CHAPTER - 4
PROPOSED SEMANTIC APPROACH FOR WEB SERVICE DISCOVERY

4.1 Preamble

With the advancement of the World Wide Web, Web service discovery has become an area of research. The process of service discovery is a three-step process including parsing of the user’s query, matching the description against a catalogue and finding a combination of the matched constraints that satisfy the cost constraints of the query.

In IR-based approach, there are methods such as term-frequency, inverse-document-frequency and TF-IDF to propose an answer for a given query. The problem with these methods is that they are primarily meant for keyword-based search and are unable to find semantic relationships. SVD-based methods are capable of finding the semantic associations among WS. SVD-based approaches are expensive [126] when applied to the matrix of high dimension which represents the training dataset. Thus, new methods are still required to reduce the dimensionality that will make SVD computation fast. At the same time, the reduced dimension should not affect the performance of preserving the similarity to find semantically similar terms.

In order to overcome the limitations of SVD computation, this research work proposes a new approach addressing the issue of scalability by proposing a new concept of dimensionality reduction by clustering and merging which has also been evaluated thoroughly.

4.2 Proposed Semantic Web Service Discovery Approach

This section introduces the proposed Web service discovery method using latent semantic kernel. Fig. 3.1 gives an overview of the proposed Web service discovery method using latent semantic kernel. A WSDL dataset is formed by collecting WSDL documents from Internet. A query set is defined for user to find WS that match with the query from the collection of WSDL documents and the semantically similar WS are found using the latent semantic kernel created from a general-purpose pre-processed training dataset for the defined query set and the steps followed for pre-processing namely, tokenization, compact word cleavage, stop-word removal and stemming are discussed below.

**Step 1: Tokenization**

This step uses standard lexical analyzer to tokenize the documents. The words, numbers and
punctuations are extracted from the documents. As in this work only words are considered as features, numbers and punctuations are not considered.

**Step 2: Compact word cleavage**
The names of services, functions and parameters that are always named in Pascal or Camel form contain important functional information of services. Thus, they need to be cleaved. For example, service name “RealTimeMarketData” need to be split into “real time market data”.

**Step 3: Stop Word Removal**
In this step, frequently occurring words such as where, when, is, and, etc., are removed as they convey no (or little) semantic information. Also, words occurring very often in most of the documents are removed as they have very little power with respect to that document.

**Step 4: Stemming**
The number of distinct terms by conflating words (for e.g. goes, going, gone) are reduced into a single word (here it is ‘go’) using Porter Stemmer algorithm [127, 128] which is a widely used algorithm for stemming the terms.

**Semantic Analysis**
The methodology used for Web service discovery using the latent semantic kernel is shown in Fig. 4.1. The approach proposed in this research work is motivated by the content of the WSDL document which provides information about the WS. For a given query, a set of samples of WS from the WSDL dataset which is considered as the working dataset is retrieved using the proposed approach. This dataset is partitioned into a set of smaller clusters by using bisecting k-means method which reduces the number of retrieved Web services. This step uses only the syntactical association between the query and WSDL documents. Next, the WSDL documents of the working data set are mined to extract the same five features, namely, service name, operation, message and port types which describe the semantic and actions performed by the WS by describing and exposing the functionality of the WS. These features are integrated for clustering into functionally similar groups. This is the initial step to help a service search engine to identify the Web service functionality and match users’ requests to WS.

The selection of training dataset plays a vital role in building the semantic kernel and the result will not be satisfactory if the dataset doesn’t include the WSDL topics. The Wikipedia
dataset released by Denoyer and Gallinari [129] which represents the world of knowledge is selected to construct the semantic kernel and this dataset selection guarantees that the WSDL data topics are largely included. Also the result is not domain-specific kernel and the kernel can be used for all WSDL documents.

A general-purpose dataset mentioned above is used to build the semantic kernel which can represent different domain. The semantic-kernel to find different topics is build using the term-document matrix which represents all training documents. Because a large number of terms and training documents are present in the dataset, the kernel construction is posing a problem for the large dimensionality of matrix. Normally the algorithms for matrix factorization are computationally expensive introducing scalability bounds [126]. A new for approach for dimensionality reduction of the term-document matrix is proposed in this work by clustering and merging of the documents in the working dataset which creates the semantic kernel.
The proposed method does match-making to discover WS which are semantically similar for a given user query against semantic kernels. The Web service discovery method using latent semantic kernels helps to find WS which are similar semantically. For example, if a user looking for WS associated to ‘weather’, sometimes be interested in WS associated to ‘rainfall’ or ‘climate’ since they are related semantically.

The constructed kernel consists of WS represents diverse domains which are semantically similar. The semantic kernel which is used to find different topics is constructed using the term-document matrix which represents the documents in the working dataset. The formed semantic kernel supports to discover these unnoticed topics and their associations to the term and document set. Fig. 4.2 gives the overview of the semantic analysis method used in this method suffers from low speed, as the number of comparisons gets larger, when the number of items in the dataset increases discovery. The proposed support-based semantic kernel is used to find the similarity between the user query and WS. After extracting the content of the WSDL documents, the pre-processing steps which have been discussed earlier are performed to create the semantic kernel. Then the importance of documents is calculated for the documents in the dataset. Based on the document importance value, the clusters consisting of WSDL documents are formed such that each cluster holds equal number of important documents. Then the documents in every cluster are merged forming a single document which represents the content of that cluster. Thus the matrix size is reduced by keeping the number of terms intact as of original dataset. Thus this reduced size matrix is used to create latent semantic kernel representing a diverse range of domains by matrix factorization method such as SVD.

The similarity between WSDL documents and a query is found using the created support-based semantic kernel is used and the Top-K WSDL documents which are ranked using the similarity value calculated using the proposed support based semantic kernel are returned as recommendations to the user. The methodologies to create the support-based semantic kernel along are discussed below with essential background information.

**Proposed Semantic Kernel Creation Method**

As term-document matrix is large in dimensionality because of more number of terms and documents in the training dataset, the time for matrix computations and required amount of space are high. For a $m \times n$ matrix $A$, the time complexity for singular vector
decomposition is $O(mn^2 + m^2n + n^3)$ [126]. This research work presents a new approach of minimizing the dimensions of the term-document matrix by clustering and merging the documents. The dimensionality of matrix is reduced in the proposed method without any information loss as terms which are pre-processed are used in the construction of semantic kernel. There are two different clustering and merging methods which have been used to reduce the matrix size, one is selecting documents randomly and the other selecting documents by proposed support-based method. Equal number of documents are selected randomly in the random document selection method and placed into specified number of clusters. In this work, bisecting k-means algorithm is used because it can produce consistent clusters with relatively uniform sizes (in this case, every sub-cluster might include an equal number of WSDL documents) and the computation on partitioning a data set is simplified [130]. The partitioning processing continues until the desired number of clusters is acquired.

A systematic approach is followed in the support-based selection by selecting documents such that equally important documents are placed in each cluster. The aim of the support-based selection method is to eliminate any partiality caused by the random selection
method. Consider the working dataset as $TD = \{TD_1, TD_2 \ldots, TD_m\}$. It is assumed that a document $TD_i$ contains $n$ unique terms $(ut_{i1}, ut_{i2}, \ldots, ut_{in})$ after pre-processing with the frequencies $(utf_{i1}, utf_{i2},\ldots, utf_{in})$. Assuming $UT_j$ as a unique term in the $m$ dataset, the total frequency of $TF_j$ is denoted by $F_j = \sum_{i=1}^{m} utf_{ij}$.

For merging several documents, “document importance” concept is introduced in order to reduce the dimensions of term-document matrix. This document importance value is used to weight the documents and the documents are equally distributed across several clusters. Then a single document is formed after merging the documents in each cluster. This method lessens the dimensions of the term-document matrix significantly subject to the requisite number of clusters. In order to compute the document importance value, two measures are defined such as weightage and support of a term. Support of a term specifies the importance of the term in the whole dataset relatively and weightage of a term indicates the importance of the term in the document. The bias of a frequent term present only in a document is removed while using support and weightage in the calculation of the document importance.

The weightage $TW_j$, of a term $ut_j$ present in a document indicates the relative importance of that term in the document. The ratio of the frequency of the term in the document to the frequency of all terms present in that document is calculated using the Eqn. 4.1 which is given below.

$$TW_j = \frac{utf_{ij}}{\sum_{i=1}^{n} utf_{ij}}$$

(4.1)

The sum of the frequencies of all terms present in the dataset is given by $\sum_{j=1}^{n} \sum_{j=1}^{m} utf_{ij}$.

$S = \{SS_1, SS_2 \ldots SS_n\}$ is assumed to be the set having the support of $n$ unique terms present in the working dataset. The relative importance of the term present in the dataset is shown as support $SS_j$ of a term $UT_j$ in a dataset. That is, $SS_j$ of term $UT_j$ is the ratio of the total frequency of $TF_j$ in the corpus to the size of the dataset which is computed using the Eqn. 4.2.

$$SS_j = \frac{F_j}{F} = \frac{\sum_{i=1}^{m} utf_{ij}}{\sum_{j=1}^{n} \sum_{j=1}^{m} utf_{ij}}$$

(4.2)

Eqn. 4.3 gives the document importance ($DI_j$) using the weightage and support of all terms.
Then the documents are selectively chosen and placed into different clusters. All documents in a cluster are then merged to form a single document reducing the term-document matrix dimensions without having loss of terms. Finally, there is a reduction in the number of documents without varying the number of terms.

C is considered as the group of ‘q’ clusters, \( C = \{C_1, C_2, \ldots, C_q\} \) containing equal number of documents in each cluster, dividing the dataset TD into q clusters.

MD is assumed as a group of ‘q’ documents, \( MD = \{MD_1, MD_2, \ldots, MD_q\} \) where \( MD_i \) is a document after merging all the documents from cluster \( C_i \). The frequency of the terms in the merged document is computed. In this work, terms have frequency less than 5 and terms having frequency greater than 10000 are considered as outliers and they are removed.

RK which is considered as the modified term-document matrix is used for constructing the latent semantic kernel. In the matrix RK, unique terms are represented as rows, the merged document are represented as column and each cell has the support value (Eqn. 4.2) of a term in a merged document. The matrix factorization method, SVD is applied on the matrix RK to create the semantic kernel. The output of SVD on matrix RK is the semantic kernel represented as \( U_k \) with a suitable value of \( k \).

Using the proposed approach of clustering and merging, two algorithms are used in this work for constructing the semantic kernel. Fig. 4.3 presents the algorithm for selecting the WSDL documents using randomness. The inputs for the algorithms are training dataset (TD), total number of documents (m), total number of clusters (q) and term-document matrix (K) and the output is the latent semantic kernel (SK).

The algorithm for support-based document selection is presented in Fig. 4.4. The training dataset (TD), total number of documents (m), total number of clusters (q) and term-document matrix (K) are used as the inputs for the algorithm and the output of the algorithm is the latent semantic kernel (SK).

\[
DI_i = \sum_{t=1}^{n} TW_j * SS_j
\]  

1. Perform pre-processing on documents in TD

/* Clustering Documents */

2. Form q clusters by using bisecting k-means algorithm
/* Documents Unification*/
3     for every $C_q$ in $C$
4         Assign $MD_q = \emptyset$
5     for every $D_i$ in $C_q$
6         Compute $MD_q = MD_q \cup D_i$
7     end for
8 end for

/* Initialization of the term-document matrix RK */
9     for every $x = 0$ to $T_j$
10        for every $y = 0$ to $D_i$
11           Assign $RK[x][y] = 0$
12        end for
13    end for
14    for every $MD_q$ in $MD$
15        for every $T_j$ in $MD_q$
16           Compute $RK[T_j][MD_q] = RK[T_j][MD_q] + S_j$
17        end for
18    end for
19    Perform SVD on RK
20    Assign Latent semantic Kernel $SK = U_k$

Figure 4.3: Algorithm for random selection method

1 Perform pre-processing on every document in $TD$
2 Weightage ($W_j$)and support ($S_j$) are calculated for terms $T_j$ using Eqns. 4.1 and 4.2 respectively.

/* Calculate the Document Importance*/
3     for every $TD$ in $D$
4         Compute $DI_i = \sum_{i=1}^{n} W_j \times S_j$
5     end for
6     Sort $DI$ in ascending order

/* Clustering Documents */
for ii = 0 to q
  Assign $C_{ii} = \phi$
  for jj = 0 to m/q
    Compute $C_{ii} = C_{ii} \cup TD_{jj+q+ii}$
  end for
end for

/* Merging Documents */
for each $C_q$ in C
  Assign $MD_q = \phi$
  for each $D_i \in C_q$
    Assign $MD_q = MD_q \cup D_i$
  end for
end for

/*Initializing the term-document matrix RK*/
for every $x = 0$ to $T_j$
  for every $y = 0$ to $D_i$
    Assign $RK[x][y] = 0$
  end for
end for
for each $MD_q$ in $MD$
  for each $T_j$ in $MD_q$
    Compute $RK[T_j][MD_q] = K[T_j][MD_q] + S_j$
  end for
end for
Perform SVD on RK
Assign Latent Semantic Kernel $SK = U_k$

Figure 4.4: Algorithm for support-based method

Computing similarities between query and WSDL documents
The similarity between a user query $Q$ and a WSDL document $D$ is computed according to the VSM which finds the cosine of the inner product between their document vectors and the equation is given as:
\[ \text{Sim}(Q,D) = \frac{\sum Q_i D_i}{\sqrt{\sum Q_i^2 \sum D_i^2}} \]  

(4.4)

where \( Q_i \) and \( D_i \) are the weights with respect to the two vector representations to recommend the relevant service documents to a user if the pre-defined threshold value is exceeded by the score based on the similarity value. For a given query, ranking of documents is based on the values similar to the query. This model is the well-known bag of words model which is widely used for document retrieval.

A WSDL document consists of Operation, Port Type, Types, Message, Binding, Port and Service tags and thus to calculate the similarity value processing of various components is required. The contents of service name (\( P' \)), operation (\( R' \)), message (\( S' \)) and port types (\( T' \)) are extracted from the WSDL document. Then the contents which are extracted are pre-processed. Using Eqn. 4.4, the similarity value between the query and various selected features of the WS is computed. Finally, the total similarity (\( TS_{sum} \)) between the query and the WS is computed using Eqn. 4.5.

\[ TS_{sum} = w_1 \times \text{sim}(Q, P') \times w_2 \times \text{sim}(Q, R') \times w_3 \times \text{sim}(Q, S') \times w_4 \times \text{sim}(Q, T') \]  

(4.5)

where \( w_1, w_2, w_3 \) and \( w_4 \) are assigned equal values such that \( w_1 + w_2 + w_3 + w_4 = 1 \).

4.3 Experiments and Evaluation

Experiments are conducted on two datasets to assess the proposed approach for Web service discovery. The proposed algorithms are implemented using java under Windows platform. The first dataset used for semantic kernel construction is a Wikipedia dataset released by [129]. The kernel constructed using the Wikipedia dataset represents the knowledge of various domains that a Web service may belong to. This dataset consists of more than 48,000 documents which is too large for a SVD-based method to handle on a PC of Pentium Dual Core 1.86 GHz processor with 1GB of main primary. The proposed algorithms given in Figs. 4.3 and 4.4 are used to reduce the size of the original term-document matrix representing the dataset. The kernel constructed from this dataset is used to build the semantic kernel and this kernel is used to find the similarity between a query and the WSDL documents.

The second dataset is a group of WSDL documents which is used to evaluate the proposed methodology. This second dataset contains 800 WSDL documents collected from
XMethods.com, SOATrader.com and the QWS Dataset [102]. The WSDL documents in the dataset represent a variety of WS such as news, stock market, music, literature, weather information, sports etc.

In order to evaluate the proposed approach, various experiments have been performed. After carefully examining the domain and content of the WS (WSDL documents), 25 queries were created (listed in Appendix – A) and used for each of these experiments. The topics covered by the queries are finance, sports, barcode, email verification etc. Various topics have been considered to create the query since in real life the queries are related to any topic. Few examples of queries are weather by postcode, residence information by phone number, music audio song, currency conversion, image conversion etc. Thus, queries as obvious are created to estimate the performance of the semantic kernel. A number of 10 queries have been considered and the outcomes are averaged to present a balanced evaluation of the proposed approach. The query set is balanced because it was created by analyzing the WS (WSDL documents). Two terms are there in each query and the average query size is two. Once the query set is finalized, the next task is to set the evaluation criteria.

### 4.3.1 Evaluation Criteria

The accuracy of the proposed Web service discovery methods is measured using Precision, Recall and F-score values. Precision is defined as the capacity to offer the related WS from a set of retrieved WS and it is computed using Eqn. 4.7. Recall is defined as the capacity to offer the maximum number of related WS from a set of relevant WS and it is computed using Eqn. 4.8 and Eqn. 4.9 is used to calculate the F-score value which is defined as harmonic means of precision and recall values.

\[
\text{Precision} = \frac{\text{Number of relevant Web services retrieved}}{\text{Total number of retrieved Web services from the dataset}} \tag{4.7}
\]

\[
\text{Recall} = \frac{\text{Number of relevant Web services retrieved}}{\text{Total number of relevant Web services in the dataset}} \tag{4.8}
\]

\[
F - \text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.9}
\]

An average of 10 queries is presented as the result of these measures. Top-N precision has been
considered for evaluating the approaches since the most users are interested in the first $n$ WS and 5, 10 and 20 are assigned to $n$ in all experiments.

**4.3.2 Experiment Design**

This section explains about the design of the experiments performed for analyzing the proposed method and compared with other standard methods. In order to assess the performance of the WS discovery method using the proposed support-based semantic kernel, several experiments are performed. Two types of kernels are built using term-document matrix based on the merged documents using support-based and the random distribution. The performance of built semantic kernels is compared with the semantic kernels built without merging and using the support-based selection and the random projection method. In random projection method, the training dataset is projected to $m$ dimensions where $m > k$ and $k$ is the dimension of the semantic kernel, in order to obtain a smaller representation of the original dataset and then SVD is performed on the reduced dimension matrix. For this experiment, a collection of 1500 documents are selected from the Wikipedia dataset based on their document importance to perform random projection. The number of documents selected is based on the maximum number of documents a machine with 1 GB of memory could handle.

Two measures such as the standard TF-IDF values (Salton & Buckley [131]) and the proposed support measure as given in Eqn. 4.3 are used in different experiments, in order to represent the matrix to construct the semantic kernels.

The proposed work is compared with the approach proposed by Jiangang Ma et al. [37] which is considered as baseline method. Experiments are performed to evaluate the proposed clustering and merging approach using document importance measure with respect to the efficiency of reduction in dimensions. Support based approach is used in order to construct the compressed kernel using the term-document matrix built on the merged documents. Also the semantic kernels built using the proposed merging approach is compared with the semantic kernels which are constructed without merging with respect to performance.

**4.3.3 Results Evaluation**

This section provides the evaluation results of the experiments that are performed based on the experiment design given in the preceding section.
Table 4.2 includes results of the WS discovery terms of Precision, Recall and F-score. The values are the average results of 10 queries. This table has the outcomes of the proposed methods for reducing the size of the matrix in constructing semantic kernels and the performance of the proposed approach in creating the semantic kernel using the term-document matrix representing TF-IDF measure is better than the kernel built using the support-based method. This is due to the fact that certain words in the dataset are present sparingly and low support value is obtained from Eqn. 4.2.

The sample queries used for experimentation are Barcode Reader, Holiday Date, Phone Number, Company Information and Stock Management. These five queries are selected because semantically similar WS are present in the WSDL dataset related to these queries. This experiment is executed in order to assess the performance of various methods in finding WS which are semantically similar.

Results of Table 4.1 and Fig. 4.5 indicate that the accuracy of the semantic kernel constructed with support-based selection method is better than the kernel constructed with random selection method. The important issue is the selection of the lower-size dimension \(k\) with which the semantic kernel is created. As the WS include various domains, the constructed kernel should be able to find the semantics differences in all WSDL documents. Various experiments using Wikipedia dataset is performed to calculate the trade-off between computation cost (response time, disk space) and the accuracy (recall, precision, F-score).

Table 4.2 shows the effect on the disk space and average response time for various values of \(k\) for creating latent semantic kernel. The presented values in the Table 4.2 are the average values of 10 queries. From the values shown in Table 4.2, it is clear that a value of \(k=350\) has a better value for F-Score as well as average response time and required disk space. Though \(k=750\) has a better F-Score value, it is costlier in terms of time and space. Hence, in this work, \(k=350\) is preferred for creating the semantic kernel.

For any training documents, the goal of the proposed approach is to construct an effective and scalable semantic kernel. Also semantic kernel is formed without merging documents in the cluster. The performance of these semantic kernels is matched with the semantic kernels with the matrix reduced in size by merging to evaluate their effect. Table 4.3 matches the various types of kernels that are created without merging the documents.
Table 4.1: Comparison of F-Score values of various methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>Types (Merging)</th>
<th>Sub types (Uk = 350)</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Top 20</td>
</tr>
<tr>
<td>Proposed Latent Semantic Kernel</td>
<td>Support-based Selection</td>
<td>TF-IDF</td>
<td>50.74</td>
</tr>
<tr>
<td></td>
<td>Random Selection</td>
<td>TF-IDF</td>
<td>49.22</td>
</tr>
<tr>
<td></td>
<td>Support-based Selection</td>
<td>Support-based</td>
<td>48.33</td>
</tr>
<tr>
<td></td>
<td>Support-based Selection</td>
<td>TF-IDF</td>
<td>47.83</td>
</tr>
<tr>
<td>Jiangang Ma et al. [37]</td>
<td>TF-IDF</td>
<td></td>
<td>36.66</td>
</tr>
</tbody>
</table>

Figure 4.5: Comparison of F-Score values for a set of queries

Table 4.2: Average response time and disk space with F-Score for different values of Uk

<table>
<thead>
<tr>
<th>Semantic Kernel (Uk)</th>
<th>Precision Top 10</th>
<th>Recall Top 10</th>
<th>F-Score Top 10</th>
<th>Average response time in seconds</th>
<th>Disk space in MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>39.3</td>
<td>68.50</td>
<td>49.95</td>
<td>79</td>
<td>65.5</td>
</tr>
<tr>
<td>350</td>
<td>40.7</td>
<td>73.45</td>
<td>52.38</td>
<td>96</td>
<td>96</td>
</tr>
<tr>
<td>450</td>
<td>39.7</td>
<td>71.65</td>
<td>51.09</td>
<td>116</td>
<td>135</td>
</tr>
<tr>
<td>550</td>
<td>40.4</td>
<td>71.11</td>
<td>51.53</td>
<td>136</td>
<td>165</td>
</tr>
<tr>
<td>650</td>
<td>40.4</td>
<td>71.25</td>
<td>51.56</td>
<td>156</td>
<td>190</td>
</tr>
<tr>
<td>750</td>
<td>40.9</td>
<td>73.68</td>
<td>52.60</td>
<td>177</td>
<td>224</td>
</tr>
</tbody>
</table>
Table 4.3: Comparison of F-Score measures of the proposed kernel creation methods without merging

<table>
<thead>
<tr>
<th>Methods</th>
<th>Types (without merging)</th>
<th>Sub types ($U_k = 350$)</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Top 20</td>
<td>Top 10</td>
</tr>
<tr>
<td>Proposed Latent Semantic Kernel</td>
<td>Support-based Selection</td>
<td>TF-IDF</td>
<td>33.27</td>
</tr>
<tr>
<td></td>
<td>Random Selection</td>
<td>TF-IDF</td>
<td>30.22</td>
</tr>
</tbody>
</table>

The proposed support-based selection achieves improved result than the random selection method because of the randomness in selecting the documents is eliminated. From the Fig. 4.6, it is obvious that proposed support-based selection method is capable of creating more number of clusters and has an equal number of important documents and hence supports removal of randomness introduced by random document selection.

Fig. 4.6: Number of clusters based on the document importance value

Fig. 4.7 proves that the proposed latent semantic approach based on TF-IDF method shows a better F-Score value compared to the approach proposed by Jiangang Ma et al. [37]. Based on the experiments conducted, it is obvious that the recommended semantic approach in the Web service discovery using the support-based approach allows in finding more relevant services compared to random selection method. Also the scalability of the proposed approach is efficient. Thus the proposed approach offers a two-fold benefit in terms of accuracy and efficiency.
4.4 Chapter Summary

The proposed semantic approach based on a new notion of reducing the dimensions by clustering and merging has been assessed and compared with other different methods. Based on the experiments, the proposed support-based clustering and merging WS discovery method has an improved F-Score value compared to the random selection method. This is because of the randomness brought by the selection of documents arbitrarily. Also, it is apparent that the proposed support-based selection method is able to create clusters having equally important documents helping in removing the randomness introduced by selecting documents randomly.

As several functionally similar WS exist in the network, selection of suitable service from equivalent service set is done using QoS values. The quality of WS should be wisely considered in order to confirm the reliability of WS during selection. Since the service invoker lacks adequate historical information for some specific Web service as the service user has not invoked the service, the assessment of QoS values of those services from other similar users or the user’s record on invocations on other WS is required. Next chapter proposes a new approach for QoS forecasting for a better recommendation of WS to the service user.