Automated Web Service Discovery by using Probabilistic Matchmaking

Mr. Gajanan V. Bhole¹, Prof. Prakash. R. Devale²
¹,²Department of Information Technology,
Bharati Vidyapeeth University College of Engineering,
Pune-43, India

Abstract
Paper highlights the work on web service discovery model with combination of different approaches. The model will enable user to input the query for search and discover the web services based on user’s requirements. Our model uses approaches such as probabilistic matchmaking and logic based search to find the latent factors from user input or query and match with probable web service by searching in local. Otherwise model will discover the probable web service from web and it will get added into local repository for further use.

Keywords - Probabilistic Matchmaking, Logic Based Search, Web Service Discovery, Folding-In Approach, Clustering

1 Introduction

Software components mainly invoked, published and coupled over the web interface are the web services. The web services must be retrieved and managed in order to get the best available result for the queries as there is an increase in the number of web services on the web. In order to find or locate the web services that are required the organizations create new n different search problems as there are large numbers of web services [1]. There are many questions that arise regarding web services such as Retrieved web services are related to the subject or user query? How to search the required web service? Which web service is satisfying the user query? To check these web services lots of human efforts are required in order to answer or find a solution to these problems and the approach is also not feasible. Searching and invoking the web services is becoming more popular and for this external services are being used by many companies. Hence, in web based application using web services service discovery has a vital role.

Keyword-matching technologies are often adopted for the discovery of web service in order to locate or search the published web services. Due to lack of semantic description regarding the web services effectively inadequate search results are returned by the service registries. It is a difficult and challenging task to search the similarity of web services as the names of the input output parameters cannot convey completely semantics of the operation and also the descriptions of web services and their operations are not available [1]. Software components supporting machine to machine interaction
in network using industry standard protocols such as SOAP (Simple Object Access Protocol) on Hyper Text Transfer Protocol (HTTP) and interoperability are internet services. In order to exchange information between Web service Simple Object Access Protocol is used which is a light weight protocol [2]. WSDL (Web Service Description Language) describes Internet services.

In order to invoke the service so as to serve all required data or information to other developers or programmers when a Web service is published, WSDL file will be created [3]. The development tool contains certain necessary information regarding data types assigned by programmer or developer even though the description is automatically generated.

Development of registries of web services such as UDDI (Universal Description, Discovery, and Integration) has taken place in order to encourage the interoperability and adoption of web services [3]. Registration is required by UDDI registries or it contains and many obsolete entries and it is made either available publicly [4]. The registry might store only some limited description in the case of available services. Web services are published at registries of services such as UDDI (Universal Description, Discovery and Integration) and these web services are self containing, self describing and modular applications as a technology concern [5]. The link between service to the applicants and the service provider is bridged by Universal Description, Discovery and Integration (UDDI) in the service-oriented architecture.

This paper defines the proposed model for an automated web service discovery and the different approaches for web service discovery and computing with the help of probabilistic matchmaking and combination of different approaches. In the rest of this paper describes: Related work is contained in section 2, section 3 discusses the proposed system architecture in detail. Section 4 contains overview of proposed system, approaches and methods in detail. Finally, conclusion is made in section 5.

2 Related Work

For information retrieval a web service-based distributed search engine on the vector space model is created. Expanding the existing methods to make the concept of distributed environments with a technique and an overview of the basic technologies. Finally, their prototype implementation has been examined and shown that the method presented also works for the large WSDL repositories [3].

In order to get or extract the latent factors from the semantic description of web service and search the web services in latent factor space, techniques such as machine learning in particular PLSA (Probabilistic Latent Semantic Analysis) and LDA (Latent Dirichlet Allocation) are used by a non logic based matchmaking approach. This paper also defines generation of a lower dimensional vector model for representation of services and use of an probabilistic machine learning for service matching and ranking and these LDA PLSA unsupervised probabilistic machine learning methods used for the service description data[11].

2.1 Semantic approaches

In order to extend the scope of web service utilization, accessing and investigating large service repositories by useful techniques. By using semantic concepts, they specified the method for the sake of categorization by labeling the web services automatically [4].

To extract the semantic topics hidden behind the words in the query in services they use PLSA (Probabilistic Latent Semantic Analysis) so that the matching of services will be done at concept level [6].
The analysis of WSDL is done by the Semantic Web Service Classification (SWSC) is the method discussed in [7] and checks its structure and configuration for future processing.

Concept of ontology semantically defines every input and output of the web service as OWL-S standard is considered for defining the web services semantically [8].

The approach is used as an effective reduction of dimension techniques; they are able to acquire semantic relations between word-item and item-document interpreted in terms of probability distributions [9].

For modeling and item removal in information retrieval they developed and used Probabilistic models Latent Semantic Analysis (LSA) and probabilistic Latent Semantic Analysis (PLSA) [9]. They described a new approach for learning terminological ontology’s based on the “Information Theory Principle for Concept Relationship” and topic hierarchy learning algorithms with the help of these LSA and PLSA topic models.

### 2.2 Clustering approaches

Clustering approach for searching the web services is presented by Dong [1]. Search operation consists of two stages. Service search engine is given the input as text by the user, for services that are required. After that, based on the starting web services returned, given approach extracts semantic topics from the natural language descriptions provided in the web services.

Semantic Web services Clustering (SWSC) method was presented so as to extend the groups, the similar Web services and semantic representation of services for the improvement of the service discovery [7]. The effectiveness of the Semantic Web Services clustering (SWSC) is shown by three methods such as, a) use of the WSDL information for basic keyword search; b) use of the clusters which are derived from the WSDL for keyword search; c) use of WSWC method for keyword search. Empirical analysis shows the improvement in service discovery with the use of SWSC.

### 2.3 Latent factor approaches

Depending on the probabilistic machine learning techniques, they proposed a latent factor approach for service ranking to extract latent factors from the concepts or data in the service descriptions [10]. To assign a probability to every link among the latent factor and a service description this approach is located in the inherent property of probabilistic machine learning and this is advantage or versatility of this approach.

### 2.4 Ranking approaches

For the matching of web services, a rank approach was presented in [5] which uses logic based reasoning along with approximate matching. This ranks the service descriptions matching degree as per the OWL-S. In this approach of matching service is based on latent semantic indexing and logic based reasoning. Ranked results give criteria or facility to select particular service from the large set of results.

In order to help the search of the needed web services by the users in precise format new approach for semantic web services ranking is used [8]. An algorithm for ranking depending on the VSM (Vector Space Model) is presented by this approach. In this model the user query and retrieved services related to that user query will be considered as vector and based on that which service vector is very close to the user query vector will considered as higher rank. Whenever the user fire the
required query to system the rank of services are automatically calculated values of the rank are not predefined to the web services.

3 System Architecture

![System Architecture Diagram]

Figure 1: System architecture of automated web service discovery.

The Fig. 1 shows the process of automated web service discovery model. The user query from which the latent factors are get extracted for matchmaking; after that service matchmaking and search from the local repository or web according to latent factor and then required service is discovered and it will get added to the repository at a time. Then model will invoke or compute the web selected service.

4 Proposed Work

For automatic web service discovery and invoking the proposed work for automated web services discovery and computing has different approaches and methods. To get good results of web service discovery probabilistic matching, logic based search and combination of different approaches are used in the model. The model has some functional blocks or discovery models such as

4.1 Finding the latent factor from user query

To find out hidden topic or latent factor behind the service descriptor or user query. Latent Factors are used whenever a user gives service description or required query. Proposed model will use a latent factor approach. In model the latent factor in the sense the meaningful or required part from the user query.

4.2 Repository search and Retrieval of Service Information

The presence of service in the Registry is checked by the model, after completion of extracting Latent Factors from service description or user query. The user receives the description or related information if the service is present in repository otherwise service information will be search on web. after that Folding-In approach is used.

4.3 Folding-In approach
After the model is trained Folding-in method is used for fitting new web service description into the model. To search the required or needed web services, the folding-in approach is used and also to add new services into the model [11]. In proposed work our model will search the service in local registry, if not then search services on web and then add that new service to the local registry for future use, for this purpose our model will use this folding-In approach.

### 4.4 Web service discovery

To register Web services, as the central point UDDI registries are designed and also to make these services publicly available. Also it is possible to use data directly from UDDI registries [3], since the data is not required to be in XML but can be only a textual description too. It was a particularly hard task to form a satisfying repository retrieving enough WSDL Files from the Internet. our proposed model, uses probabilistic matchmaking for searching the web services according to user query and the these required services are discovered.

### 4.5 Service computing

After discovering the service and reading WSDL provide input message to web service and process output message after computing or invoking the services.

### 5 Algorithm

For efficient processing, the above modules use following algorithmic steps

**Input Query**

- **Equivalent Processing Phase**
  - Search necessary web services in open repositories or on web using TSM-LP
  - Comparing input query and Specific part of web service using TSM-LP
  - If (match)
    - Then fetch data
  - Else next specific part of web service

- Retrieved data converted to XML file

- Convert XML to Database Table

**5.1 Clustering Phase**

- Create clusters using K-Nearest Neighbor
  - For each cluster create View in Database

**5.2 Categorization Phase**

- Create Category using “Categorization on pairing” approach
  - Each service is compared to every other service in the repository once
  - Compute approximately (N*(N-1)/2) unique similarity scores
  - Unique similarity score sorted from largest to smallest
  - The pair of services with greatest similarity score makes a category for these two services.

**5.3 Filtering Phase**

- Filter data from either Clustering or Categorization OR from BOTH
  - For filtering data system use Prediction Algorithm which calculate similarity score.
5.4 Service Integration Phase
Service Integration i.e. result are displayed to user after processing the given input.
Composite Service Output

6 Implementation Results
This model uses the probabilistic matchmaking methods for searching the web service in repository or on the web also it uses clustering approach for categorization of services into the repository. Also it uses folding-In approach is used to add new services to repository. The implementation results of model are as follows.

Automatic Service Discovery Tool

Enter String: information about weather

Figure 2: User query or the required service information as input.

Fig 2 shows that the model enables user to make the required query about service.

Latent Factor: weather
WSDL address: http://www.webservicex.net/globeweather.asmx?WSDL

Invoke Message

Operates: 
- GetGlobalWeather
- GetCitiesByCountry
- GetWeather
- GetCitiesByCountry

Output:
- Data
  - System.String
    - Value
  - Enter value but what attention
  - Type parameter

Invoke

Figure 3: Extraction of wsdl file and getting the related services.

Fig 3 shows that the model will extract the wsdl file from repository or web according to the user query. And shows the services included into that wsdl.
This paper defines that the automated web service discovery model enable users to input

**6 Conclusion**
In order to gain better and fast web service discovery, the automated web service discovery model has used a logic based search, probabilistic matchmaking methods as well as combination of different approaches. The paper defines that the automated web service discovery model enable users to input
the query for the required web service and model will automatically discover the related web services from local repository, if the services are not present in the repository the model will search the related service on the web and it will automatically added to the local repository and after that it shows the result to the user. After that user will invoke web service but invocation is depend on the service provider.

7 References

Biographies

**Mr. Gajanan V. Bhole**, has completed degree in B.E. Information Technology from Shivaji University, Kolhapur in 2011 and currently pursuing the M.Tech. in Information Technology from Bharati Vidyapeeth University College of Engineering, Pune in Maharashtra (INDIA).

**Prof. Prakash. R. Devale**, Currently working as Associate professor in Information Technology Department at Bharati vidyapeeth University College of engineering, Pune.