further boots the discovery accuracy. When only user invocation data is used, the accuracy generally increases as α increases. There are some slight fluctuations, which are introduced by the random selection of the relevant and irrelevant service pairs and the use of k-means to initialize NMTF. When user invocation data and extended semantics are integrated together, the fluctuations become more obvious. This is introduced by the inconsistent impacts brought by these two different external information sources.

To further illustrate the performance difference, Figure 2 shows the confusion matrices with the best AC performances from the four different algorithms. NMTFUS, NMTFU, and NMTFS all achieve a best AC at 64.4%. Figure 2 (a) shows the corresponding confusion matrix. The best AC achieved by NMTF, SVD Co-clustering and semantic k-means are 62.8%, 47.6%, and 52.9%, respectively. Figure 2 (b), (c), and (d) show the corresponding confusion matrices from these three algorithms, respectively. Among the four algorithms, NMTF+Semantics correctly clusters

	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Comm	41	0	1	0	0	0	0
Econ	1	79	1	0	0	2	0
Educ	0	5	120	2	1	11	0
Food	1	0	19	0	0	3	0
Medi	0	0	16	6	8	10	5
Trav	0	0	47	0	0	43	0
Weap	0	0	30	0	0	0	0

(a)

	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Comm	41	0	0	0	0	1	0
Econ	1	78	1	0	0	4	0
Educ	0	6	83	0	37	12	1
Food	1	10	0	0	0	12	0
Medi	0	0	5	10	16	9	5
Trav	0	0	24	0	0	66	0
Weap	0	0	29	0	0	1	0

(b)

	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Comm	0	31	1	0	0	10	0
Econ	0	59	2	0	0	22	0
Educ	0	2	83	0	6	12	36
Food	0	0	9	0	0	13	1
Medi	0	1	13	0	4	19	8
Trav	13	3	20	5	3	41	5
Weap	0	0	2	0	0	0	28

(c)

	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Comm	30	0	12	0	0	0	0
Econ	0	57	24	0	0	2	2
Educ	0	1	119	0	17	0	2
Food	1	0	22	0	0	0	0
Medi	0	0	26	6	13	0	6
Trav	0	0	60	9	0	20	1
Weap	0	0	30	0	0	0	0

(d)

Figure. 2. Confusion Matrices with the best AC performances. (a) NMTFUS: $AC = 64.4\%$; (b) NMTF: $AC = 62.8\%$; (c) SVD Co-clustering: $AC = 47.6\%$; (d) Semantic k-means: $AC = 52.9\%$. Comm, Econ, Educ, Food, Medi, Trav, and Weap are the seven domains obtained from the WSDL corpus. C_1 to C_7 are the service communities discovered from the WSDL corpus.

the most number of services from three domains: Comm, Econ, and Educ. NMTF correctly clusters the most number of services from two domains: Medi and Trav. SVD Co-clustering correctly clusters the most number of services from the Weap domain.

One interesting observation from the confusion matrices is that none of the Food services has been correctly clustered by any of these algorithms. Most Food services are clustered as either Educ or Trav services. This may be because that the descriptions of the Food services share many common terms with Educ or Trav services. Another possible reason is due to the inappropriate definitions of the domains in the given WSDL corpus. For example, food and travel are two

highly related domains and it may be hard to set a clear boundary to differentiate services that belong to these domains. In this regard, the community discovery result can provide guidance to improve the service domain definitions.

6. RELATED WORK

We give an overview of existing works that are most relevant to the proposed approach in this section.

6.1 Service Community Discovery

A WSDL clustering technique is proposed in [Elgazzar et al. 2010] to bootstrap the discovery of Web services. Five key features are extracted from WSDL descriptions to group Web services into functionality-based clusters. These features include content, types, messages, ports, and name of the Web service. Each feature is assigned an equal weight when computing the similarity between two services. Then, the Quality Threshold (QT) clustering algorithm is applied to cluster Web services. QT is a partitional clustering algorithm, like k-means, but does not require specifying the number of clusters. A similar service clustering algorithm is proposed by using four types of features to determine the similarity between services, including content, context, service host, and service name [Liu and Wong 2008]. A weighting mechanism is used to combine these features to compute the relatedness measure between services. A service-operation co-clustering strategy is proposed in [Yu and Rege 2010] to discovery homogeneous service communities from a heterogenous service space. A SVD based algorithm is adopted to achieve the co-clustering of services and operations. Experimental result on a set of real-world Web services shows that co-clustering generates communities with better quality than just applying one-side clustering (e.g., k-means or QT) on services. The proposed service community discovery framework adopts a NMTF process that also clusters services and operations simultaneously. More importantly, the proposed framework enables the integration of key external evidences that are helpful to further boost the community discovery accuracy.

6.2 Service Search and Discovery

Woogle, a Web service search engine, is developed in [Dong et al. 2004] that helps service users discover their desired service operations and operations that may be composed with other operations. Woogle exploits a clustering algorithm and association rule mining to group parameters of service operations into concept groups. The concept groups will then be used to facilitate the matching between users' queries and the service operations. Woogle aims to combine multiple sources of evidence, including description of services, description of operations, and input/output of operations, to measure similarity. A similar approach is developed in [Liu et al. 2009] for service discovery. A service aggregation graph is also proposed to facilitate service composition. A service discovery approach is proposed in [Ma et al. 2008] based on Probabilistic Latent Semantic Analysis (PLSA). This approach treats service descriptions as regular documents without considering the limited information available in these descriptions. A common issue with the above approaches is that they solely rely on the information carried by the WSDL service descriptions. The limited descriptive capacity of the WSDL files may limit the effectiveness of these approaches. Some recent efforts have investigated to exploit semantic extensions of the WSDL files to improve service discovery [Liu et al. 2010; Bose et al. 2008]. The semantic extensions are directly used to match users' queries or compute the semantic distance between terms. However, using external resources may lead to semantic extensions that are irrelevant to the original services, which may negatively affect the service discovery accuracy. This has also been justified through our experiment results.

6.3 Service Selection and Recommendation

Service selection aims to find a proper service provider with the best user desired Quality of Service, or QoS (e.g., latency, fee, and reputation) [Yu and Bouguettaya 2008; Yu et al. 2007;

Zeng et al. 2004]. The selection is conducted within a set of services that compete to offer similar functionalities. Most existing service selection approaches assume that services with similar functionalities have already been discovered. In this regard, the proposed service community discovery framework can be used to preprocess the Web service space before service selection can be performed. Collaborative filtering based techniques have been recently adopted to provide personalized service recommendation to users [Shao et al. 2007; Zheng et al. 2009; Zhang et al. 2011; Jiang et al. 2011]. The primary focus is to accurately predict the QoS of unknown Web services based on the historical QoS data obtained from the user-service interactions. Our user invocation data integration also exploits the historical user-service interaction data. However, it focuses on improving the overall accuracy of service community discovery. Furthermore, the proposed framework allows the integration of other external information, such as the extended semantics of the service description.

7. CONCLUSION AND FUTURE DIRECTIONS

We present a generic framework for service community discovery. The proposed framework enables the integration of Non-negative Matrix Tri-Factorization (NMTF) with other key external information. NMTF in essence clusters services and operations simultaneously. In this way, it not only exploits the service descriptions but also leverages the duality relationship between services and operations to improve the performance of service community discovery. We identify two important external information sources: user invocation data and extended semantics of service descriptions. The amalgamation of NMTF and these key information helps improve the overall accuracy of community discovery. We evaluate the proposed framework on a real-world WSDL corpus and the effectiveness has been clearly justified via the comparison with three other algorithms.

REFERENCES

Baeza-Yates, R. A. and Ribeiro-Neto, B. 1999. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Bose, A., Nayak, R., and Bruza, P. 2008. Improving web service discovery by using semantic models. In *Proceedings of the 9th international conference on Web Information Systems Engineering*. 366–380.

Cai, D., He, X., and Han, J. 2005. Document clustering using locality preserving indexing. *IEEE Trans. Knowl. Data Eng. 17,* 12, 1624–1637.

Dhillon, I. S. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, New York, NY, USA, 269–274.

Ding, C. H. Q., Li, T., Peng, W., and Park, H. 2006. Orthogonal nonnegative matrix t-factorizations for clustering. In *KDD*. 126–135.

Doan, A., Ramakrishnan, R., and Halevy, A. Y. 2011. Crowdsourcing systems on the world-wide web. *Commun. ACM 54*, 86–96.

Dong, X., Halevy, A., Madhavan, J., Nemes, E., and Zhang, J. 2004. Similarity search for web services. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*. VLDB Endowment, 372–383.

Elgazzar, K., Hassan, A. E., and Martin, P. 2010. Clustering wsdl documents to bootstrap the discovery of web services. In *ICWS*. 147–154.

Jiang, Y., Liu, J., Tang, M., and Liu, X. F. 2011. An effective web service recommendation method based on personalized collaborative filtering. In *ICWS*. 211–218.

Klusch, M., Fries, B., and Sycara, K. 2006. Automated semantic web service discovery with owls-mx. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. AAMAS '06. ACM, New York, NY, USA, 915–922.

Kulis, B., Basu, S., Dhillon, I., and Mooney, R. 2005. Semi-supervised graph clustering: a kernel approach. In *Proceedings of the 22nd international conference on Machine learning*. ICML '05. ACM, New York, NY, USA, 457–464.

Lee, D. D. and Seung., H. S. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature 401*, 788–791.

Liu, F., Shi, Y., Yu, J., Wang, T., and Wu, J. 2010. Measuring similarity of web services based on wsdl. In *ICWS*. 155–162.

Liu, W. and Wong, W. 2008. Discovering homogenous service communities through web service clustering. In *SOCASE*. 69–82.

Liu, X., Huang, G., and Mei, H. 2009. Discovering homogeneous web service community in the user-centric web environment. *IEEE T. Services Computing 2,* 2, 167–181.

Lovasz, L. 1986. *Matching Theory (North-Holland mathematics studies)*. Elsevier Science Ltd.

Ma, J., Zhang, Y., and He, J. 2008. Efficiently finding web services using a clustering semantic approach. In *CSSSIA '08: Proceedings of the 2008 international workshop on Context enabled source and service selection, integration and adaptation*. ACM, New York, NY, USA, 1–8.

Sahami, M. and Heilman, T. D. 2006. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*. WWW '06. ACM, New York, NY, USA, 377–386.

Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., and Mei, H. 2007. Personalized qos prediction forweb services via collaborative filtering. In *ICWS*. 439–446.

Wang, F., Li, T., and Zhang, C. 2008. Semi-supervised clustering via matrix factorization. In *SDM*. 1–12.

Wilkinson, M. D. and Links, M. 2002. Biomoby: An open source biological web services proposal. *Briefings in Bioinformatics*, 331–341.

Xu, W., Liu, X., and Gong, Y. 2003. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. SIGIR '03. ACM, New York, NY, USA, 267–273.

Yu, Q. 2011. Place semantics into context: Service community discovery from the wsdl corpus. In *ICSOC11: The Ninth International Conference on Service Oriented Computing*.

Yu, Q. and Bouguettaya, A. 2008. Framework for web service query algebra and optimization. *TWEB 2,* 1.

Yu, Q., Liu, X., Bouguettaya, A., and Medjahed, B. 2008. Deploying and managing web services: issues, solutions, and directions. *VLDB J. 17,* 3, 537–572.

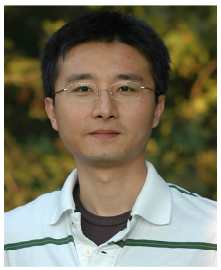
Yu, Q. and Rege, M. 2010. On service community learning: A co-clustering approach. In *ICWS*. 283–290.

Yu, T., Zhang, Y., and Lin, K.-J. 2007. Efficient algorithms for web services selection with end-to-end qos constraints. *ACM Trans. Web 1,* 1, 6.

Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., and Chang, H. 2004. Qos-aware middleware for web services composition. *IEEE Trans. Softw. Eng. 30,* 5, 311–327.

Zhang, Q., Ding, C., and Chi, C.-H. 2011. Collaborative filtering based service ranking using invocation histories. In *ICWS*. 195–202.

Zheng, Z., Ma, H., Lyu, M. R., and King, I. 2009. Wsrec: A collaborative filtering based web service recommender system. In *ICWS*. 437–444.



Qi Yu received the PhD degree in computer science from Virginia Polytechnic Institute and State University (Virginia Tech). He is an assistant professor at the college of computing and information sciences of Rochester Institute of Technology. His current research interests lie in the areas of service computing, databases, and data mining. His publications have mainly appeared in well-known journals (e.g. the VLDB journal, ACM TWEB, WWW Journal, and IEEE TKDE, and IEEE TSC) and conference proceedings (e.g., ICSOC and ICWS). He is a guest editor of the IEEE Transactions on Services Computing special issue on service query models and efficient selection. He frequently serves as a program committee member on service computing and database conferences (e.g, IEEE Cloud, SOCA, WISE, CollaborateCom, IRI, ICSOC, and APSCC). He is also a reviewer for various journals (e.g., the VLDB Journal, ACM TWEB, and IEEE TSC). He is a member of the IEEE.

Jai Kang received the PhD degree in operations research from the State University of New York at Buffalo. He is an associate professor at the college of computing and information sciences of Rochester Institute of Technology. He is a certified computing professional (CCP) specializing in systems development, IT management and core IT skills from ICCP. He has over 15 years of computing industry consulting experiences. His current research interests lie in cloud computing and data management and warehousing. His publications have appeared in conference proceedings in ACM SIGITE, ICEIS, IIE and ASEE. He is a member of the ACM.