

Novel Semantic Model for Intelligent Discovery of Web Services

Shailja Sharma¹, J. S. Lather² and Mayank Dave³

¹*Department of Computer Applications, National Institute of Technology,
Kurukshetra, India*

²*Department of Electrical Engineering, National Institute of Technology,
Kurukshetra, India*

³*Department of Computer Engineering, National Institute of Technology,
Kurukshetra, India*

Abstract

Main challenges to the current semantic web service technologies are exponential continuous growth in the number of services on the Internet, syntax based discovery, lack of common agreed upon semantic service standards and heterogeneity of ontologies. In this paper, a service discovery approach independent of semantic service description models is proposed to solve the challenges of the current web service discovery. The idea is to combine principles from machine learning, data mining, statistical techniques and measures of semantic relatedness to make the semantic web service discovery process more intelligent, efficient and effective. The proposed approach exploits the use of semantic as well as syntactic information present within the service description profiles. Our approach is unique in terms of its application to any web service description language and the use of Omiotis measure of semantic relatedness for service discovery. The proposed approach has been implemented on OWL-S based service descriptions profiles and is able to find semantic relationship between the services which were otherwise discarded by the OWL-MX matchmaker. Empirical analysis shows that the proposed method out performs the TF-IDF based cosine similarity method, Jaccard method and LSA based approach.

Keywords: *Semantic Web Service Discovery, Measures of Semantic Relatedness, Machine learning, Text Mining, OWL-S*

1. Introduction

Web services provide interoperability of applications and integration of large scale distributed systems irrespective of platform [1]. Currently, the platform elements for Web services are Extensible Markup Language [2], HTTP [3], Web services Description Language [4] and Universal Description Discovery and Integration [5] specifications. Web services are usually described using WSDL definitions and advertised in UDDI registries. Significant drawback of current description standards for Web services such as WSDL and UDDI is that they are restricted to the syntactic aspects of service. The vision of Semantic Web services [6] is to describe and annotate the various aspects of a Web service using machine-understandable semantics, enabling the automatic location, combination, and use of Web services. Various ontology based frameworks like Ontology Web Language for Services [7] (OWL-S), Web Service Modeling ontology (WSMO) [8] etc. have been proposed in past to ease the discovery of semantic Web services. Similarly efforts have been put to add semantics to the existing WSDL standard specifications leading to the introduction of WSDL-S [9] and the W3C recommendation (Semantically Annotated WSDL [10] (SAWSDL) etc. Based on these specifications many authors have proposed semantic service matchmakers like OWL-S i-Matcher [11], OWLS-MX [12], WSMO-MX [13], SAWSDL-i-Matcher [14] and many more. In addition to these, many

solutions based on machine learning, ontology, graph theory, data mining and Information Retrieval based approaches have been proposed for SWS discovery. It has been analyzed that the existing approaches differ on functional, nonfunctional parameter consideration, similarity finding method, expressiveness, tradeoff between accuracy and complexity, and tool support. Lack of agreed upon standards and differences in the context, understanding, domain knowledge and intelligence of user vs. domain expert are some of the challenges that restrict the discovery and reutilization of the Web services over the Internet. To overcome some of the limitations of current web service discovery process we propose an easy to use Information retrieval based approach that is independent of the service description languages and standards.

In this paper, we present a semantic web service discovery framework which utilizes the principles from Machine learning, Data Mining, Statistical techniques and Measures of Semantic Relatedness. Our framework is based on a recently proposed Measure of Semantic Relatedness, Omiotis [15] that takes into account all the semantic relations available in the WordNet, the word thesaurus and lexical ontology, and is capable of handling the synonymy and polysemy problems. Research has shown that using Omiotis, the text classification performance can be improved [16]. In the proposed approach, we combine the semantic information with the Information Retrieval based syntactic information to calculate the semantic relatedness score. We use service input parameters, output parameters, service descriptions, and other information for discovering the set of relevant Web services meeting the service requestors functionality and requirements. The proposed framework is divided into three parts 1) Service-Dimension semantic relatedness matrix calculation 2) Integration of syntactic information with semantic information 3) Calculation of degree of semantic match between a query and a service. OWL-S and WSDL based services have been used as our datasets for experimental purposes but this framework is not restricted to any specific approach for describing services on the Semantic Web like OWL-S, WSMO, WSDL-S.

The remainder of this paper is structured as follows. We review various approaches on semantic service discovery in subsection 2.1 followed by a discussion on Omiotis measure of semantic relatedness in subsection 2.2. Section 3 presents our proposed approach to semantically discover the services, overview, related process and algorithm. Details on the implementation of OSD approach, tools, dataset are given in Section 4 followed by an experimental analysis, comparisons, results and discussion. We present conclusion and future work in Section 5 followed by references.

2. Literature Review

2.1. Related Work

Most of the service discovery approaches present today strictly adhere to single service description languages and standards like DAML-S, OWL-S, WSMO, WSDL, SAWSDL *etc.*, thereby posing the problem of heterogeneity. These approaches vary on I/O matching vs. Input Output Precondition Effects (IOPE) matching. DAML-S based approach proposed by Paolucci *et al.*, [17] and OWL-S based approaches proposed by Klusch *et al.*, [12], Bruijn *et al.*, [18], Kawamura *et al.*, [19], Gronmo and Jaeger [20], Li and Horrocks [21] and Wang and Li [22] and SAWSDL based approach proposed by Wei *et al.*, [14] performs Input-Output (IO) matching on service profiles whereas approaches proposed by Klusch and Kaufer [13] and Sycara *et al.*, [23] perform IOPE matching. Research has also been done to extend the existing frameworks like WSDL using SAWSDL and WSDL-S. A WSDL-S based discovery technique over federated registries using the METEOR-S infrastructure has been proposed by Verma *et al.*, [24].

Some hybrid approaches by Klusch *et al.*, [12], Klusch and Kaufer [13], Sycara *et al.*, [2] and Wei *et al.*, [14] have also been proposed which consider semantic as well as

syntactic description of the services. OWLS-MX proposed by Klusch *et al.*, [12] is an OWL-S based hybrid matchmaker that applies five different matchmaking filters: exact, plug in, subsumes, subsumed-by, and nearest-neighbor to give semantic as well as hybrid degree of matching. In comparison to input and output (I/O) parameters matching in [12], IFL based LARKS by Sycara *et al.*, [23] performs IOPE matching. LARKS does not support logical subsumes and hybrid nearest neighbor and has never been evaluated experimentally. Similar to OWL-MX, WSMO-MX [13] is also a hybrid matchmaker but it matches IOPE's of profiles instead of (I/O) and is based on WSMO framework. SAWSDL-iMatcher [14] annotates semantics to the existing profiles and supports user customizable matching strategies according to different application requirements.

Among the various data mining based approaches, Nayak and Lee [25], Batra and Bawa [26], and Wen *et al.*, [27] use classification and clustering to find semantic similarity between the services. SWSC [25] method uses Jaccard coefficient and hierarchical agglomerative clustering whereas Wen *et al.* [27] has modified the K-Means clustering mechanism to sort out the problem. A classification based approach that uses the Normalized semantic score MSR has been proposed by Batra and Bawa [26] for semantic web service discovery. In the approach proposed by Gracia *et al.* [28], SPARQL-based repository filter has been introduced for improving the semantic web service discovery. They have added one more preprocessing stage before discovery where the service's functional and non-functional terms that do not match with the user requests are filtered out thereby reducing the search space.

Despite their semantic tagged descriptions there are several limitations associated with these approaches. Some of the limitations are:

1. Frameworks like OWL-S by [7], WSMO [8] and WSDL-S [9] assume request itself as a web service and ask the user to express the request in a formal specified language which requires that web service users be friendly with these technologies and frameworks as well as to have detailed knowledge about the description and implementation details of a web service. This may not always be possible for a naive user and hence the usage becomes difficult. Service requestors are more familiar with the keyword based requests as they do in search engines and UDDI registries. These frameworks are not flexible enough to incorporate keyword based or natural language based discovery.
2. It is not possible to expect all service requestors and service providers to have same understanding of context and to use same ontological concepts. Service provider and Service requestor may not use the same ontology. Moreover, service requestor may not have the complete knowledge about the domain Ontology and the other technical terms. Therefore, he frames his request using user ontology which needs to be mapped with domain ontology. Some of the current approaches for describing services on the Semantic Web (*e.g.*, OWL-S [7]) do not support establishment of semantic correspondences between different ontologies. Mediation aspects are not considered in the OWL-S [7]. Some frameworks like, WSMO [8] and WSDL-S [9] have been proposed to provide support for the needed inter-ontology mapping, but it is a time consuming task and may sometimes lead to misinterpretation of concepts.
3. Discovery in many approaches [12, 13, 25, 17, 23, 24, 30, 14] are restricted and limited to Web services that are published in a specific description standard only (OWL-S [7], WSMO [8], WSDL-S [9], SAWSDL [10] standards *etc.*). These approaches do not consider matching a request with Web services described in other formats and specifications. Further, there is deficiency of Inter standard mapping techniques.
4. To expect semantic tagged descriptions for all the new services is not practical.
5. Already existing Web services whose descriptions are written in syntax based languages like WSDL do not have explicitly associated semantics in their

- descriptions. To annotate semantics to the existing millions of services is time consuming, cumbersome and nearly impractical.
6. No agreed upon single standard is there which can overcome the problem of heterogeneity in all aspects like ontology, language and frameworks.
 7. Most of the approaches [12, 13, 17, 21, 30, 31] deal with input and output matching only whereas service descriptions sometimes play a vital role in web service discovery.
 8. Some of these approaches [21] do not tell the semantic similarity values between a request and services. Therefore, two services having same degree but different semantic content cannot be distinguished and hence cannot be ranking.

In order to address some of the cited limitations of existing approaches, we propose a framework that is independent of the service description language and standard and generates concrete similarity score leading to ranking of resultant services.

2.2. Semantic Relatedness and the Omiotis Measure

In Omiotis, final semantic relatedness score between texts is calculated by combining the words' statistical information with the semantic relatedness at word level. For measuring the relatedness between different texts Semantic Relatedness (SR) value between words is used. The word-to-word relatedness measure, in its turn, is based on the construction of semantic links between individual words, according to WordNet. Omiotis is based on a sense relatedness measure, called SR. For weighting the semantic path Omiotis considers three key factors: (a) the semantic path length, (b) the intermediate nodes specificity denoted by the node depth in the thesaurus' hierarchy, and (c) the types of the semantic edges that compose the path. Omiotis is the first measure of semantic relatedness between texts that considers all three factors for measuring the pairwise word-to-word semantic relatedness scores. Experimental evaluation proved that Omiotis measure of semantic relatedness approximates human understanding of semantic relatedness between words better than previous related measures[15]. Semantic relatedness for a pair of terms $T(t_1, t_2)$ is calculated as follows [15]

Definition 1 Let a word thesaurus O , let $T = (t_1, t_2)$ be a pair of terms for which there are entries in O , let X_1 be the set of senses of t_1 and X_2 be the set of senses of t_2 in O . Let $S_1, S_2, \dots, S_{|X_1| \cdot |X_2|}$ be the set of pairs of senses, $S_k = (s_i, s_j)$, with $s_i \in X_1$ and $s_j \in X_2$. Now the semantic relatedness of $T(SR(T, S, O))$ is defined as

$$\max_{S_k} \{ \max_P \{ SCM(S_k, O, P) \cdot SPE(S_k, O, P) \} \} = \max_{S_k} \{ SR(S_k, O) \} \text{ for all } k = 1.. |X_1| \cdot |X_2| \quad (1)$$

For details about the Omiotis and various terms used here reader is requested to refer [15].

3. Proposed Methodology for Semantic Service Discovery

3.1. Overview

We propose an Omiotis based Service Discovery (OSD) approach which combines the semantic as well syntactic technologies to generate a concrete similarity value, which would be useful for ranking services and is not dependent on any specific service modeling language. This approach helps in the intelligent discovery of services by incorporating the techniques of machine learning, data mining, statistical means and Measures of semantic relatedness.

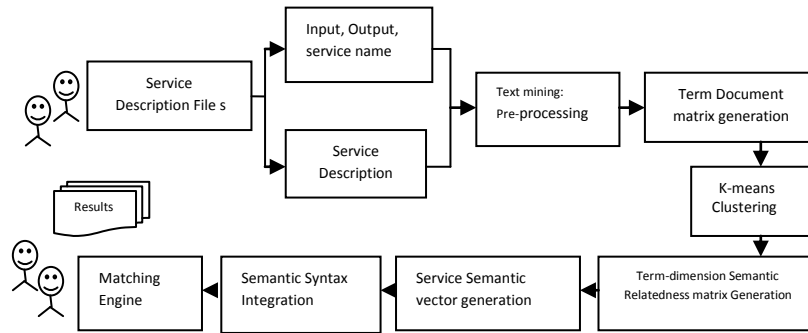


Figure 1. Semantic Web Service Discovery Framework

Any Web service specification language that provides required information about a Web service could be used for the Web service description. We choose the widely used language WSDL and OWL-S, which is proposed by W3C. Among the Service Profile, Service Model and Service Grounding provided by the OWL-S description, in this paper we are using OWL-S Profile concepts only. Our proposed method is not restricted to OWL-S only.

The proposed framework is divided into three parts:

- 1) Service-Dimension semantic relatedness matrix calculation.
- 2) Integration of syntactic information with semantic information.
- 3) Semantic match degree calculation.

In our approach, the semantic information is combined with the Information Retrieval based syntactic information to compute the semantic similarity score. Complete framework is depicted in Figure 1. Function of the Hybrid Semantic Interpreter is to extract and integrate the semantic and the syntactic information of each service. Hybrid Semantic Interpreter transforms the information contained within the service profiles into semantically enriched vectors and passes them to the similarity match engine where Semantic match degree calculation is done by matching the query vector with all services within the cluster to generate a ranked list of services. The proposed algorithm for semantic service discovery is depicted in Figure 2.

3.2. Hybrid OSD Algorithm

To correlate each service semantically with the various dimensions, services need to be pre-processed. In the first step of the algorithm we have used Text Mining techniques, mainly Pre-processing to extract relevant information from a service description file. The Pre-processing deals with extraction of Input, Output, Service names and other textual documentation accompanying the Web services. After pre-processing, the term frequency- Inverse document frequency ($TF-IDF$) [29] matrix is generated for all extracted terms in second step. Many services present in the service set may contain different content that may not be relevant for the user query. To match a user query with all the services in the service set may be a computationally costly task. Therefore, in order to improve the efficiency of service discovery and reduce the computational cost, the clustering algorithm is used where the related services are grouped together in the same clusters and the non relevant services present in other clusters are not considered; thereby reducing the search space. In step 3, K-Means Clustering algorithm has been used to cluster the service datasets. Step 4 deals with integrating the syntactic and semantic information of the service.

Within every cluster, the semantic relatedness of each service with all the pre specified dimensions are calculated. For all the terms present in the input, Output, service name and descriptions/comments with in a service profile, a Semantic term-dimension relatedness

matrix is computed by applying the Omiotis MSR to different dimensions. Syntactic information as retrieved by using the $TF-IDF$ measure in the second step is enriched with this semantic information of semantic term-dimension relatedness matrix thus giving a hybrid semantic service vector. Finally the cosine similarity between the semantic vector of the query and services is calculated in step 5 and if this matching degree is better than or equal to the minimum threshold specified by the user, then this service is considered as potentially relevant and returned to the user (Step 6) else it is dropped.

Input: Services dataset S , dimension set D , query $Query$, Minimum Threshold Value Min_T_Value .

Output: S Resultant service set with Similarity scores

Method:

1. Preprocess all the services to extract Input, Output, Service Name and description from each profile.
2. Generate $TF-IDF$ Term frequency-Inverse document frequency [29] matrix for the extracted terms.
3. Run K-Means clustering algorithm to group the services into k clusters $C_1, C_2, C_3, \dots, C_k$.
4. For every service $Service_j \in C_x$ in each cluster where $\{j: 1 \text{ to } |C_x| \text{ and } x: 1 \text{ to } k\}$
 - a. Calculate term dimension vector $TDV_i(Service_j)$ for each term T_i (extracted through the pre-processing phase) of the $Service_j$ using Omiotis Measure of Semantic Relatedness for each dimension $D_k \in D$ as:

$$TDV_i(Service_j, T_i, D) = [S(T_i, D_1), S(T_i, D_2), S(T_i, D_3), \dots, S(T_i, D_n)]$$
 - b. Integrate the syntactic and semantic values by multiplying the term dimension vector with term weightage $TF-IDF_i$ values of each term T_i to get semantic term dimension vectors:

$$STDV_i(Service_j) = TF-IDF_i * TDV_i(Service_j)$$
 - c. Calculate the sum of all the semantic term dimension vectors for all the terms T_i in the service $Service_j$

$$ASV(Service_j) = \sum_{\text{for } i=1}^{\text{all terms in the Service } j} STDV_i(Service_j)$$
5. For the given user query
 - a. Find out the appropriate cluster C_{Query} for the query.
 - b. Calculate term dimension vector $TDV_i(Query)$ for each term T_i of the $Query$ using Omiotis Measure of Semantic Relatedness with each dimension $D_k \in D$ as:

$$TDV_i(Query, T_i, D) = [S(T_i, D_1), S(T_i, D_2), S(T_i, D_3), \dots, S(T_i, D_n)]$$
 - c. Integrate the syntactic and semantic values by multiplying the term dimension vector with term weightage $TF-IDF_i$ values of each term T_i to get semantic term dimension vectors:

$$STDV_i(Query) = TF-IDF_i * TDV_i(Query)$$
 - d. Calculate the sum of semantic term dimension vectors for all the terms T_i in the Query

$$ASV(Query) = \sum_{\text{for } i=1}^{\text{all terms in the Query}} STDV_i(Query)$$
6. Calculate cosine similarity between query vector and all the services in the recommended cluster.

$$OSDSim(Service_j, Query) = \frac{ASV(Service_j) \cdot ASV(Query)}{\|ASV(Service_j)\| \cdot \|ASV(Query)\|}$$
7. Filter the set of services based on the threshold value so as that services which meet the user defined threshold are returned to the user only.

$$\text{If } OSDSim(Service_j, Query) \geq Min_T_Value$$

then add $Service_j$ to result set R else drop $Service_j$.
8. Return R i.e. Resultant service set for the query ranked based on their corresponding similarity scores.

Figure 2. Proposed OSD Algorithm

3.3. Hybrid Semantic Interpreter

The Hybrid Semantic Interpreter mainly performs preprocessing, clustering, Service-Dimension semantic relatedness matrix calculation and Integration of syntactic information with semantic information. Hybrid Semantic Interpreter plays a vital role in the service discovery process as it integrates and transforms the service profiles into semantically enriched vectors. The following subsections elaborate the functions performed by the Hybrid Semantic Interpreter.

3.3.1. Text Data Mining Pre-processing for Web Services Descriptions:

The process of extracting the relevant, non-trivial and quality information from the text is known as Text data mining or intelligent text analysis [33]. Our approach employs the text data mining for extracting the relevant terms from the Web Service descriptions. Two services meant for a particular task may have syntactically different interfaces. To resolve this kind of problems, the service descriptions need to be pre-processed. In our approach the Functional parameters like input, output, service names and service description are extracted from each service document. Pre-processing includes following steps

1. Tokens are generated using the string tokenizer.
2. Combined terms are split down. For *e.g.*, “getCountryWeather” to “get”, “Country” and “Weather”.
3. Stop words are removed from the file. We used some English stop words like as, and, when, being, any etc. and a list of stop words related to web services like get, post, soap, request etc.
4. *TF-IDF* [29] matrix is generated.

3.3.2. Clustering of Services

In this step, the services in the dataset are grouped together based upon their similarities using Euclidean distance function into k different clusters *i.e.*, $C_1, C_2, C_3, \dots, C_k$. We have used K-Means algorithm to filter the non relevant services during the service discovery. Given a user query, the appropriate cluster C_{Query} is found, where C_{Query} can be any one cluster from $C_1, C_2, C_3, \dots, C_k$ and all the services within the cluster C_{Query} are matched with the user query by using their Aggregate Semantic Vectors *i.e.*, *ASV* vectors. For each service in the cluster the semantic vector is generated using Omiotis MSR and the Cosine similarity between the query and all the Service(s) vector(s) of the matching cluster of the query is calculated to get the relevant set of services for query. Therefore, by using the clustering approach, a user query need not to be matched with all the services in a large service set and only the related services within the same cluster are considered for the relevance and query is matched with the services present within the recommended cluster that keeps the cost overhead of matching the user query with all the services aside. This helps in filtering non relevant services at a very early stage, thereby improving the efficiency of the discovery process in case of large service set.

3.3.3. Term-Dimension Semantic Relatedness Matrix Calculation

In this phase, we calculate the Semantic relatedness score between some predefined dimensions and the terms of the services present in k clusters *i.e.*, $C_1, C_2, C_3, \dots, C_k$. For each extracted term T_i from Input, Output, Service name and service description of the web service description file through the pre-processing phase, whose *TF-IDF* weight is non zero; semantic relatedness value between the term T_i and each dimension is calculated. We have used eight different dimensions in order to convert a service into

semantic vector based form. Here the Semantic dimension space is explicitly defined by a set of dimension vector $D = \{D_1, D_2, D_3, \dots, D_n\}$, where each term represents a single dimension. A service may not always belong to a single dimension. Partially, it may represent its relation to different dimensions. Keeping this in view, semantic relatedness of a particular service is calculated with respect to different dimensions using the Omiotis Measure of Semantic Relatedness. These dimensions have been explicitly chosen from different specific domains. For each term T_i of the web service $Service_j$, the individual Term Dimension Vector TDV_i is calculated as:

$$TDV_i (Service_j, T_i, D) = [S(T_i, D_1), S(T_i, D_2), S(T_i, D_3), \dots, S(T_i, D_n)] \quad (2)$$

Similarly, Term Dimension Vectors for all the terms extracted from the pre-processing phase for service $Service_j$ are calculated to form a term dimension matrix. Like-wise, the term dimension matrices for all the services in the dataset are calculated and are passed to the semantic-syntactic integrator.

3.3.4. Semantic Syntactic Knowledge Integration

In this phase the semantic knowledge of the service is integrated with the syntactic knowledge. This includes combining the Term Dimension Matrices with the corresponding $TF-IDF$ vectors. To achieve this combination TDV_i vector for term T_i of service $Service_j$ is multiplied with the respective $TF-IDF_i$ value of the term T_i so as to get the Semantic Integrated Term Dimension Vector $STDV_i$.

$$STDV_i = TF-IDF_i * TDV_i \quad (3)$$

Here, the scalar multiplication has been used, where the $TF-IDF_i$ value of each term is multiplied with the TDV_i vector of that term. The $STDV_i$ vector gives integrated scores for single term T_i only. Similarly, Semantic Integrated Term Dimension Vectors for all other terms of service $Service_j$ are calculated by multiplying with the corresponding $TF-IDF_i$ values for all the terms. Once the $STDV_i$ vectors for all the services have been calculated, there arises a need for a cumulative vector which can describe the service as a single vector which can be used for the finding similarity scores with other services. This need is resolved with the help of $ASV (Service_j)$ vector. The $ASV (Service_j)$ vector represents the sum of all the Semantic Integrated Term Dimension Vectors of all the terms of $Service_j$. This gives the combined semantic meaning of the service $Service_j$ and can be attained by using (4) below.

$$ASV (Service_j) = \sum_{\substack{\text{all terms in the } \\ \text{Service } j \\ \text{for } i=1}} STDV_i \quad (4)$$

Similarly, the $ASV (Service)$ vectors for all the services present in all the clusters $C_1, C_2, C_3, \dots, C_k$ are calculated and these are passed to the Similarity Match Engine. These $ASV (Service)$ vectors forms a Semantic collection of service vectors which will be used for satisfying the user queries.

3.3.5. Generation of Query Vector

Whenever some query is received, the query is preprocessed for extracting out the important words and the appropriate cluster C_{Query} for the query is determined. Further, the semantic relatedness value between the terms of the query and the dimension vector

$D = \{D_1, D_2, D_3, \dots, D_n\}$ is calculated using the Omiotis measure of semantic relatedness to generate Term Dimension Vectors for all the terms of the query $TDV_i(Query)$.

$$TDV_i(Query, T_i, D) = [S(T_i, D_1), S(T_i, D_2), S(T_i, D_3), \dots, S(T_i, D_n)] \quad (5)$$

These $TDV_i(Query)$ vectors are merged with the syntactic information to form the Semantic Integrated Term Dimension query Vector $STDV_i(Query)$. The $STDV_i(Query)$ is calculated by multiplying the term dimension vector $TDV_i(Query)$ for a term T_i with its respective $TF-IDF_i$ value.

$$STDV_i(Query) = TF-IDF_i * TDV_i(Query) \quad (6)$$

$ASV(Query)$ is calculated by adding all the Semantic Integrated Term Dimension Vectors, for all the extracted terms of the query. $ASV(Query) = \sum_{for\ i=1}^{all\ terms\ in\ the\ Query} STDV_i(Query)$ (7)

The computed $ASV(Query)$ vector is passed to the Similarity Match engine where $ASV(Query)$ is matched with all the $ASV(Service)$ vectors in the C_{Query} cluster to get the most relevant service results.

3.4. Similarity Match Engine

Finally Cosine measure is used for calculating the semantic similarity between semantic service dimension vector $ASV(Service)$ of all the services in the cluster C_{Query} and the semantic query dimension vector $ASV(Query)$.

$$OSDSim(Service_j, Query) = \frac{ASV(Service_j) \cdot ASV(Query)}{\|ASV(Service_j)\| \cdot \|ASV(Query)\|} \quad (8)$$

In Eq. (8), the cosine of the angle between both vectors tells about the similarity between service $Service_j$ and query. Service $Service_j$ and the query will be more similar if the Cosine angle between query and service vector is small and cosine value is more close to 1. Cosine similarity of 1 indicates that the service provides exactly the same functionality as desired by the user through query. If the cosine similarity between the service and the query is greater than the minimum specified threshold value, then the service will be returned to the user as a resultant set else it will be dropped.

4. Empirical Evaluation and Results

4.1. Dataset and Tools Used

We implemented our approach on web service profiles written in two different description languages *i.e.*, OWL-S. For implementing our approach for OWL-S based services, we used some of the OWL-S services from the OWLS-TC V2 service test collection. This collection contains services which are retrieved mainly from public IBM UDDI registries, and are semi-automatically transformed from WSDL to OWL-S. The collection OWLS-TC V2 is available as open source [34]. Out of this service collection, we used selectively chosen 124 services as dataset 1. These 124 services covered nine application domains that are Film, Book, Country, Hotel, Automobile, Hospital, Missile, Surfing, and University.

We used Rapid Miner, Matlab and some other toolkits for statistical applications. For calculating semantic relatedness values, we used the freely available Omiotis MSR [35].

4.2. Evaluation Measures for Service Discovery Approach

To evaluate our proposed approach we have used three performance measures commonly used in information retrieval. Precision, Recall and F-Measure are three widely

used performance measures for measurement of search efficiency. Given a query q , precision P is the proportion of the relevant documents retrieved to all the retrieved documents, and is described as

$$P = \frac{|\text{relevant_documents} \cap \text{retrieved_documents}|}{|\text{retrieved_documents}|} \quad (11)$$

and recall is the proportion of relevant documents which have been retrieved to all the relevant documents, and is described as

$$R = \frac{|\text{relevant_documents} \cap \text{retrieved_documents}|}{|\text{relevant_documents}|} \quad (12)$$

Finally we have calculated and compared the F-Measure score. The F-Measure which is the harmonic sum of Precision and recall gives the accuracy of the approach and is described as:

$$F = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (13)$$

4.3. Discovering OWLS-S based Services using OSD Approach:

For evaluating service discovery using OSD approach, 124 services of the Dataset 1 from nine different domains were selected and pre-processed. The combined words were split down and stop words as given in the file “stopwords.txt” were filtered out. A total of 235 regular attributes were extracted from the pre-processing phase. Further, the $TF-IDF$ values were generated for all the terms within a document, thereby forming a $TF-IDF$ matrix (124,235). Further, we have used clustering to filter out the non relevant services. Although, in past also many researchers have used clustering to filter out the services at first stage, keeping in view the related advantages we have also used the same. Our dataset are tested on two clustering algorithms with different evaluation measures to know which algorithm will give better clustering solution.

K-Means performed better than the K-Medoids at each value of k . The value of this index was found maximum in the range for $k = 8$ and therefore, based upon the evaluation comparison between the two clustering algorithms, we selected K-Means with value of $k = 8$. After clustering the data into eight groups, services within each cluster are transformed into semantic service vectors. The Term dimension vectors were calculated for each term of the all the service in the cluster. The semantic relatedness value between all the terms of a service and the dimension vector was calculated. The dimension vector constitutes the following nine domains:

$D = \{\text{“Film”}, \text{“Book”}, \text{“Country”}, \text{“Hotel”}, \text{“Automobile”}, \text{“Hospital”}, \text{“Missile”}, \text{“Surfing”}, \text{“University”}\}$

Implementation of the proposed approach using some of the scenarios is presented below:

4.3.1. Scenario 1:

The client aims to know about hotel in a city of a country. He provides the query as citycountry_hotel_service.owl of the owls-tc2_2_rev_2 test set [41]. The relevance set for the query is taken from OWLS-MX [12] project dataset [41] and depicted in Table 4. The columns of Table 1 represents the Sid (Service id), set of the relevant services, and their IR similarity values, *i.e.*, the Cosine (Cos) similarity, Extended Jaccard (EJ) similarity value and Jensen Shannon (JS) metrics similarity value, with their semantic degree of match and hybrid degree of match respectively [41].

Table 1. Relevance Set for Query Travel/ citycountry_hotel_service.owl

| Sid | Relevance set | Cos | EJ | JS | Semantic degree match | Hybrid degree of match |
|-----|--|------|------|------|-----------------------|------------------------|
| 1 | countrycity_hotel_service.owl | 0.95 | 0.8 | 0.97 | Exact | Exact |
| 2 | country_hotel_service.owl | 0.95 | 0.8 | 0.97 | Exact | Exact |
| 3 | citycountry_hotel_service.owl | 0.95 | 0.8 | 0.97 | Exact | Exact |
| 4 | city_hotel_service.owl | 0.93 | 0.89 | 0.98 | Exact | Exact |
| 5 | city_hotel_Germanservice.owl | 0.93 | 0.89 | 0.98 | Exact | Exact |
| 6 | hotel_Worldwideservice.owl | 0.5 | 0.5 | 0.5 | Exact | Exact |
| 7 | geopolitical-entity_hotel_service.owl | 0.96 | 0.75 | 0.96 | Plugin | Plugin |
| 8 | geographical-region_hotel_service.owl | 0.88 | 0.64 | 0.9 | Plugin | Plugin |
| 9 | city_luxuryhotel_service.owl | 0.78 | 0.64 | 0.89 | Plugin | Plugin |
| 10 | luxuryhotel_Heidelburgservice.owl | 0.35 | 0.25 | 0.4 | Plugin | Plugin |
| 11 | city_accommodation_service.owl | 0.78 | 0.62 | 0.83 | Subsumed-by | Subsumed-by |
| 12 | countrycapital-city_hotel_service.owl | 0.92 | 0.87 | 0.99 | Failed | NN |
| 13 | citycountry_destinationhotel_service.owl | 0.91 | 0.87 | 0.96 | Failed | NN |
| 14 | countrycity_sportshotel_service.owl | 0.91 | 0.87 | 0.96 | Failed | NN |
| 15 | durationcountrycity_hotel_service.owl | 0.88 | 0.79 | 0.93 | Failed | NN |
| 16 | towncountry_hotel_service.owl | 0.85 | 0.77 | 0.95 | Failed | NN |
| 17 | countryvillage_hotel_service.owl | 0.83 | 0.75 | 0.95 | Failed | NN |
| 18 | village_hotel_service.owl | 0.83 | 0.75 | 0.95 | Failed | NN |
| 19 | city_bedandbreakfast_service.owl | 0.62 | 0.5 | 0.73 | Failed | NN |
| 20 | city_hotel_Saarlandservice.owl | 0.56 | 0.54 | 0.55 | Failed | NN |
| 21 | citycountry_accommodation_service.owl | - | - | - | Failed | Failed |
| 22 | countrycity_luxuryhotel_service.owl | - | - | - | Failed | Failed |
| 23 | countrycity_luxuryhotel_Gelservice.owl | - | - | - | Failed | Failed |

```

<profile:Profile rdf:ID="CITYCOUNTRY_ACCOMMODATION_PROFILE">
<service:isPresentedBy rdf:resource="#CITYCOUNTRY_ACCOMMODATION_SERVICE"/>
<profile:serviceName xml:lang="en">
AccommodationInfoService
</profile:serviceName>
<profile:textDescription xml:lang="en">
This service returns accomodation information, hotel, restaurant etc
in the city of the country.
</profile:textDescription>
<profile:hasInput rdf:resource="#_COUNTRY"/>
<profile:hasInput rdf:resource="#_CITY"/>
<profile:hasOutput rdf:resource="#_ACCOMMODATION"/>
<profile:has_process rdf:resource="#CITYCOUNTRY_ACCOMMODATION_PROCESS" /></profile:Profile>
<process:ProcessModel rdf:ID="CITYCOUNTRY_ACCOMMODATION_PROCESS_MODEL">
<service:describes rdf:resource="#CITYCOUNTRY_ACCOMMODATION_SERVICE"/>
<process:hasProcess rdf:resource="#CITYCOUNTRY_ACCOMMODATION_PROCESS"/>
</process:ProcessModel>
<process:AtomicProcess rdf:ID="CITYCOUNTRY_ACCOMMODATION_PROCESS">
<process:hasInput rdf:resource="#_COUNTRY"/>
<process:hasInput rdf:resource="#_CITY"/>
<process:hasOutput rdf:resource="#_ACCOMMODATION"/>
</process:AtomicProcess>

```

Figure 3. OWL-S Profile for citycountry_accommodation_service.owl Service

If we look at services at Sr. No. 21, 22, 23 of Table 1 then according to OWLS-MX [12] relevance set the semantic degree of match as well as degree of match is shown to be FAILED. Moreover the syntactic match score with cosine similarity, Jaccard and extended Jaccard approach is also not specified. Figure 3 depicts the profile of the service at Sr. No. 21 *i.e.*, `citycountry_accommodation_service.owl`s from the OWL-TC testset [46]. Similarly, the data from the various tags like `<profile:hasInput>`, `<profile:hasOutput>` and `<profile:textDescription>` of the profiles of the query and services at Sr. No. 22, 23 were extracted out.

Table 2. Service Names, Input, Output and Description of the Query and Services whose Matches were shown to be FAILED by OWL-S Matchmaker

| SR. NO. | Service Name | INPUT | OUTPUT | Description |
|---------|---|---------------|---------------|---|
| Query | <code>citycountry_hotel_service.owl</code> s | Country, City | Hotel | It is a reliable service to return hotel of the city of the country |
| 21 | <code>citycountry_accommodation_service.owl</code> s | Country, City | Accommodation | This service returns accommodation information, hotel, restaurant etc in the city of the country. |
| 22 | <code>countrycity_luxuryhotel_service.owl</code> s | Country, City | LuxuryHotel | This service is a recommended service to know about the luxury hotel of the city of the country. |
| 23 | <code>countrycity_luxuryhotel_Gelservice.owl</code> s | Country, City | LuxuryHotel | Gel is a well known name in hotels, and provide a service to know about the luxury hotel |

We observed the profiles of services at Sr. No. 21, 22, 23 and compared the input, output and description of the these three services with the above said query in Table 2. After comparison it was found that the user request is highly related and can be satisfied by these services. The concept accommodation of service at Sr. No. 21 is semantically related with the concept hotel of the user query and moreover description field of the profile clearly states that this service provide the information about hotel, restaurant in the city of the country. Similarly services at sr. No. 22 and 23 can meet the user requirements. But OWLS-MX [12] relevance set shows the match to be FAILED.

In case of OWLS-MX matchmaker only the input and output concepts were matched within the mutually agreed upon ontology which is not always feasible. Many-a-times the input and output parameter cannot provide the complete functionality of the services. The proposed approach was applied on the same query and the semantic relatedness of the query was calculated with the services. Table 3 shows the *ASV (Service)* vector for the matching services. Table 4 shows the resultant similarity score of query with services.

Table 3. ASV Vectors of Matching Services

| Service No. | Film | Book | Country | Hotel | Automobile | Hospital | Missile | Surfing | University |
|-------------|-----------|----------|----------|----------|-------------|----------|----------|----------|------------|
| 94 | 0.0001335 | 0.122561 | 0.378125 | 0.191849 | 0.005130735 | 0.00262 | 0.000424 | 9.35E-05 | 0.0064268 |
| 95 | 0.000797 | 0.014923 | 0.403917 | 0.232884 | 0.012639317 | 0.006379 | 0.001344 | 0.001866 | 0.0080611 |
| 96 | 0.0003505 | 0.020134 | 0.018931 | 0.629458 | 0.008183388 | 0.006536 | 4.48E-05 | 2.46E-05 | 0.0108352 |
| 97 | 0.0002821 | 0.1918 | 0.470843 | 0.329242 | 0.004391917 | 0.003437 | 0.000342 | 1.12E-05 | 0.0090319 |
| 98 | 8.82E-05 | 0.000135 | 0.481569 | 0.503223 | 0.006647514 | 0.005239 | 0.000356 | 1.68E-05 | 0.0087169 |
| 99 | 0.0029769 | 0.001137 | 0.36012 | 0.544846 | 0.012235524 | 0.006215 | 0.001466 | 0.001406 | 0.0078215 |
| 100 | 5.24E-05 | 0.000126 | 0.498616 | 0.260772 | 0.013631134 | 0.002732 | 0.000366 | 3.44E-05 | 0.0086817 |
| 101 | 0.0022514 | 0.016784 | 0.386922 | 0.404054 | 0.016711422 | 0.004569 | 0.000432 | 0.010465 | 0.0080404 |
| 102 | 0.0002188 | 0.012079 | 0.345093 | 0.369168 | 0.004961455 | 0.003837 | 0.000263 | 1.46E-05 | 0.0007099 |
| 103 | 1.94E-05 | 0.000286 | 0.429701 | 0.00019 | 0.000119623 | 4.69E-05 | 0.000304 | 1.29E-05 | 7.28E-05 |
| 104 | 0.0003889 | 0.024237 | 0.470847 | 0.507534 | 0.006731483 | 0.005272 | 0.000343 | 2.25E-05 | 0.0007349 |

After matching the query with services it was found that the services at Sr. No. 21, 22, and 23 of relevance set in Table 2 were semantically highly related to the query with a semantic score of 0.959399057, 0.947630255 and 0.983742553 respectively. If we look at the service name input, output and description of the query and these services, then also the results seems to be justifiable. Therefore our approach is able to find the semantic relation for queries which were otherwise shown to be FAILED by the OWL-S matchmaker.

Table 4. Results of Proposed OSD Approach for the Query

| Service No. | Service Name | OSD Similarity |
|-------------|---|--------------------|
| 98 | countrycity_hotel_service.owl | 1 |
| 102 | countryvillage_hotel_service.owl | 0.999582515 |
| 104 | country_hotel_service.owl | 0.999199386 |
| 101 | countrycity_sportshotel_service.owl | 0.99916543 |
| 99 | countrycity_luxuryhotel_Gelservice.owl | 0.983742553 |
| 95 | citycountry_accommodation_service.owl | 0.959399057 |
| 100 | countrycity_luxuryhotel_service.owl | 0.947630255 |
| 97 | countrycapital-city_hotel_service.owl | 0.930108788 |
| 94 | citycountryduration_HotelReserveservice.owl | 0.906491219 |
| 96 | city_hotel_service.owl | 0.742299395 |

We performed a second experiment to see the importance of text description and semantics matching for a service in its intelligent discovery. By utilizing the description of the profile one can come to know the specific functionalities of the service represented in natural language. We modified the service profiles of some of the services to add more semantically similar words in the Dataset 1. A service requestor specifies the query as Input concepts, output concepts and service description in a certain description language or a set of keywords. The concepts thus used may be different than those used in services but they may be semantically correlated concepts like cost and price; car and vehicle; vehicle and automobile; car and automobile; bike and motorcycle; hotel and accommodation; sedan, luxury car and BMW or Mercedes *etc.*

4.3.2. Scenario 2:

Suppose a service requestor, specify the search for a query to know the cost of automobiles; where most of the service advertisements in the automobile domain vary in concepts like cost or prices of vehicles, automobiles, motorcycle, car and bike. The concept automobile is semantically similar to other concepts like vehicle, car, motorcycle and bike. Similarly, the cost concept is also semantically related to cost. For the above mentioned query we applied some statistical approaches like $TF-IDF$ cosine similarity, Jaccard similarity and semantics based approaches like Latent Semantic Analysis (LSA) to the dataset to get the resultant set of services. Finally, we compared our proposed OSD approach with the before mentioned methods. Table 5 gives a comparative statement of the result of applying different methods like $TF-IDF$ cosine, Jaccard, LSA, and OSD to the query for discovering semantic Web services. The bold entries in the Table(s) indicate the list of resultant service set matching the query with similarity value above the threshold value. Threshold value of 0.5 is used for filtering the semantic value of match.

For the given query, the Jaccard similarity and $TF-IDF$ cosine similarity approach found only one and three suitable services respectively. The results generated from the syntactic based $TF-IDF$ Cosine similarity and Jaccard similarity measures did not incorporate the semantically related services that may satisfy the user request. Although,

there were many services which semantically matched with the query profile but they were not present in the resultant set. Based on human judgement also the services semantically shared high degree of similarity with the query but not present. Further, LSA was used over the same dataset with different dimensions 10, 20,30,40,50 and 60 to get the services related with the query. By using LSA with 10 dimensions the number of services in the resultant service set improved over the Jaccard similarity and $TF-IDF$ cosine similarity approach. The results varied with different dimensions. We further increased the dimensions of the LSA approach from 10 to 60, by increasing the dimensions in LSA the precision was improving where as recall value was decreasing. We applied OSD approach on the same query and OSD resultant set were found to be better than the results from the other approaches.

Similarly, seven queries were run to find the semantically related set of services. The results were compared with the relevance set to know about the relevant and non relevant services and hence for calculating precision and recall values. The results clearly indicates that OSD approach outperforms the keyword based $TF-IDF$ Cosine similarity method, Jaccard similarity method and LSA with different dimensions. Entries in the Table 5 depicts the comparison of the performance between the proposed OSD approach and LSA approach in terms of Macro average precision, Macro average recall and Macro average F-Measure and clearly shows that better experimental results were achieved with the use of proposed hybrid OSD approach.

Table 5. Comparison of Proposed OSD Approach on OWL-S Services with LSA

| Method | LSA 10 dimension | Proposed OSD approach |
|-------------------------|------------------|-----------------------|
| Macro average precision | 92.8277 | 93.2773 |
| Macro average recall | 88.6543 | 94.4477 |
| Macro average F-Measure | 89.3071 | 93.6508 |

As compared with LSA, by using the proposed approach the Macro average precision and Macro average recall has increased by 0.45 % and 5.79% respectively. An improvement of 4.35 % in terms of Macro average F-Measure has been observed. We implemented the LSA approach for various dimensions from 10 to 60. At dimension 60, recall value was observed to be minimum *i.e.*, 28.2697 at the precision 1. From the experimental analysis, we conclude that the OSD approach giver better and more relevant results than the LSA approach and other methods like $TF-IDF$ cosine similarity and Jaccard similarity based approaches.

4.5. Discussion

Our approach integrates the syntactic as well as semantic information of a web service which is used in the quantification of similarity, *i.e.*, generating a concrete similarity value, which would be useful for ranking services. Secondly, our proposed framework does not make any assumptions about the description language of the advertised Web Service. In effect, a published Web service could be described in any Web Service description language such as WSDL or semantic ontology based OWL-S and WSML etc. We implemented our approach on OWL-S based service profiles for two different description languages. The results have shown the feasibility of the approach. Thirdly, the service requests can be formulated in any of the above mentioned standards or even as a set of simple keywords. Millions of ontologies available over the internet, containing thousands of concepts and instances make the selection, discovery and matching of a given corresponding concept within real world ontology very time consuming task. Finally, to overcome the ontology heterogeneity problem, our proposed framework

employs some Natural Language Processing (NLP) techniques to extract most relevant senses of the terms used in the description of Web services. Creation, maintenance and handling of ontologies are hard and require expertise. There is no need to transform all the existing services described in WSDL to other standards. With the use of Omiotis the best matching senses of the concepts are used for similarity calculation. Our method tries its best to utilize the hidden semantics within the existing services. Most significant is the ease of use, even a naive user can use the services without learning the technical know-how of the underlying technologies. User need not to learn the complex languages, frameworks and underlying ontologies. Therefore, by overcoming some of the limitations of the current approaches, our proposed approach puts efforts in a direction to improve the scenario of discovering the relevant Web services.

5. Conclusion and Future Work

In this research, we proposed a semantic Web Service Discovery Framework based on Hybridization. The method consists of a combination of Measure of semantic relatedness, machine learning, text mining and other statistical techniques in order to intelligently discover the most related services for a given request. As per best of our knowledge usage of Omiotis in the web service discovery is novel and has not been used by any other author in the same area. The proposed approach was tested on OWL-S service. From the experimental analysis we conclude that OSD approach performs better than the LSA and other above mentioned keyword based approaches like $TF-IDF$ cosine similarity method, Jaccard similarity. As compared with LSA, by using the proposed approach the precision and recall has increased by 0.45 % and 5.79% respectively for OWL-S services. It is also worth mentioning here that although by increasing the number of dimension in case of LSA dimensionality reduction, the precision improves impressively but it has very low total recall and hence total reduced F-Measure. The overall F-Measure using the OSD approach is 4.35 % higher than the LSA approach, which shows that OSD can help in the computation of more accurate relatedness values and hence better discovery of services. Our future work will focus on extending this approach for composite Web services.

References

- [1] S. Robak and B. Franczyk, "Modeling Web services variability with Feature diagrams", Proceedings of Web services and database systems, (2002), pp. 20-128.
- [2] E. Dashofy, M. van der Hoek and A. Taylor, "A highly-extensible, XML-based architecture description language Software Architecture", Proceedings of Working IEEE/IFIP Conference, Amsterdam, Netherlands, (2001).
- [3] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi and S. Weerawarana, "Unraveling the Web services, Web: An introduction to SOAP, WSDL, and UDDI", IEEE Internet Computing, vol. 6, no. 2, (2002) April.
- [4] World Wide Web Consortium (W3C), Web services Description Language (WSDL) 1.1., 2001. <http://www.w3.org/TR/wSDL/>.
- [5] UDDI Technical White Paper, 2000. <http://www.uddi.org/pubs/Iru-UDDI-Technical-White-Paper.pdf>.
- [6] N. Shadbolt, T. B. Lee and W. Hall, "The Semantic web revisited, IEEE Intelligent Systems", vol. 21, no. 3, (2006), pp. 96-101.
- [7] D. Martin, M. Burstein, J. Hobbs, O. Lassila and K. Sycara, "OWL-S: Semantic markup for Web services", <http://www.w3.org/submission/OWL-S/>, (2004).
- [8] D. Fensel, H. Lausen, A. Pollers, J. Bruijn, M. Stollberg, D. Roman and J. Domingue, "Enabling Semantic Web services: The web service modeling Ontology", Springer Verlag, ISBN 978-3-540-34520-6, (2007), pp. 38-61.
- [9] R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, T. Schmidt, A. Sheth and K. Verma, "Web service semantics-WSDL-S", <http://www.w3.org/submission/WSDL-S/>, (2005).
- [10] J. Farrell and H. Lausen, "Semantic Annotations for WSDL and XML Schema", <http://www.w3.org/TR/SAWSDL/>, (2007).
- [11] C. Kiefer and A. Bernstein, "The creation and evaluation of ISPARQL strategies for matchmaking", Proceeding of Fifth European Semantic Web Conference (ESWC), volume- 5021, LNCS, Springer Verlag Berlin Heidelberg, (2008), pp. 463-477.

- [12] M. Klusch, B. Fries and K. P. Sycara, "OWL-MX: A Hybrid semantic Web Service matchmaker for OWL-S services", *Web Semantics*, vol. 7, no. 2, (2009), pp. 121-133.
- [13] M. Klusch and F. Kaufer, "Wsmo-mx: A hybrid semantic web service matchmaker", *Web Intelligence and Agent Systems*, vol. 7, no. 1, (2009), pp. pp. 23-42.
- [14] D. Wei, T. Wang, J. Wang and A. Bernstein, "SAWSDL-iMatcher: A customizable and effective Semantic Web Service Matchmaker", *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 9, no. 4, (2011), pp. 402-417.
- [15] G. Tsatsaronis, I. Varlamis and M. Vazirgiannis, "Text relatedness based on a word thesaurus", *Artificial Intelligence Research*, vol. 37, (2010), pp. 1-39.
- [16] J. A. Nasir, A. Karim, G. Tsatsaronis and I. Varlamis, "A Knowledge-based Semantic Kernel for Text Classification", *Proceedings of the 18th international conference on String processing and information retrieval*, (2011), pp. 261-266.
- [17] M. Paolucci, T. Kawamura, T. R. Payne and K. P. Sycara, "Semantic matching of Web services capabilities", Horrocks, I., Hendler, J. (eds.) *ISWC, LNCS, Springer, Heidelberg*, vol. 2342, (2002), pp. 333-347.
- [18] D. Bruijn, R. Lara, A. Polleres and D. Fensel, "OWL- DL vs. OWL light: conceptual modeling and reasoning for the semantic Web", *Proceeding of WWW conference, Chiba, Japan*, (2005).
- [19] T. Kawamura, J. A. D Blasio, T. Hasegawa, M. Paolucci and K. Sycara, "Public deployment of semantic Service Matchmaker with UDDI business Registry", *LNCS, Springer, Heildelberg*, vol. 3298, (2004), pp. 752-766.
- [20] R. Gronmo and M. C. Jaeger, "Model-Driven Semantic Web Service Composition", *APSEC, December (2005)*, pp. 79-86.
- [21] L. Li and I. Horrocks, "A software framework for matchmaking based on Semantic Web technology", *Proc. of WWW, Budapest*, (2003) May, pp. 331-339.
- [22] H. Wang and Z. Li, "A Semantic Matchmaking Method of Web services Based On SHOIN(D)*", *Proc. of IEEE APSCC, Guangzhou, China*, (2006) December, pp. 26-33.
- [23] K. Sycara, S. Widoff, M. Clutch and J. Lu, "LARKS: Dynamic Matchmaking among Heterogenous Software Agents in Cybersace", *Autonomous Agents and Multi agents Systems*, vol. 5, (2002), pp. 173-203.
- [24] K. Verma, K. Sivashanmugam, A. Sheth, A. Patil, S. Oundhakar and J. Miller, "METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web services", *Information Technology and Management*, vol. 6, no. 1, (2005), pp. 17-39.
- [25] R. Nayak and B. Lee, "Web Service Discovery with additional Semantics and Clustering", *Proceeding of IEEE/WIC/ACM International Conference on Web Intelligence*, (2007).
- [26] S. Batra and S. Bawa, "Semantic discovery of Web services using principal component analysis", *Physical Sciences*, vol. 6, no. 18, (2011), pp. 4466-4472.
- [27] T. Wen, G. Sheng, Y. Li and Q. Guo, "Research on Web Service Discovery with Semantics and Clustering", *Proceedings of IEEE sixth joint international Conference on Information technology and Artificial Intelligence (ITIAC)*, (2011), pp. 62-67.
- [28] J. M. García, D. Ruiz and A. Cortes, "Improving semantic Web services discovery using SPARQL-based repository filtering", *Web Semantics: Science, Services and Agents on the World Wide Web*, Elsevier, (2012), pp. 12-24.
- [29] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval", *Information processing and Management*, vol. 24, no. 5, (1988), pp. 513-523.
- [30] M. Senvar and A. Bener, "Matchmaking of Semantic Web services Using Semantic-Distance Information", *ADVIS, LNCS, Springer*, vol. 4243, (2006), pp. 177-186.
- [31] L. Chen, Z. Song, Y. Zhang and Z. Miao, "Wordnet Enhanced Dynamic Semantic Web services Discovery", *Emerging Research in Artificial Intelligence and Computational Intelligence, CCIS*, vol. 237, (2011), pp. 529-536.
- [32] A. Grintsvayg, V. D. Veksler, R. Lindsay and W. D. Gray, "Vector generation of explicitly defined multidimensional semantic space", *Proceedings of ICCM Eight international conference on Cognitive Modeling, Oxford, UK*, (2007), pp. 231-232.
- [33] M. Crasso, A. Zunino and M. Campo, "AWSC: An approach to Web services classification based on machine learning techniques", *CONICET, Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, vol. 12, no. 37, (2008), pp. 25-36.
- [34] <http://projects.semwebcentral.org/projects/owls-tc/>.
- [35] <http://Omiotis.hua.gr/WebSite/wsinfo.html>.