International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.8 (2016), pp.345-358
http://dx.doi.org/10.14257/ijmue.2016.11.8.35

WSDL Retrieval for Web Services Based on Hybrid SLVM

Luda Wang[1,2], Peng Zhang[1*] and Shouping Gao[1]

[1]*Xiangnan University, Chenzhou, China*
[2]*School of Information Science and Engineering, Central South University, Changsha, China*
wang_luda@163.com, mimazp@126.com, gaoshoup@263.net

Abstract

Recently, two operable WSDL retrieval approaches, bipartite-graph matching and KbSM, were developed for Web service discovery. But their models and similarity metrics of WSDL ignore some term or semantic feature, and involve formal method problem of representation or difficulty of parameter verification. SLVM approaches depend on statistical term measures to implement XML document representation. Consequently, they ignore the lexical semantics and the relevant mutual information, leading to dataset analysis errors. This work proposed a service retrieval method, hybrid SLVM of WSDL, to address above problems in feature extraction. This method constructed a lexical semantic spectrum using WordNet for characterizing the lexical semantics, and built a special term spectrum based on TF-IDF. Then, feature matrix for WSDL representation was built on the hybrid SLVM. Applying to NWKNN algorithm, on dataset OWLS-TC-v2, our method achieves better F1 measure and query precisions than bipartite-graph matching and KbSM.

Keywords*: WSDL Retrieval, Web Service, SLVM, Lexical Semantics*

1. Introduction

Web service is an important paradigm for developing SOA (Service-Oriented Architecture) software. Growing number of web services raise the issue of efficiently locating the desired web services [1]. Many approaches have been proposed with respect to the way in which services are described. As the dominant approach, The Web Services Description Language (WSDL) is an XML-based interface definition language that is used for describing the functionality offered by a web service.

According to SOA software, at design time, developers identify which activity is to be performed and then try to find and select the Web services closest to such requirements. A hotspot in the research of service is to realize large-scale service retrieval and integration in Internet. In Universal Description Discovery and Integration (UDDI), conventional matching mechanisms are mainly based on keyword search, however, the precision of those retrieval approaches are relatively low. [2] To address this problem, several projecting approaches considering structural information have sprung up.

The two-phase similarity metrics-based bipartite-graph matching [1] and the Kernel based Structure Matching [2] are proposed for Web services retrieval. In particular, relative to WSDL, the Structured Link Vector Model (SLVM) was proposed for representing XML documents [3]. The bipartite-graph matching cannot apply to VSM-based analysis algorithms, and it cannot characterize the terms which are not recognized by semantic knowledge. At a disadvantage, the Kernel based Structure Matching depends on crude TF-IDF term statistic, meanwhile, its key parameter is difficult to verify. For XML document representation, Structured Link Vector Model depends on statistical term

* Corresponding Author

ISSN: 1975-0080 IJMUE

measure for feature extraction. However, in the information retrieval field, statistical term measures cause semi-structured document analysis to perform on the level of term string basically, and neglect lexical semantic content in the structural elements.

Semantic approach is an effectively used technology for document analysis. It can capture the semantic features of words under analysis, and based on that, characterizes and analysis the document. Close relationship between the syntax and the lexical semantics have attracted considerable interest in both linguistics and computational linguistics. For WSDL document analysis, the design and implementation of hybrid SLVM take account of both the lexical semantics and the terms which are not recognized by semantic knowledge particularly. Unlike present models, using WordNet [4], our model developed a new representation of WSDL which can characterize the lexical semantics and term feature, and provides a practical method for WSDL document similarity metrics. Theoretical analysis and relevant experiments are carried out to verify the effectiveness of this model.

2. Related Work

2.1. WSDL Retrieval Approaches and SLVM

Web services retrieval [5] assumes that the Web service interfaces are defined with Web Service Description Language (WSDL) and the algorithm combines the analysis of their structures and the analysis of the terms used inside them. In a WSDL description, data types are expressed by XML Schema Definition (XSD) specifications. Currently, two main approaches of WSDL analysis have emerged and improved to achieve service retrieval, which are the two-phase similarity metrics-based bipartite-graph matching [1] and the Kernel based Structure Matching [2]. In relation to WSDL, Structured Link Vector Model was proposed for representing XML documents [3].

Matching bipartite-graph [5] based on two-phase similarity metrics, firstly employs the external knowledge to compute the semantic distance of terms from two compared services. Services similarity is measured upon these distances. This method can reflect the underlying semantics of web services by utilizing the terms within WSDL fully. Bipartite-graph matching can support the similarity metric of WSDL structure, but cannot provide the vector representation of service. As a result, the two-phase similarity metrics based bipartite-graph matching cannot apply to VSM-based analysis algorithms. Besides, it cannot characterize the special terms which are not recognized by semantic knowledge.

The Kernel based Structure Matching (abbreviated as KbSM) contains essential mechanisms as follows [2]. To describe the structural similarity, it extracts document trees from the two WSDL documents and aligned their nodes according to the label's textual similarity. Aligned nodes are considered identical. After that, it models the documents trees as a vector in a n-spectrum vector space (n = 2; 3; …), and use the n-spectrum kernel function [6] to compare common hierarchical relationships between the two trees. To calculate two WSDL documents' text similarity with classical VSM, the other kind of feature extraction calculates text similarity using TF-IDF. Then, it combines structural and textual similarities to estimate the functional similarity. According to the mechanisms, the label's textual similarity and text similarity are calculated based on crude TF-IDF term statistic. Besides, as a key parameter, aligned threshold of nodes is difficult to verify.

Analogous to KbSM in structural similarity and kernel function, Structured Link Vector Model proposed by Jianwu Yang [3], which forms basis of our work, was proposed for representing XML documents. It was extended from the conventional vector space model (VSM) by incorporating document structures (represented as term-by-element matrices), referencing links (extracted based on IDREF attributes), as well as element similarity (represented as an element similarity matrix).

SLVM represents a semi-structured document docx using a document feature matrix $\Delta_x \in R^{n' \times m}$, given as [3]

$$\Delta_x = [\Delta_{x(1)}, \Delta_{x(2)}, \cdots\cdots, \Delta_{x(m)}], \quad (1)$$

Where m is the number of distinct semi-structured document elements, $\Delta_{x(i)} \in R^n$ is the TF-IDF feature vector representing the i^{th} XML structural element (ei), given as $\Delta_{x(i,j)} = TF(w_j, doc_x.e_i) \times IDF(w_j)$, and $TF(w_j, doc_x.e_i)$ is the frequency of the term w_j in the element e_i of docx.

When the kernel function $k(x_i, x_j)$ [6] is regarded as the similarity function between two XML documents, the SLVM-based document similarity between two semi-structured documents doc_x and doc_y is defined as [3]

$$k(x_i, x_j) = Sim(doc_x, doc_y) = \sum_{j=1}^{m} \sum_{i=1}^{m} M_{e(i,j)} \times (\tilde{\Delta}_{x(i)}^{T} \cdot \tilde{\Delta}_{y(j)}), \quad (2)$$

where $\tilde{\Delta}_{x(i)}$ or $\tilde{\Delta}_{y(j)}$ is the normalized document feature matrix of documents doc_x or doc_y, and M_e is a matrix of dimension $m \times m$ and named as the element similarity matrix. The matrix M_e captures both the similarity between a pair of document structural elements as well as the contribution of the pair to the overall document similarity. To obtain an optimal M_e for a specific type of semi-structured data, SLVM-based document similarity learn the matrix using pair-wise similar training data (unsupervised learning) in an iterative manner [7].

2.2. The Analysis of WSDL Similarity Metrics

As a new bipartite graph model for service retrieval, two-phase similarity metrics based bipartite-graph matching [1], cannot approach formalization of VSM. Besides, it cannot characterize the special terms which are not recognized by semantic knowledge. In WSDL description, a number of special terms exist in XML Schema, such as ‘tns’, ‘xsd’ and many special abbreviations. This omission causes the WSDL similarity metrics to lose the mutual information [8] which comes from special terms in different document.

Example 1.

<!--*Doc A.*--> <?xml version="1.0" encoding="UTF-8"?><definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="urn:timeserviceMyTime" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" name="MyTime" targetNamespace="urn:timeserviceMyTime"> <types/> <message name="MyTimeIF_getTime"> <part name="String_1" type="xsd:string"/></message> <message name="MyTimeIF_getTimeResponse"> <part name="result" type="xsd:string"/></message> <portType name="MyTimeIF"> <operation name="getTime" parameterOrder="String_1"> <input message="tns:MyTimeIF_getTime"/> <output message="tns:MyTimeIF_getTimeResponse"/></operation></portType> <binding name="MyTimeIFBinding" type="tns:MyTimeIF"> <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/> <operation name="getTime"> <soap:operation soapAction=""/> <input> <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded" namespace="urn:timeserviceMyTime"/></input> <output> <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded" namespace="urn:timeserviceMyTime"/></output></operation></binding> <service name="MyTime"> <port name="MyTimeIFPort" binding="tns:MyTimeIFBinding"> <soap:address location="http://127.0.0.1:9090/cssiunam/Time-deploy/Time" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/></port></service></definitions>	<!--*Doc B.*--> <?xml version="1.0" encoding="UTF-8"?><definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="urn:ClockserviceMyClock" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" name="MyClock" taracquireNamespace="urn:ClockserviceMyClock"> <types/> <message name="MyClockIF_acquireClock"> <part name="String_1" type="xsd:string"/></message> <message name="MyClockIF_acquireClockReply"> <part name="solution" type="xsd:string"/></message> <portType name="MyClockIF"> <operation name="acquireClock" parameterOrder="String_1"> <input message="tns:MyClockIF_acquireClock"/> <output message="tns:MyClockIF_acquireClockReply"/></operation></portType> <binding name="MyClockIFBinding" type="tns:MyClockIF"> <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/> <operation name="acquireClock"> <soap:operation soapAction=""/> <input> <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded" namespace="urn:ClockserviceMyClock"/></input> <output> <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded" namespace="urn:ClockserviceMyClock"/></output></operation></binding> <service name="MyClock"> <port name="MyClockIFPort" binding="tns:MyClockIFBinding"> <soap:address location="http://127.0.0.1:9091/cssiunam/Clock-deploy/Clock" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/></port></service></definitions>

These structural similarity models based on TF-IDF are perceived as the mode using statistical term measures. As a sort of ontology methods [9], semi-structured document representations based on statistical term measures ignore recognition of lexical semantic contents. It causes the WSDL similarity metrics to lose the mutual information [8] of term meanings which comes from synonyms in different document. Our comment on statistical term measures and semi-structured document representations can be clarified by analyzing a small XML corpus *Example 1*.

$$\Delta_A = \begin{array}{c} \begin{matrix} types & message & portType & binding & \end{matrix} \\ \begin{bmatrix} 0 & 4 & 2 & 5 & time \\ 0 & 0 & 0 & 0 & clock \\ 0 & 2 & 3 & 1 & get \\ 0 & 0 & 0 & 0 & acquire \\ 0 & 1 & 1 & 0 & response \\ 0 & 0 & 0 & 0 & reply \\ 0 & 1 & 0 & 0 & result \\ 0 & 0 & 0 & 0 & solution \end{bmatrix} \end{array} \quad \Delta_B = \begin{array}{c} \begin{matrix} types & message & portType & binding & \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & time \\ 0 & 4 & 2 & 5 & clock \\ 0 & 0 & 0 & 0 & get \\ 0 & 2 & 3 & 1 & acquire \\ 0 & 0 & 0 & 0 & response \\ 0 & 1 & 1 & 0 & reply \\ 0 & 0 & 0 & 0 & result \\ 0 & 1 & 0 & 0 & solution \end{bmatrix} \end{array}$$

Figure 1. The Term Matrices for Example 1

In *Example 1*, the two simple WSDL documents are viewed as two samples. Evidently, the term meanings of *Doc A* and *Doc B* are extremely equivalent. Thus, the correlation and semantic similarity between two samples are considerable. But, TF-IDF term statistic cannot characterize the document similarity between them. According to the WSDL description, the methods based on TF-IDF can succinctly characterize two documents using two structuralized matrices shown in Figure1, which select uncommon features using Information Gain [10] and Mutual Information [10]. Obviously, in Figure 1, because two feature matrices make each $\tilde{\Delta}_{A(i)}^{T} \cdot \tilde{\Delta}_{B(j)} = 0$, so the $Sim(doc_A, doc_B) = 0$, using the Eq. (2). Then, on behalf of statistical term measures, the document representations on *Example 1* did not perform well for semantic similarity.

3. Proposed Program

3.1. Preliminary Conception and Theoretical Analysis

For Web services retrieval, WSDL document representations which depend on statistical term measures shall lose mutual information of term meanings. Besides, using semantic distance, bipartite-graph matching can characterize the feature of term meanings but shall lose mutual information of special terms.

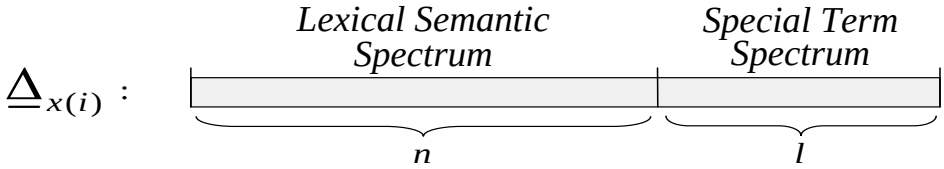

Figure 2. The Hybrid Eigenvector in SLVM

To solve these problems, utilizing SLVM, the motivation is connecting a lexical semantic vector with a special term vector to form the hybrid eigenvector of WSDL element. In hybrid eigenvector space of WSDL element, the eigenvector consist of lexical semantic spectrum and special term spectrum (shown in Figure 2). On the basis of synset, this method ought to construct a synset vector for characterizing lexical semantic contents, and as a template synset vector shall further characterize semantic relations to accomplish lexical semantic spectrum. Additionally, special term spectrum shall characterize the unusual string in documents, which is neither the recognizable word in

WordNet nor function word [11]. Formally, $\underline{\Delta}_{x(i)} \hat{I} R^{n+l}$ is the hybrid eigenvector representing the i^{th} WSDL structural element (e_i), where n is the scope of lexical semantic spectrum and l is the scope of special term spectrum.

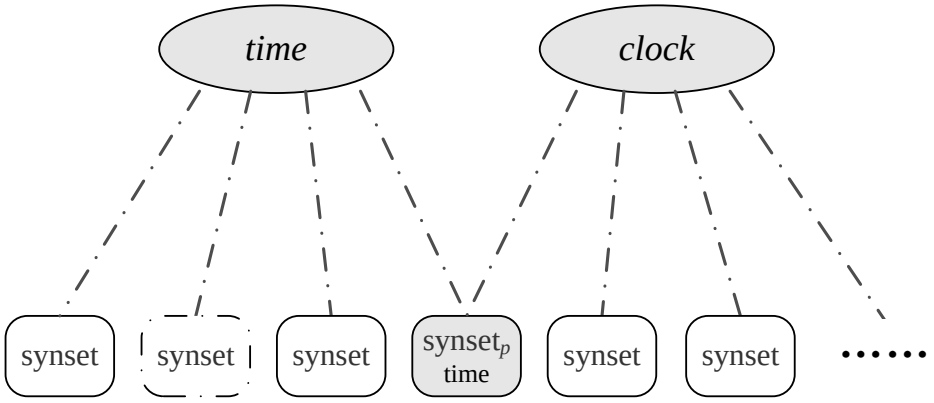

Figure 3. Common Semantic-Factor of Words

In WordNet, because one word or term refers to particular synonym sets, several particular synonym sets can strictly describe the sense of one word for characterizing lexical semantic contents. Then, our method defines these particular synsets as the semantic-factors of word. Thus, lexical semantic spectrum resorts to WordNet [12], a lexical database for English, for extracting lexical semantic contents. Then, the WSDL document representation shall preliminarily construct the synset VSM of WSDL elements.

Based on the above definition, involved semantic-factors can characterize the lexical semantic contents of *Example 1*, which can accomplish feature extraction of lexical semantic contents. For instance, in Figure 3, the words time and clock belong to different documents in *Example 1*, and the common synset time can represent mutual information [13] between lexical semantic contents. Then, lexical semantic spectrum shall capture the mutual information of lexical semantic contents between samples which lies in same semantic-factors of different elements or documents.

Moreover, to characterize the semantic relations of terms, weights of lexical semantic relations shall be marked on the vector in synset VSM, using Antonymy, Hyponymy, Meronymy, Troponomy and Entailment between synsets [12]. Then, lexical semantic spectrum can capture mutual information from lexical semantic relations between different elements or documents. In addition, lexical semantic spectrum connects with special term vector to form the eigenvector in hybrid SLVM. In the special term vector, the weight of each term is computed using TF-IDF.

According to the statistical theory of communications, our motivation needs further analysis for theoretical proof. The analysis first introduces some of the basic formulae of information theory [2, 7], which are used in our theoretical development of samples mutual information. Now, let x_i and y_j be two distinct terms (events) from finite samples (event spaces) X and Y. Then, let X or Y be random variables representing distinct lexical semantic contents in samples X or Y, which occur with certain probabilities. In reference to above definitions, mutual information between X and Y, represents the reduction of uncertainty about either X or Y when the other is known. The mutual information between samples, $I(X;Y)$, is specially defined to be

$$I(X;Y) = \sum_{x_i \in X} \sum_{y_j \in Y} P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i)P(y_j)} . \quad (3)$$

In the statistical methods of feature extraction, probability $P(x_i)$ or $P(y_j)$ is estimated by counting the number of observations (frequency) of x_i or y_j in sample X or Y, and normalizing by N, the size of the corpus. Joint probability, $P(x_i, y_j)$, is estimated by counting the number of times (related frequency) that term x_i equals (is related to) y_j in the respective samples of themselves, and normalizing by N.

Taking the *Example 1*, between any term x_i in Sample A and any term y_j in Sample B, there is not any counting of times that x_i equals y_j. As a result, on corpus *Example 1*, the statistical term measures indicate $P(x_i, y_j) = 0$ and the samples mutual information $I(X;Y) = 0$. Thus, the analysis verifies that the statistical methods of feature extraction lose mutual information of term meanings.

As to semantic approach, for feature extraction of lexical semantic contents, our method uses several particular semantic-factors to describe the meaning of one word or term. In different samples, words can be related to other words by common semantic-factors or lexical semantic relations. Then, lexical semantic mutual information between samples, $I(X;Y)$, is re-defined to be

$$I(X;Y) = \sum_{x_i \in X} \sum_{y_j \in Y} F(\mathrm{e}_{x_i, y_j}) \bmod N \log \frac{F(\mathrm{e}_{x_i, y_j}) \bmod N}{F(\mathrm{e}_{x_i}) \bmod N \times F(\mathrm{e}_{y_j}) \bmod N} . \quad (4)$$

To denote probability $P(x_i)$ or $P(y_j)$, function $F(\mathrm{e}_{x_i})$ or $F(\mathrm{e}_{y_j})$ is estimated by calculating the frequency of semantic-factors that describe the meaning of x_i or y_j in sample X or Y, and modulo N, the total number of semantic-factors in corpus. Meanwhile, to denote joint probability $P(x_i, y_j)$, function $F(\mathrm{e}_{x_i, y_j})$ is estimated by calculating the frequency of common semantic-factors that relate to lexical semantic contents or relations of x_i and y_j, and modulo N.

For lexical semantic feature, in *Example 1*, the frequency of semantic-factors are calculated by marking the lexical semantic contents and relations, joint probability $P(x_i, y_j)$ is estimated by counting the frequency of the common semantic-factors, and modulo N. For instance, the words time and clock are described by the common semantic-factor time (shown in Figure 3). In reality, $P(time, clock) = [F(\text{common-synset}_{time\text{-}clock}) + F(\text{relative-synset}_{time\text{-}clock})] \bmod N > 0$. Note that, the $F(\text{common-synset}_{time\text{-}clock})$ denotes the frequency of the common semantic-factor, which is caused by the lexical semantic contents, such as synset time, and the $F(\text{relative-synset}_{time\text{-}clock})$ denotes the frequency of the relative semantic-factors, which are caused by the lexical semantic relations, such as Antonymy, Hyponymy, Meronymy, Troponomy and Entailment. As a result, lexical semantic mutual information between Sample A and Sample B, $I(X;Y)$, is positive. Thus, the analysis proves that the semantic-factors and extraction of lexical semantic feature can provide the probability-

weighted amount of information (PWI) [3] between term meanings of documents on the lexical semantic level.

3.2. Hybrid SLVM of WSDL

In this work, WSDL documents are represented using the hybrid SLVM. In the model, each WSDL document is represented by a WSDL feature matrix in the structured link vector space. For organizing the hybrid SLVM, the procedures of hybrid eigenvector representing the XML structural element in WSDL are as follows.

In the first place, (1) for feature extraction of lexical semantic contents, our model makes a data structure of semantic-factor information. Secondly, (2) using semantic-factors, it constructs synset vector as template vector of the lexical semantic spectrum in the hybrid eigenvector of each XML structural element in WSDL. Last, (3) to characterize lexical semantic relations, it marks each template vector with weights of 5 semantic relations between synsets [12]. Thus, (4) hybrid eigenvector is constructed via connecting lexical semantic spectrum to special term spectrum, in which the latter is TF-IDF vector of special terms.

(1) The data structure of semantic-factor information comprises relevant information of each semantic-factor in a document sample. As a metadata, the data structure is shown in Table 1. It can record all important information of semantic-factors in a XML structural element of WSDL, such as synset ID, frequency, sample ID, element ID and relevant information.

(2) As the template of lexical semantic spectrum in hybrid SLVM, synset vector of WSDL element is constructed using semantic-factors. In WSDL dataset, all referred synsets are fixed by corresponding semantic-factors. Then, each identical *Synset ID* of all semantic-factors fills one dimension in synset vector space respectively. And, each template vector of lexical semantic spectrum is built to characterize lexical semantic contents of WSDL element. In synset vector space, each template vector of WSDL element is just the lexical-semantic-content vector. Specifically, each WSDL element identified by *Element ID* is represented by the template vector of lexical semantic spectrum, for only characterizing lexical semantic contents. The synset vector space represents element e_i in WSDL doc_x, using a lexical-semantic-content vector $d_{x(i)} \hat{I} R^n$, given as

$$d_{x(i)} = \left(d_{x(i,1)}, d_{x(i,2)}, \cdots\cdots, d_{x(i,n)}\right)^T, \quad (5)$$

Where n is the number of identical *Synset ID* of all semantic-factors in WSDL dataset, $d_{x(i,j)}$ is the feature value on the j^{th} synset, given as $d_{x(i,j)} = FS(s_j, doc_x.e_i)$ for all $j=1$ to n. $FS(s_j, doc_x.e_i)$ is the corresponding Frequency of the j^{th} synset s_j in the element e_i of doc_x.

Table 1. Data Structure of Semantic-Factor Information

Item	Explanation
Synset ID	Identification of synonym set
Set of Synonym	Synonymy is WordNet's basic relation. WordNet uses sets of synonyms (synsets) to represent word senses.[12]
Frequency	Frequency of semantic-factor in the XML structural element (Occurrence of the synset in the

	WSDL element)
Sample ID	Identification of WSDL document
Element ID	Identification of structural element or unit in semi-structured document

(3) On the basis of synset vector space, to characterize lexical semantic relations, our method marks *Antonymy*, *Hyponymy*, *Meronymy*, *Troponomy* and *Entailment* on each dimension of the template vector. According to empirical parameter tune, the processing is formulized as

$$\Delta d_{x(i,j)} = \sum_{k=1}^{n} R(j,k) \times d_{x(i,k)}, \tag{6}$$

$$R(j,k) = \begin{cases} 0.4 & \textit{Antonymy} \\ 0.2 & \textit{Hyponymy, Meronymy, Troponomy} \text{ or } \textit{Entailment} \\ 0 & \text{Unrelated} \end{cases}, \tag{7}$$

where j and $k=1$ to n, n is dimensional number of the synset vector, and $d_{x(i,k)}$ is value of the k^{th} template vector element. $\Delta d_{x(i,j)}$ is semantic relation increment to the j^{th} dimensional value of template vector, and function $R(j,k)$ denotes semantic relation coefficient for $\Delta d_{x(i,j)}$. Specifically, when the synset of k^{th} dimension is related to synset of j^{th} dimension via semantic relation such as *Antonymy*, *Hyponymy*, *Meronymy*, *Troponomy* or *Entailment*, the $R(j,k)$ assignment is shown in Eq. (7), which are empirical values. The assignments of $R(j,k)$ reflect the semantic relations which are organized into synsets by WordNet.

$$\Delta_A = \begin{array}{c} \begin{array}{ccccc} \text{types} & \text{message} & \text{portType} & \text{binding} & \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & \vdots \\ 0 & 4 & 2 & 5 & \text{synset}_{time,clock} \\ 0 & 0 & 0 & 0 & \vdots \\ 0 & 2 & 3 & 1 & \text{synset}_{get,acquire} \\ 0 & 0 & 0 & 0 & \vdots \\ 0 & 1 & 1 & 0 & \text{synset}_{response,reply} \\ 0 & 0 & 0 & 0 & \vdots \\ 0 & 1 & 0 & 0 & \text{synset}_{result,solution} \\ \vdots & \vdots & \vdots & \vdots & \textit{special terms} \end{bmatrix} \end{array} \quad \Delta_B = \begin{array}{c} \begin{array}{ccccc} \text{types} & \text{message} & \text{portType} & \text{binding} & \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & \vdots \\ 0 & 4 & 2 & 5 & \text{synset}_{time,clock} \\ 0 & 0 & 0 & 0 & \vdots \\ 0 & 2 & 3 & 1 & \text{synset}_{get,acquire} \\ 0 & 0 & 0 & 0 & \vdots \\ 0 & 1 & 1 & 0 & \text{synset}_{response,reply} \\ 0 & 0 & 0 & 0 & \vdots \\ 0 & 1 & 0 & 0 & \text{synset}_{result,solution} \\ \vdots & \vdots & \vdots & \vdots & \textit{special terms} \end{bmatrix} \end{array}$$

Figure 4. The Lexical-Semantic and Special Term Matrices for Example 1

(4) As for WSDL dataset, all synset dimensions carry the corresponding semantic feature values. Then, hybrid SLVM represents a WSDL doc_x using a WSDL feature matrix $\Delta_x \in R^{(n+l)\times m}$, defined as

$$\underline{\Delta}_x = \left[\underline{\Delta}_{x(1)}, \underline{\Delta}_{x(2)}, \cdots\cdots, \underline{\Delta}_{x(m)}\right], \tag{8}$$

where m is the number of distinct WSDL elements, $\underline{\Delta}_{x(i)} \in R^{n+l}$ is the hybrid eigenvector representing the i^{th} structural element in WSDL, given as

$$\underline{\Delta}_{x(i,j)} = \begin{cases} d_{x(i,j)} + \square d_{x(i,j)} & j = 1 \text{ to } n \\ TF(w_{j-n}, doc_x.e_i) \times IDF(w_{j-n}) & j = n+1 \text{ to } n+l \end{cases}, \tag{9}$$

where n is the number of identical *Synset ID* of all semantic-factors in WSDL dataset, and l is the number of special terms in WSDL dataset. And $TF(w_{j-n}, doc_x.e_i) \times IDF(w_{j-n})$ for all j-n=1 to l, is the TF-IDF feature of the special term w_{j-n} in e_i , in which e_i is the i^{th} structural element of WSDL doc_x.

Consequently, in hybrid SLVM, eigenvector of WSDL element is constructed via connecting lexical semantic spectrum to special term spectrum, which is shown in Eq. (9). Therefore, in the structured link vector space, each WSDL document is represented by a WSDL feature matrix, and the WSDL similarity metrics employ Eq. (2). Then, on behalf of lexical-semantic and special term measures, hybrid SLVM can succinctly characterize two documents in *Example 1* using two structuralized matrices shown in Figure 4.

4. Experiment and Result

4.1 The Experiment

For the purpose of comparison with the current approaches in service retrieval, the WSDL dataset has been obtained from OWLS-TC version 2 [14]. All WSDL documents are translated from the original OWL-S document. Totally 581 services which are translated from OWLS to WSDL documents belonging to 7 categories are obtained after translation. Experiments are taken upon these services and the query requests in OWLS-TC.

The nature of the names normally exists in an automatically generated WSDL. Consequently, name similarity can be applied only after a tokenization process which produces the set of terms to be actually compared. For this reason, the experiments perform the tokenization to decompose a given name in its terms. [5] In most WSDL documents, the terms appearing in the portType names are the same as the definition of message types which is consistent with the naming convention of programmers and can be decomposed into individual terms. Thus, experiments utilize the decomposing rules from [5].

To evaluate the service classification, we use the $F1$ measure [15]. This measure combines recall and precision in the following way:

$$F1 = \frac{2 \times \text{Recall} \times \text{Precision}}{(\text{Recall} + \text{Precision})}. \tag{10}$$

Using $F1$ measure, we can observe the effect of different kinds of data on a service classification system [15]. For ease of comparison, we summarize the $F1$ scores over the different categories using the macro-averages of $F1$ scores, in the same way, we can obtain the Macro-Recall and Macro-Precision [15].

As the matching degree, metrics of service similarity hope that a query return can be adapted to replace the query in Web environment. Therefore, it should provide almost the same function to the query. The query return must be textually similar to the query document with structural similarity. Therefore, matching degree between the query and the WSDL documents decide the services of query return, and the numbers of services in query return are selected by matching degree descending-order. The mean precisions are

evaluated on Top5 and Top10 of service numbers in query return, and the average of all query precisions in which the top number *n* varies from 1 to the number of all the relevant services for the given query. [1]

In this work, experiments use three WSDL retrieval approaches: 1) two-phase similarity metrics-based bipartite-graph matching, 2) the KbSM, 3) WSDL retrieval based on hybrid SLVM.

For the hybrid SLVM, to tackle unbalanced WSDL dataset, the WSDL retrieval selects an optimized KNN classification, NWKNN (Neighbor-Weighted K-Nearest Neighbor) algorithm, defined as [16]

$$score(doc, c_i) = \text{Weight}_i \left(\sum_{d_j \in KNN(d)} Sim(doc, doc_j)\delta(doc_j, c_i) \right)$$

$$\text{subjected to} \quad \delta(doc_j, c_i) = \begin{cases} 1 & doc_j \in c_i \\ 0 & doc_j \notin c_i \end{cases}. \tag{11}$$

KNN(*d*) indicates the set of K-nearest neighbors of document *doc*. $\delta(doc_j, c_i)$ is the classification for document doc_j with respect to class c_i. For each WSDL document *doc*, the experiment first selects *K* neighbors among training documents contained in *K** categories $\{C_1^d, C_2^d, ..., C_{K^*}^d\}$. The Weight is obtained by Equ. (12), where Exponent>1. [16]

$$\text{Weight}_i = \frac{1}{(\text{Num}(C_i^d)/\text{Min}\{\text{Num}(C_l^d) | l = 1, ..., K^*\})^{1/\text{Exponent}}}. \tag{12}$$

In the process of Eq. (11), between representations of *doc* and doc_j [3], this algorithm uses similarity function in Eq. (2) to calculate the $Sim(doc, doc_j)$. Besides, according to experience of NWKNN algorithm [16], the parameter of Weight_i, Exponent [16], is equal to 3.5.

4.2. The Result

To evaluate the service classification, this work uses the *F*1 measure [15]. Then, we can observe the effect of different kinds of data on a classification system [15]. For ease of comparison, we summarize the *F*1 scores over the different categories, and display precision and recall. Table 2 manifests the performances of service classifications.

Table 2. Classification Evaluation

Approaches	Precision	Recall	*F*1
Bipartite-graph Matching [1]	67%	56.5%	61.3%
KbSM [2]	68.9%	41.9%	52.1%
Hybrid SLVM	73%	40%	51.7%

Table 3. Query Return Evaluation

Approaches	Top5-precision	Top10-precision	Average precision
Bipartite-graph Matching [1]	100%	90%	48%
KbSM [2]	78.6%	66.7%	64.3%
Hybrid SLVM	100%	95%	68.4%

To evaluate the service query, query return selects services in matching degree descending-order [1]. Then, we can observe the mean precisions on query return with different service numbers [1]. For ease of comparison, we also display the average of all query precisions. Table 3 manifests the performances of service query return.

5. Conclusion

In proceeding work, a data structure of semantic-factor information is constructed in order to form synset vector. After marking the lexical semantic relations on it, the synset vector composes lexical semantic vector of WSDL element. As the lexical semantic spectrum in hybrid eigenvector of WSDL element, lexical semantic vector can characterize lexical semantic contents and relations. In hybrid SLVM, hybrid eigenvector of WSDL element is constructed via connecting lexical semantic spectrum to special term spectrum. Using the NWKNN algorithm, the hybrid SLVM achieve better performance of classification and query return than the bipartite-graph matching model and the KbSM.

The future aim focuses on applying the hybrid SLVM to more current algorithms and developing a semantic knowledge base for Web service discovery.

Acknowledgements

This project was supported by the National Natural Science Foundation of China (Grant No. 61402165), Nature Science Foundation of Hunan Province of China (No. 2015JJ3058, 12JJ3069) and Informational-education Application Project of universities in Hunan Province. We would like to thank the anonymous referees for their helpful comments and suggestions. The authors are also grateful for the help of Xin-pan Yuan.

References

[1] F. F. Liu and Y. L. Shi, “Measuring Similarity of Web Services Based on WSDL”, Proceedings of the Web Services, Shanghai, China, **(2010)**.
[2] J. L. Yu, S. M. Guo, H. Su, and H. Zhang, “A Kernel-based Structure Matching for Web Services Search”, Proceedings of the 16th international conference on World Wide Web, New York, USA, **(2007)**.
[3] J. W. Yang and X. O. Chen, “A Semi—Structured Document Model for Text Mining”, Computer Science and Technology, vol. 17, no. 603, **(2002)**.
[4] G. A. Miller, “WordNet: a lexical database for English”, Communications of the ACM, vol. 38, no. 39, **(1995)**.
[5] P. Plebani and B. Pernici, “URBE: Web services retrieval Based on Similarity Evaluation”, Knowledge and Data Engineering, vol. 21, no. 1629, **(2009)**.
[6] J. S. Taylor and N. Cristianini, “Kernel Methods for Pattern Analysis”, Cambridge University Press, United Kingdom, **(2004)**.
[7] J. W. Yang, K. C. Wlliam and X. O. Chen, “Learning element similarity matrix for semi-structured document analysis”, Knowledge and Information Systems, vol. 19, no. 53, **(2009)**.
[8] K. W. Church and P. Hanks, “Word association norms, mutual information, and lexicography”, Computational Linguistics, vol. 16, no. 22, **(1990)**.
[9] D. Sanchez and M. Batet, “A semantic similarity method based on information content exploiting multiple ontologies”, Expert Systems with Applications, vol. 40, no. 1393, **(2013)**.

[10] C. C. Aggarwal and C. X. Zhai, "Mining Text Data: A survey of text classification algorithms", Springer Science & Business Media, New York, **(2012)**.
[11] S. Banerjee and T. Pedersen, "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet", Lecture Notes in Computer Science, vol. 2276, no. 136, **(2002)**.
[12] M. Lintean and V. Rus, "Measuring Semantic Similarity in Short Texts through Greedy Pairing and Word Semantics", Proceedings of the 25th International Florida Artificial Intelligence Research Society Conference, Stroudsburg, USA, **(2012)**.
[13] W. Zhang, T. Yoshida and X. Tang, "A comparative study of TF*IDF, LSI and multi-words for text classification", Expert Systems with Applications, vol. 83, no. 2758, **(2011)**.
[14] M. Klusch, B. Fries and K. Sycara, "Automated semantic web service discovery with OWLS-MX", Proceeding Autonomous Agents and Multiagent Systems, New York, USA, **(2006)**.
[15] C. J. Rijsbergen, "Information retrieval", Butterworths Press, London, **(1979)**.
[16] S. B. Tan, "Neighbor-weighted k-nearest neighbor for unbalanced text corpus", Expert Systems with Applications, vol. 28, no. 667, **(2005)**.

Authors

Luda Wang, He received the M.S. degree of Computer Application Technology from Hunan University in 2009, and he is currently a Ph. D. candidate of Computer Science and Technology in Central South University (CSU). His research interests include AI, Data Analysis and Information Retrieval.

Peng Zhang, She received the M.S. degree of Computer Application Technology from Hunan University in 2010. She has been a faculty member of Computer Science at Xiangnan University, China, since 2004, where she is currently a lecturer. Her research interests include Information Retrieval and Information Fusion.

Shouping Gao, He received the Ph. D. degree of Applied Mathematics from Tongji University in 2003, He has been a faculty member of Computer Science at Xiangnan University,China, since 1991, where he is currently a professor. His research interests include Data Analysis and Symbolic Computation.