Place Semantics into Context: Service Community Discovery from the WSDL Corpus

Qi Yu

College of Computing and Information Science
Rochester Institute of Technology
qi.yu@rit.edu

Abstract. We propose a novel framework to automatically discover service communities that group together related services in a diverse and large scale service space. Community discovery is a key enabler to address a set of fundamental issues in service computing, which include service discovery, service composition, and quality-based service selection. The standard Web service description language, WSDL, primarily describes a service from the syntactic perspective and rarely provides rich service descriptions. This hinders the direct application of traditional document clustering approaches. In order to attack this central challenge, the proposed framework applies Non-negative Matrix Factorization (NMF) to the WSDL corpus for service community discovery. NMF has demonstrated its effectiveness in clustering high-dimensional sparse data while offering intuitive interpretability of the clustering result. NMF-based community discovery is further augmented via semantic extensions of the WSDL descriptions. The extended semantics are first computed based on the information sources outside the WSDL corpus. They are then seamlessly integrated with NMF, which makes the semantic extensions fit in the context of the original services. The experiments on real world Web services are presented to show the effectiveness of the proposed framework.

1 Introduction

Web services are increasingly being adopted to access data and applications across the Web [19]. This has been largely the result of the huge investment in Web application development and the many standardization efforts to describe, advertise, discover, and invoke Web services [3]. The emergence of cloud infrastructure also offers a powerful yet economical platform that greatly facilitates the development and deployment of a large number of Web services. Based on the most recent statistics, there are 28,593 Web services being provided by 7,728 distinct providers over the world and these numbers keep increasing in a fast rate [1]. Despite the abundance of various supporting technologies to facilitate the access to these Web services, there currently lacks a meaningful organization of the large and diverse Web service space. Most current Web services exist on the Web in a disorganized manner, which poses significant challenges for users to fully leverage the wealthy computing resources offered by these services.

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

G. Kappel, Z. Maamar, H.R. Motahari-Nezhad (Eds.): ICSOC 2011, LNCS 7084, pp. 188–203, 2011.

Discovery of service communities that group together related services is a key enabler to address a set of fundamental issues in service computing that include service discovery, service composition, and quality based service selection:

- **Service discovery:** Searching a service with a desired functionality can be performed solely within the service communities that offer relevant functionalities. This not only increases the searching accuracy but also significantly reduces the searching time because services from irrelevant communities are directly filtered out.
- **Service composition:** 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 selection:** As competing Web services that offer "similar" functionalities will be 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.

Existing efforts in constructing service communities can be categorized into either *top-down* or *bottom-up* approaches. A top-down approach usually 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), where a centralized control on the services can be enforced. Unfortunately, when a large scale of Web services from an open environment (e.g., the Web) are considered, the top-down strategy presents key challenges. One 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.

Bottom-up approaches directly infer service communities from the Web service descriptions. Most existing Web services are described using the standard Web service description language, WSDL. However, WSDL primarily describes a service from the syntactic perspective and rarely provides rich service descriptions [7]. 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, both of which, however, suffer some major issues.

- **The first category** aims to fully exploit the information carried by the WSDL service descriptions [7,8,13,12]. 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 [7]. 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.*

- **The second category**, on the other hand, explores external information sources, such as WordNet, Wikipedia, and search engines, to extend WSDL with rich semantics [11,2]. 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 a novel framework to discover service communities that group together related services from diverse and large scale Web services. We adopt the bottom-up strategy so that the communities can be automatically discovered from the WSDL corpus. 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. NMF-based community discovery is further augmented via semantic extensions of the WSDL descriptions. 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 [5,20]. 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 [10]. 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.

Outline: The remainder of the paper is organized as follows. We propose a framework for service community discovery in Section 2. The cornerstone of the proposed framework is the usage of Non-negative Matrix Tri-Factorization (NMTF) to simultaneously cluster services and operations. We present a strategy for computing the semantic extensions of the WSDL corpus in Section 3. We then elaborate on how to integrate the extended semantics into the community discovery framework. We evaluate the effectiveness of the proposed service community discovery framework via real-world Web services in Section 4. We give an overview of related work in Section 5 and conclude in Section 6.

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

Table 1. 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 $\mathbf{X}$

$\mathbf{o}_j$), 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 1 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 [9]. This dataset consists of over 450 services from 7 different 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

[2] Other values, such as the TFIDF score [1], can also be used.

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 $\hat{o}_2$.

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

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:

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

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:

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

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

3 Semantic Extension Integration

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 [7]. 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 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 [11,2]. 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.

3.1 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 [16] 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 4 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 Semantic Extension 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

$$\mathcal{R} = \sum_{p=1}^{k} \mathcal{R}_p = \mathrm{Tr}(\mathbf{S}^T \mathbf{L} \mathbf{S}) \tag{6}$$

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{7}$$

$$\mathbf{D}(i, i) = \sum_{j} \mathbf{W}(i, j) \tag{8}$$

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).

Table 2. 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

Thus, service community discovery with semantic extensions can be formulated as the following optimization problem:

$$\min_{\mathbf{S}\geq 0,\mathbf{R}\geq 0,\mathbf{O}\geq 0} ||\mathbf{X} - \mathbf{SRO}^T||_F^2 + \lambda \mathrm{Tr}(\mathbf{S}^T\mathbf{LS}) \tag{9}$$

where λ is the regularization parameter. The above optimization problem can be solved by using an iterative approach that exploits the auxiliary functions and the optimization theory [10].

4 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 [9]. The WSDL corpus consists of over 450 services from 7 different application domains. Table 2 lists the number of services from each domain.

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 [20]. 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 $\mathbf{X}$.

4.1 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. We adopt two metrics to measure the community discovery performance: ACcuracy (i.e., AC) and Mutual Information (i.e., MI). Both AC and MI are widely used metrics to assess the performance of clustering algorithms [17,4].

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:

$$AC = \frac{\sum_{i=1}^{m} \delta(z_i, map(y_i))}{m} \tag{10}$$

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 [14].

MI **metric:** Let D be the set of application domains obtained from the WSDL corpus and C be the service communities obtained from the proposed community discovery framework. The mutual information metric $MI(D, C)$ is defined as follows:

$$MI(D, C) = \sum_{\hat{d}_i \in D, \hat{c}_j \in C} p(\hat{d}_i, \hat{c}_j) \log_2 \frac{p(\hat{d}_i, \hat{c}_j)}{p(\hat{d}_i)p(\hat{c}_j)} \tag{11}$$

where $p(\hat{d}_i)$ and $p(\hat{c}_j)$ are the probabilities that a randomly selected service from the corpus belongs to domain $\hat{d}_i$ and community $\hat{c}_j$, respectively. $p(\hat{d}_i, \hat{c}_j)$ is the joint probability that the randomly selected service belongs to both domain $\hat{d}_i$ and community $\hat{c}_j$.

4.2 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 [20]. 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

Table 3. AC and MI Performance Comparison

Algorithm notation	AC (%)	MI (%)
NMTF + Semantics	**55.0**	**47.1**
NMTF	52.5	46.2
SVD Co-clustering	45.5	36.0
Semantic k-means	45.0	28.4

the TFIDF scores of the top-r terms returned by a web search engine. Refer to Section 3 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 and NMTF 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.

We use the notation NMTF+Semantics to denote the proposed algorithm that integrates NMTF with the semantic extensions of the WSDL corpus. The notations for other algorithms are also listed in Table 3. 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 $\mathbf{S}^T\mathbf{XO}$ [6]. We run each algorithm 200 times and the average AC and MI are reported.

4.3 Performance Comparison

Table 3 compares the AC and MI performance of four different algorithms. NMTF+Semantics generates the best results on both AC and MI over all the algorithms. Thus, the results clearly demonstrate the effectiveness of the proposed service discovery framework. It is also worth to note that semantic k-means reports the lowest performance on both AC and MI. This also justifies that using semantic extensions without considering the context of the original services does not necessarily benefit community discovery.

To further illustrate the performance difference, Figure 2 shows the confusion matrices with the best AC performances from the four different algorithms. As can be seen, NMTF+semantics achieves a best AC of 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

	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)

Fig. 1. Confusion Matrices with the best AC performances. (a) NMTF+Semantics: $AC = 64.4\%$; (b) NMTF: $AC = 62.8\%$; (c) SVD Co-clustering: $AC = 47.6\%$; (c) 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.

correctly clusters 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.

5 Related Work

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

5.1 Service Community Discovery

A WSDL clustering technique is proposed in [8] 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 [12]. 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 [20] 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. NMTF is seamlessly integrated with the semantic extensions of the WSDL corpus to further improve the performance of service community discovery.

5.2 Service Search and Discovery

Woogle, a Web service search engine, is developed in [7] 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 [13] for service discovery. A service aggregation graph is also proposed to facilitate service composition. A service discovery approach is proposed in [15] 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 [11,2]. 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.

5.3 Service Selection

Service selection aims to find a proper service provider with the best user desired quality of service (e.g., latency, fee, and reputation) [18,21,22]. 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.

6 Conclusion and Future Directions

We present a novel framework that amalgamates Non-negative Matrix Tri-Factorization (NMTF) and the semantic extensions of the WSDL corpus for service community discovery. 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. The amalgamation of NMTF and the semantic extensions of the WSDL descriptions places the extended semantics into the context of the service, which enable to more effectively leverage the semantics to benefit 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.

One interesting direction that we plan to explore is to include prior knowledge or background information to further improve the performance of service

community discovery. A useful type of prior knowledge is the pairwise constraint that specifies whether two services should belong to the same community or not. Such kind of prior knowledge is usually easier to get than relying on the domain experts to actually label a number of services. In this regard, it is worthwhile to investigate how to use this specific type of supervisory information to benefit service community discovery.

References

1. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc, Boston (1999)
2. Bose, A., Nayak, R., Bruza, P.: Improving web service discovery by using semantic models. In: Bailey, J., Maier, D., Schewe, K.-D., Thalheim, B., Wang, X.S. (eds.) WISE 2008. LNCS, vol. 5175, pp. 366–380. Springer, Heidelberg (2008)
3. Bouguettaya, A., Yu, Q., Liu, X., Malik, Z.: Service-centric framework for a digital government application. In: IEEE Transactions on Services Computing, vol. 99(PrePrints) (2010)
4. Cai, D., He, X., Han, J.: Document clustering using locality preserving indexing. IEEE Trans. Knowl. Data Eng. 17(12), 1624–1637 (2005)
5. Dhillon, I.S.: 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, pp. 269–274. ACM, New York (2001)
6. Ding, C.H.Q., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering. In: KDD, pp. 126–135 (2006)
7. Dong, X., Halevy, A., Madhavan, J., Nemes, E., Zhang, J.: Similarity search for web services. In: VLDB 2004: Proceedings of the Thirtieth International Conference on Very Large Data Bases, pp. 372–383, VLDB Endowment (2004)
8. Elgazzar, K., Hassan, A.E., Martin, P.: Clustering wsdl documents to bootstrap the discovery of web services. In: ICWS, pp. 147–154 (2010)
9. Klusch, M., Fries, B., Sycara, K.: Automated semantic web service discovery with owls-mx. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS 2006, pp. 915–922. ACM Press, New York (2006)
10. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
11. Liu, F., Shi, Y., Yu, J., Wang, T., Wu, J.: Measuring similarity of web services based on wsdl. In: ICWS, pp. 155–162 (2010)
12. Liu, W., Wong, W.: Discovering homogenous service communities through web service clustering. In: Kowalczyk, R., Huhns, M.N., Klusch, M., Maamar, Z., Vo, Q.B. (eds.) SOCASE 2008. LNCS, vol. 5006, pp. 69–82. Springer, Heidelberg (2008)
13. Liu, X., Huang, G., Mei, H.: Discovering homogeneous web service community in the user-centric web environment. IEEE T. Services Computing 2(2), 167–181 (2009)
14. Lovasz, L.: Matching Theory (North-Holland Mathematics Studies). Elsevier Science Ltd. (1986)
15. Ma, J., Zhang, Y., He, J.: Efficiently finding web services using a clustering semantic approach. In: CSSSIA 2008: Proceedings of the 2008 International Workshop on Context Enabled Source and Service Selection, Integration and Adaptation, pp. 1–8. ACM, New York (2008)

16. Sahami, M., Heilman, T.D.: 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 2006, pp. 377–386. ACM, New York (2006)
17. Xu, W., Liu, X., Gong, Y.: 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 2003, pp. 267–273. ACM, New York (2003)
18. Yu, Q., Bouguettaya, A.: Framework for web service query algebra and optimization. TWEB 2(1) (2008)
19. Yu, Q., Liu, X., Bouguettaya, A., Medjahed, B.: Deploying and managing web services: issues, solutions, and directions. VLDB Journal 17(3), 537–572 (2008)
20. Yu, Q., Rege, M.: On service community learning: A co-clustering approach. In: ICWS, pp. 283–290 (2010)
21. Yu, T., Zhang, Y., Lin, K.-J.: Efficient algorithms for web services selection with end-to-end qos constraints. ACM Trans. Web 1(1), 6 (2007)
22. Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., Chang, H.: Qos-aware middleware for web services composition. IEEE Trans. Softw. Eng. 30(5), 311–327 (2004)