Cross-Language Semantic Web Service Discovery to Improve the Selection Mechanism by using Data Mining Techniques

Ahmed Sharaf Eldin, Ph.D
Information Systems Department
Faculty of Computers and Information
Helwan University

Ayman E. Khedr, Ph.D.
Information Systems Department
Faculty of Computers and Information
Helwan University

Fahad Kamal Al-Sharif
Information Systems Department
Faculty of Computers and Information
Helwan University

ABSTRACT
Web service became one of the important methods for communication through the internet and its usage increased in the levels of users and developers. Semantic web service represents the second generation of web services that contains more description and information about its contents. Searching and dealing with web service is done through process called web service discovery which returns a Semantic Web Service Description Language file (SWSDL) for each web service. This research aims to expand the semantic web service usage through adding the multilanguage capability to the web service's discovery process and through recommending other web services to the user based on his history in using web services. These aims were achieved by modifying the web service discovery model through adding two important techniques the Cross Language Information Retrieval (CLIR) technique and the data mining association rules technique. This research proposed two sub models, the first sub model proposed the application of CLIR techniques and information retrieval method to support Bilingual Web service discovery process the second language that proposed here is Arabic.

Text mining techniques were applied on SWSDL content and user's query to be ready for CLIR methods, this sub model was tested on a curated catalogue of Life Science Web Services http://www.biocatalogue.org/ and achieving 99.38 % accuracy and 87.23 precision of the effectiveness of the monolingual system.

The second sub model proposed a process of web service recommendation by applying the data mining techniques to suggest another web service beside the one he got from the discovery process based on the user's history. This sub model was tested on the mention curated web services site and the results were 65 % of users chose services from the services that recommended by the proposed sub model.

Keywords
Semantic web services discovery; Cross Language Information Retrieval; Text Mining, BioCatalogue, Data Mining, and Recommendation System.

1. INTRODUCTION
Web service discovery aims at finding services whose description matches that of a desired service. The description of service contains a functional and a non-functional part. The former provides information about what the service does and how it works. This is expressed in terms of the required inputs and generated outputs, as well as any pre-conditions that need to be satisfied in order for the service to be executed and any effects that result from its execution [1].

The discovery process is done through applying information retrieval techniques on the SWSDL content of the web service. Large number of studies used various Information Retrieval (IR) techniques to search textual service metadata for service discovery. Wang and Stroulia in [2] employed the inverted file to index and search natural language description of desired services. Platzer and Dustdar in [3] also used the Vector Space Model (VSM) to implement a search engine for a Web service, and Sajjanhar et al. have attempted to leverage Latent Semantic Analysis (LSA), a variant of VSM, for Web services discovery [4].

The earlier related work index service metadata found either in a SWSDL file (i.e., the <documentation/> tag) or from a Universal, Description, Discovery, and Integration (UDDI) registry entry. In both cases, service providers manually create service metadata written in English [5].

When user wants to search for a web service for a specific purpose, the user should write the query in English because the service metadata is written in English language. The motivation for addressing this idea came from our experience in developing the web service and studying the Cross-Language Information Retrieval (CLIR) methods and data mining techniques. This research dealt with the web services as collections of documents that should be prepared for IR techniques, then applied CLIR techniques to find the suitable service that matches the user query that written in other language. The process that is responsible for finding the suitable service is matchmaking process. It is the process of finding suitable services given by the providers for the service requests of consumers. The current service discovery mechanism of WSs is based on SWSDL and UDDI. SWSDL is an XML based language to describe properties of services that written in English language. UDDI is a registry where service providers can advertise their services and service consumers can search for services. The specific objective of our research is therefore to apply CLIR in the web services discovery; this is done by modifying the Match Maker process by adding CLIR components to support the Cross language web service discovery [6] .

Thinking about the web service as a commercial item raise the idea of using the recommender systems that used by E-commerce sites .These systems are used to suggest products to their customers. The products recommendation process can be done through analyzing several shared properties between customers like nationality, site, demographics, customer's behavior and buying history. Through analyzing these properties the customer future buying behavior can be predicted, and by the applying the same concept, when user wants to search for a web service, users have to write the query then the service discovery agent find the suitable

service. The research model works after the discovery model finishing his process by finding the required web service and the user accepts this service then our model works as a service recommender based on the analysis of users history [7].

Applying these two approaches lead to several benefits: (1) Enable different stakeholders to search for web services using their natural language, especially in the new operating systems like Android, Windows 8 metro applications and Apple IOS; the developers of applications that working on these platforms are interacting with web services to expand the application capabilities like connecting to database server, or generating reports from multiple database resources. (2) Searching and finding such a service to perform the developer required function are the heart of our research, and as mentioned above the search process was being performing by English language only, but by applying the research proposed model, the research process expanding to let the service searchers to find their required service via any language not only with English language.(3) Expand the usage of web services that provided by the web services provider, and it gives users more information about the most used services that used by other users who used the same service [6][7].

The rest of this paper is organized as follows. Section 2 provides the technical background that is used in the rest of the paper. Section 3 provides the related works that used in our research. Section 4 defines our proposed model and the modifications that had been done on the original one. Section 5 studies our methods in a case study. Finally, Section 6 contains the conclusion and future work.

2. BACKGROUND AND MOTIVATION

This section provides a succinct background of the used techniques in our research. These techniques namely are: 1) Cross Language Information Retrieval, 2) Inverted file indexes, 3) Bilingual dictionary, 4) Association rule, 5) BioCatalogue a curated web service registry.

2.1 CLIR

Cross-language information retrieval (CLIR) systems allow users to find documents written in different languages from that of their query. The goal of a CLIR system is to help the searchers to find documents that are written in languages that are different from the language in which their query is expressed. This can be done by constructing a mapping between the query and document languages, or by mapping both the query and document representations into some third feature space. The first approach is often referred to as ‘‘query translation’’ if done at query time and as ‘‘document translation’’ if done at indexing time, but in practice both approaches require that document-language evidence be used to compute query language term weights that can then be combined as if the documents had been written in the query language [8].

In all cases, a key element is the mechanism to map between languages [8]. This translation knowledge can be encoded in different forms as a data structure of query and document-language term correspondences in a machine-readable dictionary or as an algorithm, such as a machine translation or machine transliteration system. Gina-Anne Levow in [8] proposed CLIR reference architecture that focuses on the query translation architecture that illustrates the full range of opportunities to improve retrieval effectiveness [8].

Fig. 1 presents the Gina's CLIR model that illustrates the data flow between the key components in her reference architecture [8].

The dictionary based query translation architecture consists of two streams of processing for the query and documents. Moreover, she exploits methods for suitable term extraction and pseudo-relevance feedback expansion at three main points in the retrieval architecture: before document indexing and query translation, as well as after query translation [8].

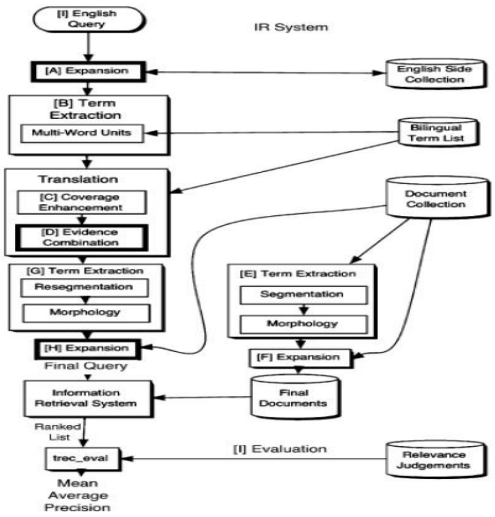

Fig 1 CLIR architecture [8]

2.2 Inverted File Indexes

Inverted file is widely used for indexing text database. To support efficient information retrieval, the words of interest in the text are sorted alphabetically. For each word, the inverted file records a list of identifiers of the documents containing that particular term. Consider a sample text database consists of five documents. The indexer parses these five documents, and produces a set of distinct words for constructing the inverted file. The inverted file has two components a vocabulary and a set of inverted lists. The vocabulary comprises a collection of distinct words extracted from the text database. For each word t, the vocabulary also records: (1) the number (ft) of documents that contain t, and (2) the pointer to the corresponding inverted list [9].

Each one of the word-specific inverted lists records: (1) a sequence of documents that contain t (notice that each document is represented as a document number d), and (2) for each document d, the frequency (fd, t) of t appearing in d [10].

Thus, the inverted list is a list of < d, fd,t > pairs. Applying the inverted file in WSDL discovery process will be done through dealing with WSDL files as a documents that containing the terms that needed to calculate the (fd,t) pairs the frequency (fd,t) of t appearing in WSDL document [9].

The SWDSL document is an XML file that contains tags for each information about the service, the name of this tag describes the type of information that exists inside it.

The inverted file index process is done through two main steps [6]:

- SWSDL term extraction: SWSDL term extractor is responsible for extracting the textual content directly from the interface definition to support information retrieval tasks, and the extracted information is indexed in of < d, fd,t > pairs.
- SWSDL term tokenization: tokenize the extracted terms into meaningful terms for example: tokenize the term

"GetLastTradePrice" into four separate terms "" to Get, Last, Trade, and Price.

2.3 Bilingual Dictionary

Bilingual term lists are extensively used as a resource for dictionary-based Cross-Language Information Retrieval (CLIR), in which the goal is to find documents which written in one natural language based on queries that are expressed in another (Nagata, 2001). A bilingual dictionary, known as a translation dictionary, is a "specialized dictionary used to translate words or phrases from one language to another" (Nagata, 2001) . Bidirectional bilingual dictionaries usually consist of two sections, each listing words and phrases of one language alphabetically line with their translation. Moreover, a bilingual dictionary indicates the part of speech, gender, verb type, declension model, and other grammatical clues helping a non-native speaker use the word [10].

In one hand, Bilingual dictionaries are the lists of phrases, usage and style guides, verb tables, maps, and grammar references. On the other hand, the bilingual dictionary, a monolingual dictionary defines words and phrases instead of only translating them [10].

2.4 Association Rule

Association rule is one of data mining techniques that used to find the associations between several classes or clusters of items. In addition, it named by market basket analysis or Affinity analysis. Applying this technique leads to find the association between different groups or items such as the relations between students and their courses; for example students taking CC482 often also take CC481 and CC483 [11]. Data mining for association rules consist of a two-stage process:

- Finding the completely frequent itemsets.
- Generating strong rules from frequent item sets.

For the first stage, it uses Apriori algorithm, it is an algorithm for finding patterns in data based on the observation. For example in the supermarket if the sales of two items like milk and rice were analyzed and then it was found that most of customers that buying milk also buying rice, that means there is a strong relation between the sales of these two items. This fact could be used as a hint to the manager of the super market to put the milk and rice in same corner to increase the sales of the two items [12].

The second stage is used for the whole generated pairs of frequent itemsets the union of each pair is also frequent the confidence of the rule between pairs [11]:

A, B: ***c = support (A U B) / support (A).***

2.5 BioCatalogue A Curated Web Service Registry

BioCatalogue is a web service registry dedicated to the life science community. BioCatalogue provides means, by which a Bioinformatician, for instance, can subscribe his /her service within the catalogue, e.g., by uploading the web service description file (WSDL) [6][7].The Bioinformatician can then add more text to describe the web service and map the web service elements to terms in the myGrid service and domain ontology. Annotation dates and histories can also be part of the markup . The main functionalities that will be supported by BioCatalogue [6] [7]:

- Integrated access to life science web services
- Rich description of web services
- Curation of service descriptions
- Web service discovery
- Interoperability

3. RELATED WORK

This section provides the necessary background of the related work that the research be built on it.

3.1 The IR-style Web Services Discovery

Recent studies have been reported in applying IR for Web services discovery. Chen Wu [11] proposed IR-style web service discovery that uses inverted file indexes for Web services discovery that will be merged in CLIR methods in the proposed approach.

Chen Wu [11] proposed IR-style Web services discovery approach that was illustrated in fig. 2, in which term tokenization constitutes an important step for WSDL term processing. Initially, service providers deploy their Web services accessible to the public via the Web. In doing so, they also publish a service description, i.e., the WSDL documents, which captures the functional capabilities and technical details (e.g., transport bindings) of a Web service these service descriptions could be collected by a number of service Crawlers, which fetch WSDL files from the Internet [11].

Alternatively, these files could be collected from some well-known service datasets such as one of the WS curated catalogue .Crawlers hand over retrieved WSDL files and associated HTML files to the WSDL preprocessor for link analysis. This yields a list of new URLs that may point to some new WSDL files. The URL server such as Pica-Pica Web service description Crawler [11] assigns these URLs to an idle crawler.

All retrieved WSDL files are then passed to the WSDL term processor, which (1) parses WSDL files and extracts important data (e.g., operations, messages, data types, etc.), (2) tokenizes extracted content into separate terms (the focus of this paper), and (3) carries out other linguistic tasks such as lemmatization and stop-word elimination, etc. The five-tokenization methods (two baselines, Kokash and MMA, and three statistical methods MDL, TP, and PPM) are used in step two [11].

The WSDL processor generates the 'term document', which contains separated words in a flat structure. The term document is transferred to the inverted file indexer. The indexer takes as inputs tokenized and lemmatized terms with their associated occurrences information in each document and generates as outputs the compiled data arrangement with pre-aggregated information optimized for fast searching. The data structure of inverted index is consistent with the notion of term-document matrix, which consists of term vectors as matrix rows and document vectors as matrix columns. The term vector is sorted that allows fast lookup operation. After finishing the document tokenization and indexing processes, the search handler component will extract the key terms from the query then it will search via extracted terms in the inverted file index , after that the most matched WSDL documents is returned based on terms frequency in the WSDL documents [11].

Meanwhile, Chen's model is based on the document retrieval. Where the query and the documents are written in the same language, but if the query was written in different language it will not retrieve the documents or the services, for this limitation the need for applying the CLIR techniques was arising to support searching in documents with queries with different languages, so we modified Chen's model to accept queries with different languages "Arabic in our model" and retrieve the suitable service and WSDL document with a translated WSDL version to query writer's language [11].

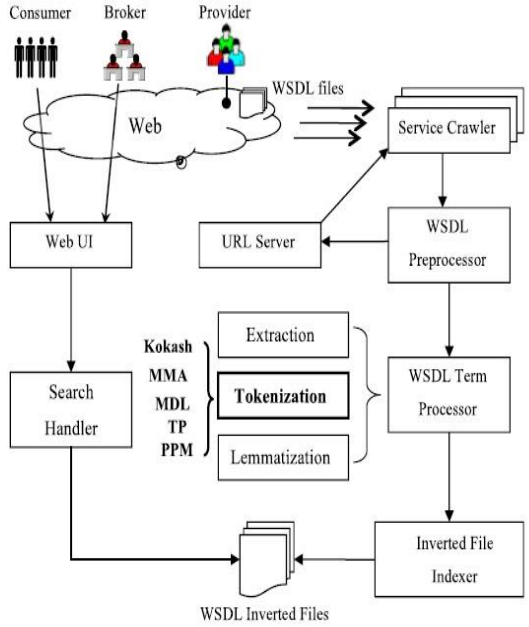

Fig 2 IR-style service discovery approach [11]

3.2 Arabic Treebank (ATB) segmentation

The Arabic language has a very rich morphology where a word is composed of zero or more prefixes. This makes Arabic data sparse compared to other languages and consequently word segmentation becomes the important key for many natural language processing tasks that deal with the Arabic language [12].

The ATB is used in most of Arabic language segmentation cases, this is a light segmentation adopted to build parse trees in the Arabic Treebank (ATB) corpus. This type of segmentation considers splitting the word into affixes if and only if it projects an independent phrasal constituent in the parse tree, for detailed examples see [12]. The ATB segmentation scheme considers splitting only a subset of prefixes and suffixes from the stem. When using ATB segmentation, the number of words is similar to its counterpart in English [12].

This is one reason why ATB segmentation is widely used in building machine translation systems for the English-Arabic language pair.

4. PROPOSED MODEL

In the preceding sections, there are two proposed sub models the first model is responsible for the CLIR web service discovery process and the second sub model is responsible for the web service recommendation process, the research in hand, first proposed the new IR-web service discovery with the two sub models inside it that described in each sub model alone.

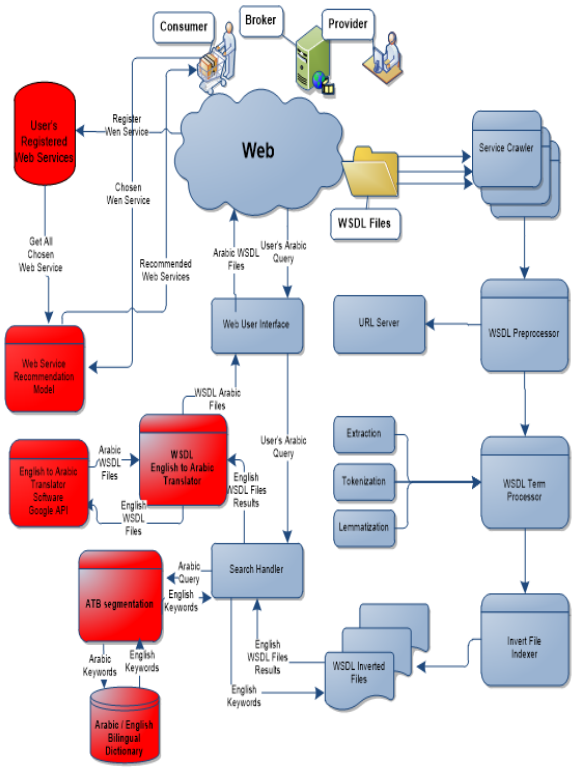

Fig 3 CLIR semantic web service discovery with service recommendation model

Figure 3 shows the design of the CLIR semantic web service discovery with service recommendation model, this model contains ten components that represents that gear box of the model, the red components represent the contribution of our paper, other components are described in IR- web service discovery model in section 3.

The added components are described in the following subsections with its data flow.

4.1 WSDL document

A WSDL document defines services as collections of network endpoints, or ports with semantic annotations that describe the type of information inside the document. In WSDL, the abstract definitions of endpoints and messages are separated from their concrete network deployment or data format bindings. This separation supports the reuse of abstract definitions: messages, which are abstract descriptions of exchanged data, and port types, which are abstract collections of operations [13] .

4.2 Web Services Crawler

WSDL documents are collected by web service Crawlers, which fetch documents from the Internet alternatively, or collect them from some well-known service datasets such as the one provided in. Crawlers hand over retrieved WSDL files and associated HTML files to the WSDL preprocessor for link analysis. This yields a list of new URLs that may point to some new WSDL files. The URL server, several WSDL crawling algorithms are presented in [11], assigns these URLs to an idle crawler.

4.3 WSDL term Processor

Web service crawlers send the WSDL documents to the WSDL term processor component, this component are responsible for extracting the important information exists in the documents. Tokenization is crucial for IR techniques to function properly show. This is because most IR methods like the inverted indexes do not perform a full-text online searching for efficiency. They use string-matching algorithms to look up a well-crafted index containing a set of pre-defined terms prior to the searching process [11].

4.4 Invert File Indexer

Invert file is widely used for indexing text database. To support efficient information retrieval, the words of interest in the text are sorted alphabetically. For each word, the inverted file records a list of identifiers of the documents containing that particular term [9].

Applying the inverted file in WSDL discovery process will be done through dealing with WSDL files as a documents that containing the terms that needed to calculate the (fd,t) pairs the frequency of t appearing in wsdl document.

4.5 Web User Interface

The web user interface represents the entry point to client for finding the required web service. The client writes his query in English or Arabic language that consists of search criteria such as type of service, preferred price range what products are associated with this service, with which categories in company and product taxonomies this web service is associated as well as other technical characteristics. A query is executed against the web service information in the registry that was entered by the web service provider [7].

4.6 Handling the user's Query

The user's query is entered in the web interface, and then it is transferred to the search handler components that have three main tasks [6]:

- If the user's query is entered in Arabic language then it sends the Arabic query to the ATB segmentation component that in return sends the translated keywords in English language to search handler again.
- The other probability is the user's query is entered in English that the search handler extracts the English keywords: this tasks is done through applying tokenization process as mention before to extract that important words for finding the required web service.
- Applying the similarity process between the user's query and the WSDL inverted files leads to find the similar WSDL documents to the user's query.

4.7 ATB Segmentation

The query translation approach in CLIR is applied in this model. The query translation approach is cost efficient especially if it was compared with the translation of each WSDL document in the collection into every language represented in it.

The first step in the Arabic query translator is query segmentation. ATB segmentation algorithm is used for dividing the user's Arabic query into several tokens [13].

The Arabic tokens is similar to the English one which makes the translation process is easier. The following example shows the Arabic query as an input and the segmented Arabic tokens as an output of the process [13].

Arabic query:

تحويل من PDF الى نص

After applying the ATB:

(S (NP <tahweel تحويل(PP <min من(NP< PDF PDF (PP <ilaY إلى(NP< nas نص)))))

After applying, the ATB segmentation algorithm the output is a several tokens, which each one could be an entry in bilingual dictionary.

4.8 Arabic / English Bilingual Dictionary

The bilingual dictionary is a specialized dictionary used to translate words or phrases from one language to another. For this purpose, we searched for Arabic / English open source dictionary. The research found that the ARABEYES is aimed at fully supporting the Arabic. It is designed to be a central location to standardize the Arabization process.

The extracted tokens from the ATB are sent it to the ARABEYES to get the translated keywords as shown in the following example:

Arabic Tokens:

- تحويل
- من
- PDF
- إلى
- نص

ARABEYES output:

Table 1: ARABEYES Translated Words

English	Arabic
Conversion	تحويل
From	من
PDF	PDF
To	الى
Text	نص

The translated keywords are sent to the search handler components. Search handler components use the inverted file techniques to find the WSDL documents that match the user's query keywords. Each one of the translated keywords is compared to inverted lists. Inverted lists contain each WSDL documents with t (notice that each document is represented as pairs < d, fd,t > d is a document number and the frequency (fd,t) of t appeared in d. then the arranged WSDL document list according the term frequency in the document is returned to the search handler component.

4.9 WSDL English to Arabic Translator

This component is needed if the user's query is written in Arabic language; the search handler component gets the WSDL Documents that matching the user's query that are written in English language. These documents should be translated to Arabic, before the translation process the information inside the WSDL document should be extracted. As mentioned in section 2.2 the important information is exists in the attribute names that contain a semantic information about the selected web services and its values

contain the information that is related to the previous attributes, the other important part in the WSDL document is the <documentation> element. This part is written in natural language as a description for the web service, which describe the purpose of the service and other information provided by the service provider, the following example in figure 4 shows some of this information:

```
..............
<service name="PdfToTextService">
  <port name="PdfToTextPort" binding="tns:PdfToTextPortBinding">
 <WSDL:address location="http://gnode1.mib.man.ac.uk:8080/
  FullTextWebServices/PdfToTextService"></soap:address>
  </port>
</service>
<documentation>
  This service will extract the text content from a PDF file.
  It uses the pdftotext executable from Xpdf
  (http://www.foolabs.com/xpdf/).
  The text returned from this service
  often contains characters which are XML-invalid,
  therefore the text is returned in i
</documentation>
<portType name="PdfToText">
<operation name="pdfToTextBase64">
<input message="tns:pdfToTextBase64"></input>
<output message="tns:pdfToTextBase64Response"></output>
</operation>
<operation name="pdfToText">
<input message="tns:pdfToText"></input>
<output message="tns:pdfToTextResponse"></output>
</operation>
</portType>
|.................................
```

Fig 4 Returned WSDL document form the search handler

The extracted WSDL documents are sent to the WSDL English to Arabic translator component. The attributes names and its values are translated to Arabic. For this purpose they will be sent to WSDL translator component as a pairs like in the following example:

< (Attribute Name) service name=(Attribute Value)"PdfToTextService">

In addition, the output will be like:

< اسم الخدمة ="خدمة نص الى PDF">

After translating all WSDL, attributes and their values the WSDL translating components will collect all translating process output and put them in the form like figure 5.

The two WSDL documents the English and the translated are sent to the user or to the service requester, which will give the user a better understanding of the WSDL documents, and let the user to decide which service is the best match for his query.

The retrieved WSDL documents were translated to Arabic. This process is done through Google translation API that shown in figure 6. This component is crucial because the WSDL document contains some attributes elements that contain a long description of the service which like:

<documentation> this service will extract the text content from a PDF file. It uses the pdftotext executable from Xpdf (http://www.foolabs.com/xpdf/). The text returned from this service often contains characters, which are XML-invalid, therefore the text is returned in I </documentation>

This section needs a full translation with applying the grammars rules and machine learning techniques. Google translation API is a free statistical multilingual machine translation service provided by Google Inc. to translate written text from one language into another. Google API client libraries are available in a number of popular programming languages like .NET and Java. The API provides access to a service and provides a single URI that acts as the service endpoint. The following figure show a sample of the programming code using c# for accessing the Google API, the following functions has two input:

- The WSDL attribute and its value as a pair.
- The language pair, which in our case is English to Arabic.

In addition, the output is the translated text:

```
..............
<اسم الخدمه="الى نص pdf ">
<اسم المنفذ="الى نص pdf " الملزم="tns:الى نص الملزم المنفذ ">
<WSDL:عنوان المكان="http://gnode1.mib.man.ac.uk:8080/
FullTextWebServices/PdfToTextService"></soap:العنوان>
</المنفذ>
</الخدمه>
<التوثيق>
وهذه الخدمة استخراج مضمون النص من ملف PDF. انه يستخدم PDF لتنفيذ النص من PDF
(http://www.foolabs.com/xpdf/).
عاد النص من هذه الخدمة
غالبا ما تحتوي على أحرف والتي هي غير صالحة XML .
ولذلك يتم إرجاع النص
</التوثيق>
<اسم نوع المنفذ="الى نص pdf ">
<اسم العملية="الى نص pdf 64">
<المدخلات></مدخلات الرساله="tns: الى نص pdf 64">
<المخرجات></مخرجات الرسالة="tns: الى نص pdf 64 استجابة">
</العمليه>
<اسم العملية="الى نص pdf ">
<المدخلات></مدخلات الرساله="tns: الى نص pdf ">
<المخرجات></مخرجات الرسالة="tns: الى نص pdf استجابة">
</العمليه>
</نوع المنفذ>
<binding name="PdfToTextPortBinding" type="tns:PdfToText">
<operation name="pdfToTextBase64">
...................................
```

Fig 5 Output of the translation process

```
public string TranslateText(string input, string languagePair)
{
 string url = String.Format("http://www.google.com/translate_t?hl=en&ie=UTF8&text={0}&langpair={1}"
, input, languagePair);
 WebClient webClient = new WebClient();
 webClient.Encoding = System.Text.Encoding.UTF8;
 string result = webClient.DownloadString(url);
 result = result.Substring(result.IndexOf("<span title=\"") + "<span title=\"".Length);
 result = result.Substring(result.IndexOf(">") + 1);
 result = result.Substring(0, result.IndexOf("</span>"));
 return result.Trim();
}
```

Fig 6 C# code for dealing with Google API

4.10 Web Service Suggestion Process

Web service's suggestion component is divided into several sub processes that are shown in the following figure and each part is explained in the next discussions.

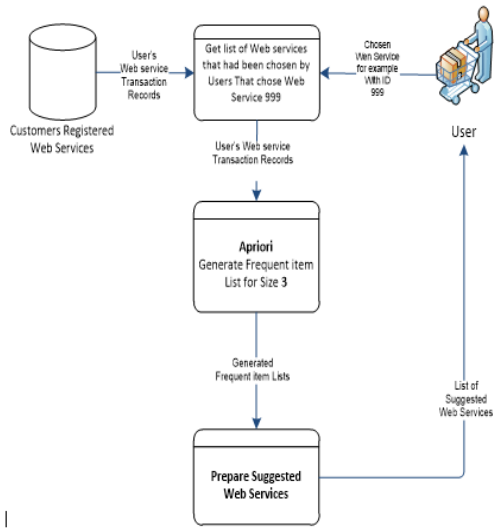

Fig 7 Web service suggestion process [7]

The suggestion process starts from the point that CLIR-Web service discovery model ends, the CLIR model responsible for return the WSDL documents that matches the user's query then the user makes the decision to choose one of the returned services. The user registers the service by using curated web service registry in the profile that is used in the suggestion process. The component consists of three processes:

a. First Process:

The first process takes the chosen web service as an input, and then it queries the web service user's registry that contains users' data with their registered web services by the following Pseudo code:

Get the chosen web service (A) by user U1

Get all users that registered the web service (A)

For each user get his registered services.

For each user prepare the list as following: <User I,W1,….Wi>, after preparing these pairs then it is send to the next process.

b. Second Process:

This process takes a list of transaction records that were extracted from the first process then put them as input Apriori algorithm.

We choose the list size 3 to reduce the memory, processor and time usage and our objective to suggest one service that is the most similar to users' behavior. Apriori algorithm is working until the item frequent list is empty, the size of pairs in frequent list is incremented by 1until the list is empty, and then put the stop condition as it is:

The items count in each Pair is three

And the final frequent list will be as following example:

L= {{A, B, C}, {A, D, F}…….}

Note: the chosen service is A

Then the process sends this list to the next process.

Third Process:

This process computes the confidence form each pairs with the condition that know is A like:

For {A, B, C}:

Confidence=A BC and so on.

Then the pair with the greatest confidence value it will send to the user as follow:

Suppose {A, B, C} is the winner pair: then the process suggest the services B and C to the user based on user's profile analysis, then user will decide if he will use it or not.

4.11 Proposed Model's Workflow

As mentioned in previous sections all WSDL documents are collected by service crawlers then it is passed to the WSDL processing components to build the inverted file index for all WSDL documents. The search handler component accesses the inverted file index to search for services that matching the user's query.

The example of proposed model workflow in the CLIR web service discovery is explained in the following steps:

- The process starts by entering the web service query through the web user interface ex : "مكان الوزن مصفوفات"
- User's query is sent to the search handler component.
- The search handler component sends the use's query to the ATB segmentation component.
- The ATB starts processing the user's query for extracting the key words by the segmentation and lemmatization methods.
- The ATB produced three keywords: "مكان","وزن","المصفوفات"
- The keywords re sent to the Arabic/English bilingual dictionary for getting the translated keywords.
- The bilingual dictionary returns the translated keywords to the ATB component that it is in return sends the keywords to the search hander component ,the translated keywords are "Position"," Weight"," Matrices".
- The search handler component calculates the similarity between the translated keywords and the WSDL inverted file index for finding the most similar documents.
- The search handler component returns a web service titled by "Jaspar_service" and it sends its WSDL document to the WSDL English to Arabic translator as shown in figure 8.
- The WSDL English to Arabic translator processes the English WSDL document and extracts tags names and their contents then send them to Google API.
- Google API translates the tags names and their contents then it is sent back to the WSDL English to Arabic translator for preparing the tags names and their contents and putting them in the WSDL document format as shown in figure 9.
- The translated WSDL document is sent to the user through we user interface.
- The example of proposed model work flow in the web service recommendation process is explained in the following steps:
- The recommendation process starts after registering the selected service by the user.

- Recommendation component get list of web services that had been chosen by users that registered the selected service then it sends them to Apriori algorithm with three frequent items stop conditions.
- Apriori algorithm scans web service registration database for all users that selected the same web service then it find the 3 most related web services to the selected web service.
- Recommendation component send the suggested web services to the user waiting for his decision if he will select one of them or not as shown in figure 10.

```
<xs:element name="searchByTagResponse">
<xs:complexType><xs:sequence>
<xs:element ref="jas2:Matrix" minOccurs="0" maxOccurs="unbounded" />
</xs:sequence></xs:complexType></xs:element>
<xs:element name="getMatrixByName">
<xs:complexType><xs:sequence><xs:element name="Name" type="xs:string" />
<xs:element ref="Format" /><xs:element ref="Database" /></xs:sequence>
</xs:complexType></xs:element><xs:element name="getMatrixByNameResponse">
<xs:complexType><xs:sequence><xs:element ref="jas2:Matrix" />
</xs:sequence></xs:complexType></xs:element>
```

Fig 8 WSDL document

```
<xs:اسم العنصر="بحث بواسطة الكلمات الدلالية الاستجابة">
<xs:تسلسل><xs:نوع مركب>
<xs:عنصر مرجع="jas2:مصفوفة" minOccurs="0" maxOccurs="غير محدود" />
</xs:عنصر></xs:نوع مركب></xs:تسلسل>
<xs: اسم العنصر ="اجلب المصفوفة بالإسم ">
<xs:نوع مركب><xs:تسلسل><xs:اسم العنصر="اسم" نوع:xs="سلسلة" />
<xs:شكل="قاعدة بيانات:xs:عنصر مرجع" /><xs:عنصر مرجع" />
</xs:عنصر></xs:نوع مركب></xs:تسلسل>
<xs:اسم العنصر="الحصول على مصفوفة حسب اسم الاستجابة">
<xs:نوع مركب><xs:تسلسل><xs:عنصر مرجع="jas2:مصفوفة" />
</xs:عنصر></xs:نوع مركب></xs:تسلسل>
```

Fig 9 WSDL document returned from Google API

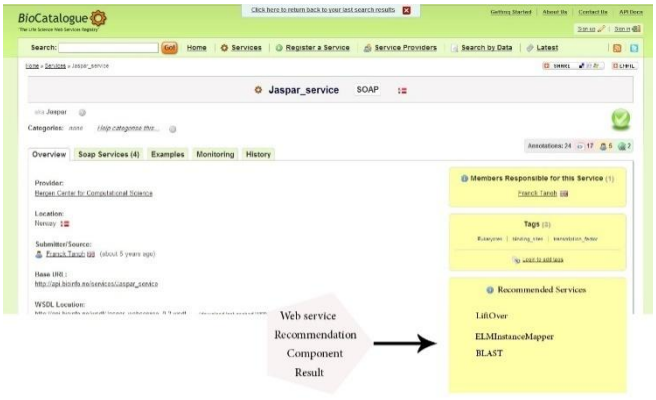

Fig 10 Web service's recommendation process output

5. EVALUATION

As mentioned above the contribution of the research proposed model is divided into two sub-models that each one needs to be tested separately, therefore the researcher made and experiment with different procedures for each sub-model.

5.1 Experiment 1: Web Service Discovery Model with CLIR Components

This experiment for testing the contribution of applying the CLIR and the inverted file index techniques are in the web service's discovery process. As we mentioned before the data set that is needed in case study was built through collection of 400 WSDL documents, the inverted file index was built for all these file and stored in a data store for users' queries.

The of cross language information retrieval contains an additional step to the monolingual information retrieval, this step is the translation process as mentioned in section two: ***CLIR=translation + monolingual IR***

Therefore, the goal of CLIR is to achieve as much as possible of effectiveness of the monolingual IR system, which means that the translation process has the highest effect on the CLIR performance. The idea to measure the effectiveness of the web service CLIR model is comparing the proposed web service CLIR model with the monolingual web service IR. The language used in the CLIR model is Arabic and the language used in the monolingual IR is English. The experiment procedures are:

- Choosing 400 key words from the 200 WSDL documents (1 word from each documents), they are listed in the supplementary file.
- All selected key words are translated to Arabic.
- There is 400 search attempts each one is done with English keywords in the monolingual IR and Arabic keywords in the CLIR.
- The results that returned form the monolingual system was supposed as the true positive results that should be returned from the CLIR model.
- The percentage of matching between the two queries' results is registered.
- The average of matching of all 400 queries represents the effectiveness of the CLIR model to the monolingual
- The accuracy and precision of the CLIR queries in matching the monolingual IR queries are calculated through binary classification as the following table:

Table 2: Binary Classification

	True	False
Positive	True positive	False positive
Negative	False negative	True negative

- The classification is done through the following rules:
  - If the result CLIR system some retrieved WSDL documents that existed in the monolingual IR result so these documents is counted as a true positive documents.
  - If the result CLIR system ignored some WSDL documents that do not exist in the monolingual IR result so these documents are counted as true negative documents.
  - If the result CLIR system retrieved some WSDL, documents that do not exist in the monolingual IR

result so these documents are counted as true positive documents.

- If the result CLIR system ignored some WSDL documents that existed in the monolingual IR result so these documents is counted as a false positive documents.

- The accuracy is defined by Nello in [14] "the proportion of true results (both true positives and true negatives) in the population":

$$accuracy = \frac{\text{true positives} + \text{true negatives}}{\text{true positives} + \text{false positives} + \text{true negatives} + \text{false negatives}}$$

- On the other hand, precision or positive predictive value is defined by Nello in [14] as "the proportion of the true positives against all the positive results (both true positives and false positives)":

$$percision = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

- Then the researcher calculated the average accuracy and average precision for the 400 Arabic queries.
- The returned WSDL documents from CLIR model were translated to Arabic language through Google APIs.

5.2 Experiment 2: Web Service Discovery Model with Recommendation Components

For testing the proposed web service's recommendation component, the research build the experiment by using the BioCatalogue curated Web Services that has the following abilities:

- Keyword search.
- Input and output data search.
- Display all details about web service.

The previous functions are available on the BioCatalogue.com website through a defined web interface, also there are an available API's to access all BioCatalogue.com searching, filtering, browsing data and WSDL documents.

For building the required data set:

- Choosing 40 user's randomly from BioCatalogue.com.
- Choosing the first registered web service for each user as a starting point of search to find similar users.

Experiment procedure for Web Service recommendation for a user as an example:

- Suppose User 1 registered a web service titled by "Convert PDF to Text"
- The BioCatalogue API will return a list of users' transaction that chose Convert PDF to Text" shown in table 6, and table 5 contains symbol of each service to make the calculations easier for readers as an example.
- Applying the Apriori algorithm confidence 70%, minimum support 20 %, and with 3 item size stop condition, with existence of "Convert PDF to Text" and eliminate other lists that don't satisfy this condition.
- Choosing the 70 % as confidence because this value is an average value, if the value is high, the chance to return items is very law and if the value is low it may return a non-related web services.
- Choosing the 20 % as minimum support value to increase the size of frequent itemsets and increase the chance to recommend service to the user.
- Table 7 shows frequent lists with size 2 items
- Applying the Apriori algorithm with confidence 70% were listed in Table 8A and Table 8B.
- The recommended service for user1 is searched with it in the user1's history; the existence of the service in the user's history proves the success of the proposed component.

Table 5: Services and their symbols

No	Service Name	Symbol	Suppor
1	Convert PDF to Text	A	9
2	Image Retrieve	B	7
3	Text search and retrieval from	C	6
4	Similarity sequence databases	D	2
5	Chemical text mining	E	2

Table 6: Users Transactions

No	Transactions
1	A, B, E
2	A,B, D
3	A,B, C
4	A, B, D
5	A, C
6	A,B, C
7	A, C
8	A, B ,C, E
9	A, B, C

Table 7: Frequent Item lists with size 2

No	Itemsets	Support	Decision
1	AB	7	OK
2	AC	6	OK
3	AD	2	OK
4	AE	2	OK
5	BC	4	Eliminated for Not Existence of A
6	BD	2	Eliminated for Not Existence of
7	BE	2	Eliminated for Not Existence of
8	CD	0	NO
9	CE	1	NO
10	DE	0	NO

Table 8 A: Frequent Item lists with size 3

No	Itemsets	Support	Decision
1	ABC	4	OK
2	ABD	2	OK
3	ABE	2	OK

4	ACD	0	NO
5	ADE	0	NO

Table 8B: Frequent Item lists with size 3 with Confidence

No	Itemsets	Confidence	Decision
1	A=>BC	4/9	NO
2	A=>BD	2/9	NO
3	A=>BE	2/9	NO
4	A=>B	7/9	OK
5	A=>D	6/9	NO
6	A=>E	2/9	NO
Note: All other Itemsets are eliminated because we put the existence of Service A a basic condition			

- Then as shown in Table 8B the item list that satisfy our conditions and the confidence threshold is A=>B, so the output of our model is:
- Recommend Service B for User1 Service B titled by
- "Image Retrieve"
- The 40 selected users were tested with same procedures and the recommended web services for each user were registered.
- The decision of each user of accepting the recommended web service was registered to measure the degree of success of the proposed model to gain the user satisfaction about the recommended web service.
- The percentage of users that accept the recommended service represents the percentage of success of the proposed model.

5.3 Results of Experiment 1

Table 9 shows the returned SWSDL documents from the monolingual and CLIR systems with the binary classification then we calculated the average accuracy and average precision for the 400 Arabic queries:

Average Accuracy = (Total accuracy/No of queries) = (39752.25/400) =99.38 %

Average Precision = (Total precision /No of queries) = (34893.71/400) =87.23 %

In order to clarify the results of the 400 queries in the monolingual model and the CLIR model, we put the queries' results of total returned documents in the two models in charts shown in figure 11 and the total returned documents in the monolingual model with only the true positive of the CLIR model in charts shown in figure 12 for displaying the effectiveness between the two models.

Table 9: Arabic Queries and Their results

No	Arabic Query	Number of returned WS from English Query	Total No of returned WS	No of True Positive	No of True Negative	No of False Negative	No of False Positive	Accuracy%	Precision%
1	تحويل PDF إلى نص	13	13	13	387	0	0	100	100.00
2	تنظيف النص من الطبقات غير المرغوب فيها	3	3	3	397	0	0	100	100.00
3	البحث عن النص واسترجاع من قواعد البيانات الكبير	3	2	2	397	1	0	99.75	66.67
4	تحليل تسلسل البروتين	12	12	12	388	0	0	100	100.00
5	التعرف على الكيان المسمى	16	16	16	384	0	0	100	100.00
6	التعرف على الطب الحيوي للكيان المسمى	2	1	1	398	1	0	99.75	50.00
7	اكتشاف الوثيقه	3	7	2	392	1	5	98.5	66.67
8	تشابه الوثيقه	1	1	1	399	0	0	100	100.00
9	تجميع الوثيقه	1	1	1	399	0	0	100	100.00
10	تصنيف النص	1	4	1	396	0	3	99.25	100.00
11	رموز الغموض الفوكليونيدات	20	24	20	376	0	4	99	100.00
12	كتابة تسلسل البروتين	14	15	14	385	0	1	99.75	100.00
13	قراءة الحمض النووي	1	1	1	399	0	0	100	100.00
14	ترجمة سلاسل الببتايد	25	29	25	371	0	4	99	100.00
15	التحليل الإحصائي للبروتين	2	2	2	398	0	0	100	100.00

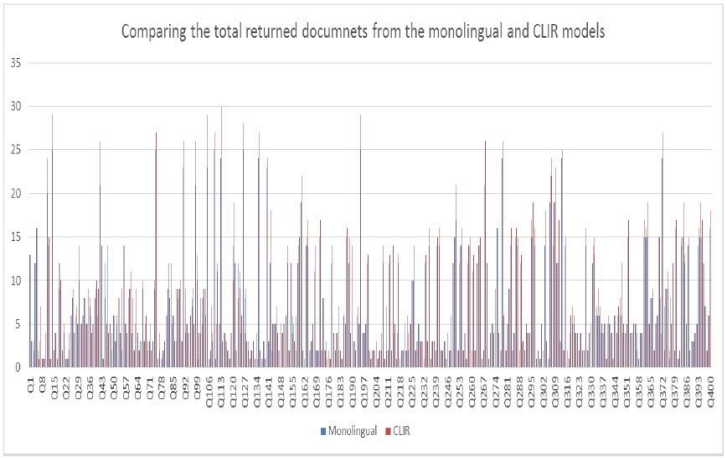

Fig 11 Monolingual vs. CLIR (Total returned documents)

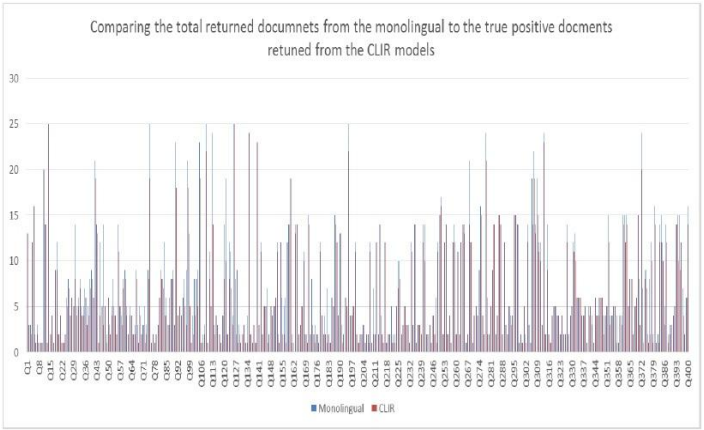

Fig 12 Monolingual vs. CLIR (true negative documents

The returned WSDL documents from CLIR web service's discovery model were translated to Arabic language. The following figures show the one of returned WSDL document in English and the translated to Arabic and we mentioned 20 of the translated documents in the supplementary file as an example of the WSDL document translation process.

```
<wsdl:message name="Count Request Message">
<wsdl:part name="parameters" element="ns:Count"/>
</wsdl:message>
<wsdl:message name="Count Response Message">
<wsdl:part name="parameters" element="ns:Count Response"/>
</wsdl:message>
<wsdl:message name="Find Boolean Request Message">
<wsdl:part name="parameters" element="ns:Find Boolean"/>
</wsdl:message>
<wsdl:message name="Find Request Message">
<wsdl:part name="parameters" element="ns:Find"/>
</wsdl:message>
<wsdl:message name="Find Response Message">
<wsdl:part name="parameters" element="ns:Find Response"/>
</wsdl:message>
<wsdl:message name="Get Data bankl nfo Request Message">
<wsdl:part name="parameters" element="ns:Get Data bank Info"/>
</wsdl:message>
<wsdl:message name="Get Databank Info Response Message">
<wsdl:part name="parameters" element="ns:Get Databank Info Response"/>
</wsdl:message>
```

Fig 13: Returned WSDL document in English language

```
<اختصار: رسالة اسم = "عدد طلب رسالة">
<اختصار: جزء اسم = "المعلمات" عنصر = "نانوتانية: الكونت" />
</ اختصار: رسالة>
<اختصار: اسم الرسالة = "عدد الرسائل الاستجابة">
<اختصار: جزء اسم = "المعلمات" عنصر = "نانوتانية: عد الاستجابة" />
</ اختصار: رسالة>
<اختصار: اسم الرسالة = "البحث منطقية طلب رسالة">
<اختصار: جزء اسم = "المعلمات" عنصر = "نانوتانية: البحث منطقية" />
</ اختصار: رسالة>
<اختصار: اسم الرسالة = "البحث طلب رسالة">
<اختصار: اسم جزء = "المعلمات" عنصر = "نانوتانية: البحث عن" />
</ اختصار: رسالة>
<اختصار: اسم الرسالة = "البحث عن رسالة الرد">
<اختصار: جزء اسم = "المعلمات" عنصر = "نانوتانية: البحث الاستجابة" />
</ اختصار: رسالة>
<اختصار: اسم الرسالة = "الحصول على البيانات البنكي طلب نفو رسالة">
<اختصار: جزء اسم = "المعلمات" عنصر = "نانوتانية: إحضار بيانات بنك معلومات" />
</ اختصار: رسالة>
<اختصار: اسم الرسالة = "الحصول على معلومات بنك المعلومات رسالة الاستجابة">
<اختصار: جزء اسم = "المعلمات" عنصر = "نانوتانية: احصل على معلومات بنك المعلومات الاستجابة" />
</ اختصار: رسالة>
```

Fig 14 Returned WSDL document in Translated to Arabic

5.4 Results of Experiment 2

Choosing randomly 40 users from the Biocatalogue curated web site with their first registered web service as a starting point for web service recommendation as mentioned in the experiment 2 procedures. Table 10 shows the first 10 users with their registered first registered web services and the rest of users with their services were mentioned in supplementary file.

Appling the Apriori algorithm for each user with confidence 70 % and with the registered web service as the item that we search for most related items to it. The following table shows the first 10 users with their recommended web services and the rest of users with their recommended services were mentioned in the supplementary file.

Table 10: Users and their registered services

User ID	Registered Web Service
User1	Drug susceptibility for HIV Protease strains
User2	FireDB
User3	Readseq
User4	EMBOSS seqret
User5	MapMi
User6	Infernal cmscan
User7	Wise2DBA
User8	PromoterWise
User9	GeneWise
User10	Clustal Omega

We applied the Apriori algorithm for each user with confidence 70 % and with the registered web service as the item that we search for most related items to it. The following table shows the first 10 users with their recommended web services and the rest of users with their recommended services were mentioned in the supplementary file.

Table 11: Users and their recommended services with their decisions

User ID	Recommended Web Services	User Decision
User1	PSIQUIC (IntAct)	Accept
User2	MutalyzerService , MetaLocGramN_SOAP	Accept
User3	genecruiser	Not Accept
User4	Blast_Service	Accept
User5	Brenda_webservice , NeuroElectro	Accept
User6	ModelWebService	Not Accept
User7	PublicationWebService	Not Accept
User8	TYNAService ,	Accept
User9	MyGene.Info	Accept
User10	GeneCruiser	Not Accept

The degree of usability of the web service recommendation model as we mentioned before is determined through measuring the percentage of users' acceptance of the recommended services, for this purpose the users' decisions were known by checking if users had registered one of the recommended services or not. Then we found 26 users accepted the recommended services.

The percentage of users' acceptance = (No of agreed)/ (Total No of users)*100

= (26 / 40) * 100=65 %

6. CONCLUSION

Web service discovery is a very important process special after applied IR techniques. This leads to IR web service's discovery approach, but this limits its use to English language users only. Our approach leads to expand the web service users to other languages and adding a recommendation component to recommend web services to the users based on their history in using them. In our research, we proposed the Arabic language approach that could be used in other languages like Indian, Chinese and so on and another data mining technique that may be lead to better results.

The proposed model achieved the 87 % accuracy and 89% precision of the proposed CLIR web service's discovery model and 65 % in the web service's recommendation model it proved the usability of proposed model for solving the main research objectives that mentioned in section 1

This may lead to further applications that could use multiple language web service and application-to-application different language data exchange.

Our future work is to expand our approach to use different data mining techniques with other languages to improve the selection mechanism.

7. REFERENCES

[1] Dustdar, Schahram, and Wolfgang S.,(2005),"A survey on web services composition.", International journal of web and grid services,Inderscience,Vol.1,No.1,Pp.1-30.

[2] Wang, Y., Stroulia, E. (2003).”Flexible interface matching for web-service discovery”,Proc.Cong. Proceedings of the Fourth International Conference on IEEE. Pp.147-156.

[3] Platzer C., Dustdar S.,(2005), “A vector space search engine for Web services”, Proc.Cong. Third IEEE European Conference on Web Services, Sweden.

[4] Sajjanhar A., Hou J., and Zhang Y.,(2004). “Algorithm for web services matching”,Proc.Cong. Advanced Web Technologies and Applications. Springer Berlin Heidelberg, Pp. 665-670.

[5] Ding, Li, "Swoogle: a search and metadata engine for the semantic web", (2004), Proc.Cong. Proceedings of the thirteenth ACM international conference on Information and knowledge management. ACM.

[6] Sultan T., Khedr A., Alsheref F., (2013), "Cross Language Information Retrieval Model For Discovering WSDL Documents Using Arabic Language Query" International Journal of Advanced Computer Science & Applications, The Science and Information organization, Vol.4, No.8. Pp.118-129.

[7] Sultan T., Khedr A., Alsheref, F., (2013),"Adaptive Model for Web Service Recommendation." International Journal for web service and computing, AIRCC. Vol.9, No.4.,Pp.21-33.

[8] Anne,G., Levow, Douglas W., Oard, Resnik P., (2005),”Dictionary-based techniques for cross-language information retrieval, “Information Processing and Management, Elsevier, Vol. 41, No.3, Pp.523-547.

[9] Callan J., Bruce C., Harding S., (1992), "The INQUERY retrieval system", Database and expert systems applications ,Springer, Pp. 78-83.

[10] Nagata, Masaaki, Saito T., Suzuki K., (2001), "Using the web as a bilingual dictionary.", Proc.Cong. Proceedings of the workshop on Data-driven methods in machine translation.

[11] Wu C., (2012), "WSDL term tokenization methods for IR-style Web services discovery" Science of Computer Programming, Publisher, Vol. 77, No. 3, Pp. 355-374.

[12] Maamouri, Mohamed, Bies A., Buckwalter T., Mekki W.,(2004), "The penn arabic treebank: Building a large-scale annotated arabic corpus.", Proc.Cong. NEMLAR Conference on Arabic Language Resources and Tools, Pp. 102-109.

[13] Christensen, Erik, Curbera F., Meredith G., Sanjiva, Weerawarana, (2001), "Web services description language (WSDL) 1.1.," World Wide Web Consortium.Vol. 1, No. 28.

[14] Cristianini N., Shawe-Taylor J.,(2000), “An Introduction to Support Vector Machines and other kernel-based learning methods”,2nd Ed, Cambridge University Press.