Research on Web Information Retrieval based on Vector Space Model

Zhang Ji Bo Ning
College of Information Science and Technology, Hainan University, Hainan, Haikou 570228, China,
Email: zhangjiboning@gmail.com

Abstract—Existing methods and technologies for Web services discovery mainly consist of two models: service matching based on grammar and based on semantics. With keywords service matching method, we introduce text information retrieval technology with vectors and concept lattice theory. Then a web service discovery algorithm using eigenvectors is put forward. It orders the Web services by correlation among the services in Web service subset. The services which have bigger correlation will get anterior places. According to this algorithm a Web service discovery system based on eigenvectors is designed. We give description to key modules of the system. At last the improved algorithm is compared to methods based on full-text retrieval and pure similarity retrieval. The results show it has better performance in accuracy and response.

Index Terms—Web services, text information retrieval, eigenvectors, clustering, correlation

I. INTRODUCTION

Effective Web services discovery algorithm is very important for Web services performance. There are a lot of relevant researches referred in [1-6] have been carried out domestically and overseas. Based on the relied technological specification difference of service match, research techniques are divided into two categories: grammatically leveling service match and semantically leveling service match. The service match in grammar level is a kind of matching algorithm based on key words and this method is mainly on the basis of WSDL service to describe language. Besides, this matching algorithm adopts simple classification and key words match. For instance, the Web services matching algorithm adopted by Web services registration center: When Web services is registering in service registration center, relative information of service will be distributed to registration center. Service matching algorithm will be implemented accurate match based on some key words like service classification, service name, service symbols or service finite attributive value, etc. Then it tries to find out service corresponding to user requirement. The advantage of this algorithm is easy to implement and it has high query efficiency. But low precision ratio also exists. Domestic and overseas scholars have performed a lot of researching work and put forward a series of solution methods. Christian, etc, propose that vector space model can be applied to Web services discovery in [7-9]. Their idea is: Web services description information is decomposed at first, then, vector space based on key words is constructed and Web services is sought according to space vector similarity. Wu Jian, etc, brought forward the similarity matching algorithm based on TF/IDF in [10]. These methods improved retrieval accuracy of Web services discovery. However, due to high complexity of constructing space vector and low quality of Web services query, as referred in [11-12], Web services precision ratio needs further improvement. Domestic scholars brought forward Web services discovery methods based on clustering in [13-16], using clustering algorithm to classify Web services on method-level and similarity value to search Web services. This kind of description which is based on method layer cannot describe Web services completely. At the same time, this discovery algorithm performs description classification on Web services at the method layer granularity. The same service tends to be divided into multi categories simultaneously, which enlarges service retrieval space. So the retrieval efficiency of Web services will be reduced to some extent.

To sum up, the development based on semantic Web services discovery is not mature enough. It cannot be put into commercial implementation in large-scale. The Web services discovery based on grammar has been broadly applied but its precision ratio is relatively low. This paper’s research purpose is: on the basis of research achievements based on grammatical service discovery, to perform currently technological improvement on Web services expression and Web services clustering match. We also try to improve the precision of Web services and to reduce response time of Web services search. With information retrieval technique of space vector, we will reduce the complexity of constructing special vector and give reasonable classification to Web service. It will help to improve the service precision and query efficiency of Web services.

II. WEB SERVICES SYSTEM STRUCTURE

Systematic structure of Web service mainly involves three main characters and three main operations. Three characters refer to service provider, service client and service registration center. Three main operations refers to service publish, service discovery and service binding. Service provider carries out registering towards service at service registration center, service registration center carries out registering management towards service,
service client seeks needed service at service registration center and service will be carried out binding and invoking according to service description. The system structure of Web services is shown as the following figure:

![System structure of Web service.](image)

As a kind of coarse-grained and loosely coupled service structure, [17] performs communication through simple and accurate definition interface without involving bottom program interface and communication model. The components in SOA systematic structure must have one or more characters from following three characters: Service provider, Service requester, Service broker. The operations among characters are:

- Publishing operation: Service provider can register its function and access interface to Service broker;
- Finding operation: Service requester can search specific kinds of service through Service broker;
- Binding operation: Service requester can truly make use of Service provider.

### III. Web Services Discovery Technology Based on Eigenvector

#### A. Text Information Retrieval based on Vectors

The text information retrieval technology based on vector mainly adopts vector space model to express texts. It describes features of files with VSM and assigns weight to each feature item with TF/IDF formulas. Inverted files are used for indexing and cosine angle is used for correlation measurement. Recall and precision ratio are adopted to give evaluation on the performance of system.

**VSM** method [18] extracts a series of feature items in files to compose eigenvectors, and then some methods are given to assign weight. For example: If file D is expressed as \( (t_1, t_2, \ldots, t_N) \), \( t_i \) is feature item, \( i \in [1, N] \). According to the importance of each item weight \( w_i \) is assigned to each item. So file D can be expressed as \( (t_1, w_1t_1, w_2t_2, \ldots, w_Nt_N) \). Assume the feature items are different form each other. Then \( (t_1, t_2, \ldots, t_N) \) is looked as N-dimension coordinate system and corresponding \((w_1, w_2, \ldots, w_N)\) is vector in this space. \( D(w_1, w_2, \ldots, w_N) \) is the VSP of D. When information of the feature items are not taken into account, a file can be expressed as an eigenvector and a file set can be expressed as a matrix.

**Similarity** [19] is an important concept to measure the correlation between two files. For example: \( \text{Sim}(D_1, D_2) \) denotes the similarity between \( D_1 \) and \( D_2 \). When both files are described with eigenvectors, we can calculate the correlation of files with distance formulas of vectors. It has two forms.

**Inner-product distance:**

\[
\text{Sim}(D_1, D_2) = \sum_{i=1}^{N} w_{1i} * w_{2i}
\]  

**Cosine distance:**

\[
\text{Sim}(D_1, D_2) = \cos \theta = \frac{\sum_{i=1}^{N} w_{1i} * w_{2i}}{\sqrt{\sum_{i=1}^{N} w_{1i}^2 \sum_{i=1}^{N} w_{2i}^2}}
\]

From above formulas, space vector model involves two basic questions: how to select feature items and how to calculate feature items weight. Feature items selection is mainly influenced by some factors like handling speed, precision and storage space. So selecting feature items should follow the below principles:

- Feature items contain much information and it has strong expressing ability on document, that is, text content can be better reflected.
- Feature items have regularity on text distribution, which is easy for statistics.
- Feature items selection should be easy and feasible. It should consume less time and space complexity.

Feature items weight is mainly influenced by recall ratio and precision ratio. In order to ensure that retrieval system has higher recall and precision ratio, retrieval system should contain weight factors which can improve recall ratio and precision ratio. Weight factors include three parts: frequency factors, document set factors and standardization factors.

- If one specific feature item appears with high frequency in the document, the weight of it should also be high. Retrieval system makes use of TF to assign weight on feature items and applies query words with high frequency to retrieval, which can improve the systematic recall ratio.

- Only using term frequency cannot sufficiently guarantee the improvement of recall ratio and precision ratio. Thus, variables related to document set should be introduced to make more obvious discrimination among documents. If the percentage of feature items is low in document set, that is, it only appears a small part of document, the IDF which is Inverse Document Frequency in document set is very large. It is supposed that total number of document is N and document numbers including one feature item is n, the document set factor is \( \text{IDF} = \log(N/n) \).

- If the document is very large, the matching possibility between query expression and document is also large. Therefore, the probability that long document is sought out is higher than that in short document. In order to eliminate this effect, standardization factor is introduced. We set \( w \) as weight of feature item and the
definition of standardization factor is \( \frac{w}{\sqrt{\sum w_i}} \).

B. Service Discovery Model based on Eigenvector

(1) Web Services Vector

Web services can be expressed as 3-tuple [20]: \( WS=(WSB, WSF, QD) \). \( WSB \) is basic description of Web services and \( WSF \) is functional description of Web services. We adopt text expression retrieval technology in [21] to make use of space vector model to express Web services and to describe feature items of Web services from WSDL description document. We apply input/output information name and type to functionally describe these elements’ expressing Web services function, that is, the Web services functional vector \( WSF=(I, IT, O, OT, WSF) \). \( I \) refers to input information, \( IT \) refers to input information type, \( O \) refers to output information and \( WSF \) refers to description information with service function. \( WSF \) refers to description information of service function. In WSDL document structure, its sub-element \(<input>\) defines input information and its input information type, \(<output>\) element defines output information and its output information type and \(<input>\), \(<output>\) and \(<Port>\) commonly define and describe functions offered by Web services.

In addition to Web services function description, Web services description also contains Web services basic description which includes Web services names and text description. On WSDL document structure [22], \(<\text{name}>\) element defines Web services names and \(<\text{name}>\) element defines text description of Web services. Thus, basic description of Web services can describe these elements expression through Web services names and Web services text description, that is, \( WSB={SN, SD} \) with \( SN \) representing Web services names and \( SD \) representing text description of Web services. Thus, basic description of Web services can describe these elements expression through Web services names and Web services text description, that is, \( WSB={SN, SD} \) with \( SN \) representing Web services names and Web services text description, that is, \( WSB={SN, SD} \) with \( SN \) representing Web services names and \( SD \) representing text description of Web services.

Then, in service discovery models based on Eigenvectors, Web services are expressed as \( WS=(I, IT, O, OT, WSF, SN, SD) \). They are called Web services feature items. Web services vector is:

\[
\overrightarrow{WS}_i = \{w_{i1}, w_{i2}, w_{i3}, w_{i4}, w_{i5}, w_{i6}, w_{i7}\} \quad (3)
\]

\( w_{i1}, w_{i2}, w_{i3}, w_{i4}, w_{i5}, w_{i6}, w_{i7} \) are corresponding weight to \( I, IT, O, OT, WSF, SN, SD \) and its value range belongs to \([0,1] \).

(2) Correlation on Web Services

On the basis of text information and retrieval technique, \( Sim(D_1, D_2) \) is the similarity of file \( D \), that is, the correlation between \( D_1 \) and \( D_2 \). When \( D_1 \) and \( D_2 \) are expressed by eigenvectors, file \( D_1 \) is expressed as \( (w_{i1}, w_{i2}, \ldots, w_{ik}) \) and document \( D_2 \) is expressed as \( (w_{j1}, w_{j2}, \ldots, w_{jk}) \). Distance formula among vectors in [23] can be used to calculate documents similarity. Calculation formula of similarity is shown as formula 1.

In the discovery model based on eigenvectors, correlation measuring among Web services can adopt the method in text information retrieval. Assume given two Web service vectors \( \overrightarrow{WS}_i = \{w_{i1}, w_{i2}, \ldots, w_{im}\} \) & \( \overrightarrow{WS}_j = \{w_{j1}, w_{j2}, \ldots, w_{jm}\} \). Then the correlation between \( \overrightarrow{WS}_i \) and \( \overrightarrow{WS}_j \) can be expressed with cosine function of their corresponding vectors:

\[
Re(\overrightarrow{WS}_i, \overrightarrow{WS}_j) = \frac{\overrightarrow{WS}_i \cdot \overrightarrow{WS}_j}{||\overrightarrow{WS}_i|| \cdot ||\overrightarrow{WS}_j||} \quad (4)
\]

This paper set the threshold \( \tau \) to judge whether two Web services is correlated in advance, that is, while \( Re(\overrightarrow{WS}_i, \overrightarrow{WS}_j) \geq \tau \), we considers Web services WS and WS is correlated, otherwise, they are not.

(3) Web Services Clustering

The purpose of Web services clustering is collecting Web services with certain correlation to further support fast retrieval of Web services and to improve precision ratio of Web services discovery, referred in [24]. We introduces concept lattice in [25] and applies aggregation theory in concept lattice to Web services discovery to further put forward the Web services aggregation algorithm.

Web services aggregation is to divide Web services in Web services set into many services sub sets. In order to realize the perfect service aggregation and to further improve precision ratio of Web services discovery, Web services aggregation should ensure that Web services in the same service sub set is correlated and Web services in different Web services sub set is uncorrelated. That is, the below conditions should be satisfied. Towards randomly given two Web services set \( P_1 \) and \( P_2 \), \( w_{i4} \) and \( w_{i5} \) are two random services in Web Service set \( P_1 \) while \( w_{i6} \) is another service in Web services set \( P_2 \) which satisfies

\[
\begin{align*}
rel(\overrightarrow{WS}_{i4}, \overrightarrow{WS}_{i5}) & \geq \tau \quad \text{add } w_{i4} \text{ to subset } P_1; \\
rel(\overrightarrow{WS}_{i4}, \overrightarrow{WS}_{i5}) & < \tau \quad \text{add } w_{i5} \text{ to subset } P_2.
\end{align*}
\]

When Web service is clustering, the relevant threshold should be set to guarantee Web services cluster results satisfying above conditions.

First we need to construct a web services subset \( P_1 \) and randomly select a service \( w_s \) to put in \( P_1 \). Then select some service \( w_s \) in the set and calculate correlation \( rel_i \) between \( w_s \) and \( P_1 \). \( \tau \) is the threshold related to Web service and its subset. If \( rel_i(w_s, P_1) \geq \tau \), add \( w_s \) to subset \( P_1 \); Otherwise, construct a second Web services subset \( P_2 \) and add \( w_s \) to \( P_2 \). And so on, when calculating \( w_s \), assume k subsets have been constructed. Calculate the correlation of \( w_s \) and k subsets respectively. Select the maximum correlation to find corresponding subset \( P_k \); Otherwise, construct new Web services subset \( P_{k+1} \) and add \( w_s \) to \( P_{k+1} \). Cycle the process until the last service is added to suitable subset.
C. Web Services Discovery Algorithm

(1) Description Information Vectorization

Description information vectorization of Web services means that Web services description information is expressed as vectors. Its process is that Web services description information is decomposed into input/output information, information type of Web services, Web services names, Web services description and functional description of Web services. This information is seen as feature items of Web services and weight value of each feature item is acquired through the weight value formula.

\[
w_{ij} = \frac{\log |P|}{d_{freq}}
\]

In this formula, \(freq_{ij}\) is the appearing frequency of feature item \(J\) in all WSDL documents’ corresponding description element, \(d_{freq}\) is the totally appearing frequency of feature item \(J\) in all WSDL documents, \(|P|\) is service numbers in Web services set. After Web services vectorization, in order to calculate in the future easily, Web services vectorization is carried out for standardization.

Usually, Web services request submitted by users is text description. This paper resorts to feature items selected from Web services’ requesting text description to reflect features of users’ needing services. In order to guarantee precision ratio, Web services name \(QN\), Web services description \(QD\), Web services output information \(QO\), Web services output information and types \(QOT\), Web services input information type \(QIT\), Web services input information \(QI\), Web services functional information \(QF\) will be taken as feature items in Web services request information. In order to be easy to perform Web services discovery, Web services retrieval request is carried out vectorization in advance. In the service discovery model based on eigenvector, the method of vectorization applying information retrieval technology carried out vectorization on retrieval request and vectorization process is shown as the following:

It is supposed that Web services retrieval request is \(r_i\), \(k_j\) is a feature item of \(r_i\) described in text. The weight of \(k_j\) can be acquired by following formula.

\[
rw_{ij} = \frac{1}{2} \frac{f_{ij}}{\max_{y} f_{ij}^y} \log \frac{|P|}{n_i}
\]

In this formula, \(f_{ij}\) refers to retrieval request of Web services. \(|P|\) denotes the most frequency of feature items. \(|P|\) refers to totally containing service numbers in Web services set. \(n_i\) refers to Web services retrieval request number containing feature item \(k_j\). Similarly, request vectorization of Web services is performed standardization.

(2) Web Services Match

In service discovery model based on eigenvector and vectorization process is shown as the following:

Through Web services cluster algorithm, Web services set is divided into \(k\) Web services sub set. Through Web services vectorization, description information and request information of Web services is expressed as vector and sub set vector of Web services (each component’s averaging value of Web services vector in the set) is calculated. Correlation formula between Web services and Web services sub set can be applied to calculate the correlation value of Web services request \(Q\) and Web services sub set \(P\). If \(\text{Rel}(Q, P) > \varepsilon\), Web services request \(Q\) is correlated to Web services sub set \(P\). So the services contained in sub set of Web services are query Web services demanded by users.

The formal Languages of Web services matching are described as follows:

\[
\text{WS Vector Match()}
\]

\[
\text{Output: WS,Web Services sequence satisfying users’ demand;}
\]

\[
\text{R=new Request();}//Vector expression for request
\]

\[
\text{S=new WebService();}//Vector expression for Web service
\]

\[
\text{SC=new ServicesConcept();}//Service subset vector
\]

\[
\text{WS=null;}//Initialization of service output set
\]

\[
\text{For (int} i=1;i<=|SC||i++)
\]

\[
\text{Re}_{i} = \text{Rel}(R,SC_i);
\]

\[
\text{if} \ (\text{Re}_{i} \geq \varepsilon)
\]

\[
WS = ADD(SC_i);
\]

\[
j++;
\]

\[
\text{Return WS;}
\]

\[
\text{For (int} i=1;i<=|j-1||i++)
\]

\[
\text{if} \ (\text{Re}_{j} < \text{Re}_{i})
\]

\[
\text{Swap (SC_j, SC_{j+1});}
\]

This paper makes use of Web services clustering algorithms to classify, to reduce Web services retrieval numbers and then to carry out Web services match. The formalized language description of Web service cluster algorithm is shown as the following:

\[
\text{WS Clustering()}
\]

\[
\text{Input: WS, r;}
\]

\[
\text{Output: P;}
\]

\[
\text{S=new WebService(WS);}//Vector expression of Web services
\]

\[
\text{SC=new ServiceConcept(P);}// Vector expression of subset \ P = \text{null;}//Initialization
\]

\[
\text{For (int} k=1;k<=|WS||k++)
\]

\[
\text{If} \ (\text{SC}_k \cdot S_j \geq r)
\]

\[
\text{SC}_i = SC_i \times (j - 1) + S_j / j;
\]

\[
P_k = Add(WS_k);
\]

\[
\text{Return } P_k
\]
For (int i=1;i≤j;i++)
    {S=Convert(SC);}
WS=S;
return WS;

(3) Web service ordering

Through Web services matching algorithm, we can get the set made up of Web services sub set, which is called service result set. However, service result set’s containing Web services numbers is very large, we can use correlation between Web services is to rank Web services in Web services sub sets. First, the request vector of Web services and all Web services vector of Web services implement correlation calculation and the correlation value is ranked from high to low. The Web services ranking position will be more forward if it has larger correlation value.

IV. SYSTEMATIC MODEL OF WEB SERVICE DISCOVERY BASED ON EIGENVECTOR

A. Web Service Discovery Systematic Design

The system is set up at the UDDI [26] registration center, using UDDI registration center to be taken as storage media to store descriptive information of Web services and related information of Web services provider. This systematic kernel module is the Web services matching module to complete the match between Web services request information and Web services descriptive information.

The purpose of Web services discovery model designed in this paper is to find out the Web services satisfying Web services requesters’ requirement and filter out Web services which cannot satisfy users’ demand. According to layering patterns, Web services discovery system is divided into three layers: user interaction layer, intermediate logic layer and data layer. The structure of Web services discovery system based on eigenvector is shown as figure 2:

User interaction layer is used to receive query information of Web services requesters and Web services descriptive information of Web services providers and transmit confirmed effective information to intermediate logic layer.

Intermediate logic layer is the systematic core and it is the connecting link between interaction layer and data layer. Its functions contain performing vectorization on Web services requesting information as well as Web services description information, performing Web services match and ordering Web services query result. Besides, it also stores Web services descriptive information at the UDDI registration center to WSDL information storage base. Data layer contains UDDI registration center and WSDL information storage base, which is mainly functioned to provide Web services data information for intermediate logic layer.

B. Function Modules

(1) User interaction module

This module mainly completes Web services registration and Web services query function. The Web services discovery should make efforts to maintain the consistence between Web service provider and the information provided by Web services requesters, like the information of I, IT, O, OT, WSF, SN, SD, etc. We applied the same one page to complete Web services query and Web services registration.

(2) Information transformation module

This module mainly completes the mapping from Web services description document UDDI on WSDL. Element description in UDDI Model should correspond to definitions in WSDL and element description in UDDI business services should correspond to element Service in WSDL. <serviceName>, <input>, <output>, <textDescription> can be acquired through element description.

(3) Web Services matching module

This module is the core of the whole Web services discovery system. Its function is to complete matching process between Web services request vector and Web services descriptive vector so as to find out satisfying users’ requirement of Web services.

Web services matching are to carry out query according to Web services request information on registered Web services. During query, the match will be performed simultaneously between SN, SD and I, IT, O, OT, WSF. Web Services matching process is shown as follows:

First, Web services vectorization module is invoked to implement vectorization expression on registered Web services and Web services clustering algorithm is applied to classify Web services to generate multi-Web services sub sets and Web services sub set vector.

When one piece of Web services request information is received, this module will invoke Web services vectorization module to select QN, QI, QIT, QO, QOT, QF, QD and perform vectorization on service request information to further generate Web services request vector.

Finally, the Web services request vector and the Web services sub set vector will implement matching to get request information and correlation values of each Web
services sub set. These correlation values will be transferred to ranking modules in the form of array and Web services sub set will be transferred to ranking module in the form of array.

(4) Ordering module

Through Web services matching module, we can acquire a series of correlation values. Since Web services sub set contains a lot of Web services, this module will perform correlation value calculation on request vector of Web services and Web services in Web services sub set, the calculation results will be ranked from high to low to return to users.

B. Algorithm Implementation

Web services match implemented by this paper adopts the strategy of classification and match. First, Web services description information and Web services request information will be carried out vectorization. Then, Web services will be implemented classification according to Web services clustering algorithm and finally Web service match is completed.

Class QueryInfoVectorization is designed to complete vectorization of Web services request information. The method getValue() belongs to this class gets users' input information which includes Web services basic description information and Web services functional description information. The method inputVectorization() is applied to input information vectorization and the vectorization information is stored in vector object. Besides, the designed DataBase class acquires published Web services description information from database, applies wsVectorization() in class QueryInfoVectorization to complete vectorization of Web services description information and stores vectorization information in vector objects.

Web services clustering is the key part of service match which classifies vectorized Web services. We design class WsCluster to complete service categorization. This class takes Web services vector in class QueryInfoVectorization as input parameter, applies Cluster() method in class wsCluster to divide into different Web services sub set and calculates Web services sub set vector of each and Web services sub set.

Web services matching is to carry out match between service input information vectors in class QueryInfoVectorization and Web services sub set vectors in class WsCluster class to finally achieve correlation value, which performs ranking from high to low. Besides, Web services sub set is stored to array. This paper makes use of match() method in class QueryInfoVectorization to complete match between service input information vector and Web services sub set vector to store match results to data array.

The following displays partial key programming code in model design.

**Acquiring user's information:**

```java
getValue(DataStore d){
    name=d.getName();
    desc=d.getDesc();
    functionDesc=d.getFunctionDesc();
    inputInfo=d.getInputInfor();
    inputInfoType=d.getInputInfoType();
    outputInfo=d.getOutputInfo();
    outputInfoType=d.getOutputInfoType();
}
```

**Service description vectorization:**

```java
wsVectorization(){
    wsName=db.name*log10(db.ws)/db.wsName
    wsDesc=db.desc*log10(db.ws)/db.wsDesc;
    wsFunctionDesc=db.functionDesc*log10(db.ws)/db.wsFunctionDesc;
    ...
}
```

**Web Services clustering and Matching**

```java
Cluster(){
    double rel=0;
    int k=0;
    while (k<800) {
        for (int j=1;j<800;j++){
            for (int i=0;i<7;i++){
                rel=rel+value[k][i]*value[j][i];
            }
            BigDecimal r=new BigDecimal(rel);
            if (r.compareTo(r9)>=0){
                l1.add(j+1);
            }
        }
        k++
    }
    Match(DataBase.db,DataSeore d){
        db.Data();
        db.itemAmount();
        db.convexVector(d);
        while(j<800) {
            for (i=0;i<7;i++){
                rel=rel+db.qValue[i]*db.value[j][i];
            }
            System.out.println(rel)
            BigDecimal z=new BigDecimal(rel);
            Int r=z.compareTo(y);
            If (r>0) {
                k=j+1;
                re[m]=k;
                m++;
            }
        }
    }
}```
V. CASES AND ANALYSIS

Towards the evaluation standards for experimental results, we adopt indicators in information retrieval: precision ratio and query response time. Query response time refers to needing time from query instruction to query results and precision ratio refers to the ratio between service numbers of query results’ concentrated satisfying query requirement and the total service number of query result set.

Since there have not currently acknowledged standard platform and dataset which are used to measure Web services discovery, this paper takes Web services selected by Google search and XMethod, webxml website as test data from this experiment. The total Web services numbers in this paper are 200, 400, 600 and 800 and the query request information is input 20 times respectively under different Web services test data set. The precision ratio with different test Web services data set is shown in Table 1.

We analyzes precision ratio and query response time under the condition of setting correlation closed value T with setting value of correlation closed value T is 0.6. Query information is input in different test sets, the measured query rate as well as query response time is performed calculation and the result is compared to full-text retrieval and pure similarity retrieval. The comparison results are shown in figure 3 and 4:

From these figures, it is discovered that service discovery algorithm of Web services vector based on eigenvector has kind of advantages with higher accurate precision ratio and less query response time.

VI. CONCLUSION

The space vector model and clustering algorithm provide solutions for mentioned problems. Using vector and clustering to classify Web services can effectively improve the precision of web service. At present, syntax-based Web service discovery has two shortcomings: independent classification and low precision. In this paper, we deeply study technologies of web service discovery at home and abroad. Firstly, we analyze Web Services description language, concept lattice theory and vector-based text information retrieval technology in-depth and introduce these theories into web service discovery. Then, on the basis of current web service discovery, we propose eigenvector-based Web service discovery algorithm. The method takes the most information from Web service description as key feature items which accurately and effectively represent web service. We use Web Service clustering algorithm to classify Web services. Reducing the amount of retrieved web services, the way can improve the precision and recall. Besides, on the base of improved algorithm, we design an eigenvector-based Web service discovery system, and describe Web service library modules. In the end, the performance of the algorithm is proved by some cases.

REFERENCES


