Hybrid Web Service Selection by Combining Semantic and Keyword Approaches

Florie Ismaili
South East European University
Faculty of Contemporary Sciences and Technologies
Ilindenska nn. Tetovo, FYROM
f.ismaili@seeu.edu.mk

***Abstract*— The challenge of Web service discovery increases with the remarkable raise of available Web services. The lack of semantic description in current keyword based service search makes it difficult for clients to find a required Web service. The pure semantic search restricts the types of queries users can perform.**

In order to solve these problems, in this paper we suggest a new hybrid Web service discovery architecture, which combines the ability to query and reason on metadata of semantic based search, and the flexibility of syntactic based search.

In the end, experiments are conducted to further demonstrate the feasibility of the proposed matching approach and its efficiency, regarded as the most promising way to improve the recall rate and precision rate.

***Index Terms*—Web services, hybrid search, semantic similarity, service discovery.**

I. Introduction

Service-Oriented Architecture is believed to be an important paradigm for efficient businesses application development. Within SOA, software components that provide a piece of functionality and communicate with each other via message they exchange are called services. Web Services are autonomous and modular applications deployed and invoked over Internet [1, 11].With the increasing number of available Web services, discovery of correct web service according to the user needs has been widely recognized as one of the most challenging problems in the application of Service Oriented Architecture [1].

Traditional proposals for service discovery typically focus on syntactical matching of service name and other attributes, which returns a large number of irrelevant services. This gap has motivated a lot existing research effort towards the Semantic Web services. In order to solve this problem, semantic meta-data is attached to web service descriptions which are mostly based on OWL-S [2], where the semantic of a web service are described in terms of inputs, outputs, preconditions and results, while service matching can be considered as ontological concepts matching.

Syntactic search, are known to suffer in general from low precision while being good at recall, while semantics-based approaches, in general, allow to reach a higher precision but lower recall [3].

The objective of the proposed hybrid Web service discovery approach is to improve semantic service retrieval performance by combining the ability to query and reason on metadata of semantic based search and the flexibility of syntactic based search.

The main contributions of the proposed approach are the following:

- Introducing a novel approach for efficiently finding Web services on the Web.
- A paradigm for uniting the diverse standards of XML-based Web technologies like XML, Web services and the Semantic Web.
- Allowing users to define customized matchmaking strategies and using similarity measures to improve the matchmaking performance.
- Collecting, analyzing and running several experiments on a large dataset consisting of real world Web services.

The rest of the paper is organized as follows:

In section 2 an overview of proposed Hybrid Web service discovery approach is presented. Section 3 discusses the keyword based searching. In section 4 the strict ontology based searching is presented. Section 5 introduces the hybrid searching followed by experimental evaluation in section 6. Section 7 concludes the paper.

II. Overview of Hybrid Web Service discovery Approach

Hybrid Web Service discovery approach supports keyword based searching, ontology based searching and hybrid searching by combining both keyword and semantic searching.

The following is an outline of the key steps of the proposed approach:

- A local register server is created which stores WSDL and OWL-S files of web services collected.
- The GHSOM is applied to the service set in order to construct self organizing maps which will be used for

SVD [5] matrix construction. This division is done in order to reduce the computational cost of directly applying SVD to the huge number of services.

- Using LSI [5] for scoring and ranking the documents by their relevance.
- Using SemWeb owl-s api [7], for mapping OWL-S documents into Ontology mapping storage in order to find relevant services to the query.
- Calculate syntactic based similarity of Web services.
- Calculate semantic based similarity of Web services.
- Compose syntactic and semantic similarity.
- Sort and return the list of relevant services to the user.

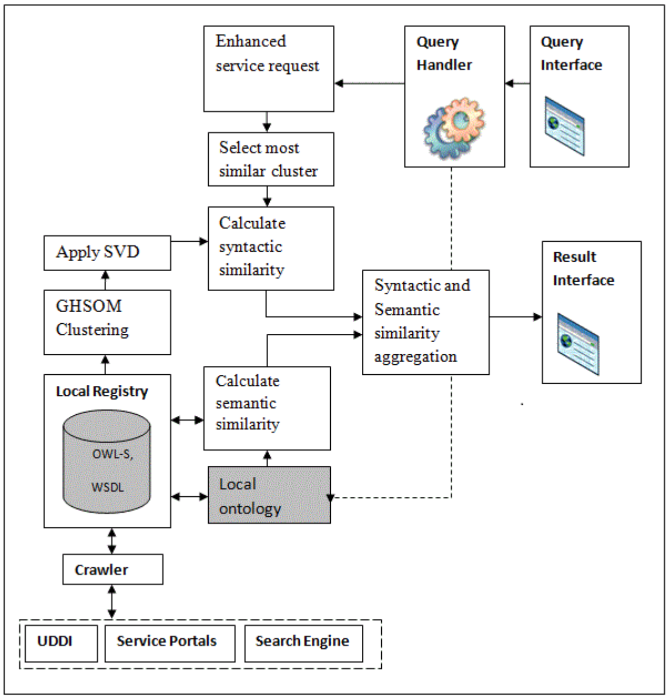

Fig. 1. The Outline of Hybrid Web Service Discovery Approach

The system supports service discovery by using keyword based, semantic based and hybrid query types.

User can customize their queries through the guided model of Query Interface.

It is possible to query by:

- Keyword based queries, where the keyword can appear in WSDL file of corresponding Web services.
- Strict ontology-based queries, where metadata searches are translated into a query language-SPARQL.
- Hybrid matching where both syntactic and semantic matching is combined.

The keyword based search for Web service WSDL files is described in [4], but here we extend the method on hybrid Web service search. First keywords of user query are extracted where all keywords are considered as terms. Each of the extracted terms is expanded using the WordNet [9] to enhance its semantics. WordNet.Net, is an Open-Source .NET Framework library for WordNet developed by Malcolm Crowe and Troy Simpson.

Such customization method improves the set of relevant service retrieved, compared with unique way of Web service searching.

Algorithm1: Enhance Web Service Request
Input: Web Service Request
Output: Enhanced Service Request SR_e

```
begin
  foreach web service request do
    perform tokenization
    word stemming
  end for
  $SR: = \{Srt_1, \ldots\ldots Srt_n\}$
  $SR_e: = SR$
  $Score\ sum := 0$
  foreach web service request vector term $Srt_i$ Є $SR$
    $bestcandidate : = -1$
    $bestscore: = -\ max\ Int$
    foreach ontology concept $C_j$Є$C$
      if ( $C_j$ is still free && $r\ [i,j] > bestscore$)
          /* r [i, j] – semantic similarity between
          the appropriate sense of word at position i of SR
          and the most appropriate sense of word
          at position i of < */
        $bestscore := R[\ i,\ j]$
        $bestcandidate <- j$
      end if
    end for
    if (bestcandidate!= -1)
      mark the best candidate as matched item
      $scoresum <- scoresum + best + score$
    end if
  end for
  return $SR_e$
end
```

III. Keyword Based Searching

Keyword based Web service discovery procedure begins with taking as input a user query, which is used as search criterion and returns a set of web services matching this query.

The process of discovery consists of the following phases:

- Decomposing the Web Service Corpus.
- Service Discovery in Latent Semantic Space

A. Decomposing the Web Service Corpus

At the beginning, the Web service collection is divided into smaller groups of related Web Services using unsupervised clustering method GHSOM.

The process starts with pre-processing of service description files. Web service description files are converted into numerical vectors suitable for GHSOM training. The service I/O as well as service description and service name concepts are extracted, where concepts are considered as terms. Here is important to note that some features of the SOMLib[6] are used to create the *tf x idf* input vectors.

GHSOM is able to cluster the representation vectors, arranged as nodes in hierarchy, where each hierarchy presents a group of the services related according to their semantic similarity. Each hierarchy is used to construct a SVD matrix.

B. Service Discovery in Latent Semantic Space

GHSOM is used to divide the Web service description files in different clusters. Each cluster has a center. Based on Euclidian distance, the similarity between a Query vector and a cluster center can be calculated. On the next step, the SVD is applied to the cluster whose cluster center is more similar to the query vector.

Suppose the relevant cluster has *n* Web Services with *m* corresponding description files $WS=\{WS_1,\ldots.,WS_n\}$. So A_{ij} matrix is created with terms as row and service description files as columns. The Latent semantic indexing [5] involves the Singular Value Decomposition technique which replaces the original matrix with a low level rank approximation matrix $A \approx A_k$.

Algorithm2: Algorithm for syntactic matching
Input: Service request SR_e
Output: Sorted list of services

```
begin
for all clustercenterr $K_{ci} \in K_c$ do
   AppCluster= $\mathrm{MinDis}(\mathrm{sim}(SR_e, K_{ci}))$
end for
return AppCluster
   $q_v \leftarrow$ EnhanceRequest(query)
   $s_v \leftarrow$ ReadSingularV alues()
for all term in $q_v$ do
   termV ector[i] ← readTermVector(term)
   w[i] ← calculate(Weight)
   $q_v[i] = \mathrm{termVector}[i] * w[i]$
end for
for all term in $q_v$ do
   $d_q = q[i] * T[i]$
   queryNorm ← calculateNorm($q_v$)
end for
for all $WS_i$ in DD do
//DD is document-by-document similarity table
   wsNorm ← readNorm()
   $\mathrm{sum} = \mathrm{sum} + DD[i] * d_q[i]$
   $\mathrm{sim} = \mathrm{sum}/(\mathrm{queryNorm} + \mathrm{wsNorm})$
   $\mathrm{Result} = \mathrm{Result} + (WS_i, \mathrm{sim})$
end for
return Sort(Result)
```

Formally, for a given service matrix, decomposition of *A*, can be represented as $A = T_0\ S_0 D_0^t$, where S_0 is *rxr* diagonal matrix composed of non-zero eigen values of AA^t. The columns of T_0 and D_0 orthogonal eigen matrixes composed of r non-zero values of AA^t. The number of the nonzero values in the diagonal in S_0 represents the rank of matrix A.

The aim here is reducing the rank of *A* and obtaining a reduced matrix A_k constructed from the *k*-largest singular triplets of A. A new *k-by-k* diagonal matrix S is obtained by deleting the zero rows and columns of S_0. Likewise, the corresponding columns of T_0 and D_0 are removed in order to derive new left singular vector *T* and a new right singular vector *S*. The resultant matrix is $A \approx A_k = TSD^t$, where *T* is the *m-by-k* matrix whose columns are the first *k* columns of T_0, *D* is the *n-by-k* matrix whose columns are the first *k* columns of D_0.

Next, the query can be represented as vectors in dimension-reduced semantic space as $Q = A_q^t TS^{-1}$, where A_q is the query's term vector in the original vector space *A*.

Once such a conversion is achieved, any user query can be compared to existing service documents which produce a ranking vector in the reduced semantic space A_k.

In order to recommend the relevant service documents to a user, the cosine similarity between document vectors and query is used:

$$Sim(S_i, Q) = \frac{\sum_i w_{Q,j} w_{i,j}}{\sqrt{\sum_j w_{Q,j}^2}\sqrt{\sum_i w_{i,j}^2}} \qquad (1)$$

where *Q* is a query, *S* is a service relevant to *Q* and *w* are weights

IV. Strict Ontology based Searching

To use strict ontology based search, two tasks must be fulfilled. The first task is to define the service's domain ontology in terms of OWL classes, properties, and instances. The second task is to collect or create OWL-S descriptions of the services, relating the description to the domain ontologies.

These descriptions are stored in Service Registry which serves as repository of all OWL-S files. These files are used during matching process where a semantic match between service advertisements and service requests is computed.

We used SemWeb Library for C# as the developing platform. SemWeb Library for C# is an RDF API for .Net framework developed by Joshua Tauberer. It provides a command-line tool for loading data into a database which is used to load the triples into a MySQL database. At the moment of new Service advertisement registration, OWL-S files are parsed by extracting necessary semantic information and load them into a triple store as triples <S, P, O>.

This API has support for SPARQL interrogations by storing them in special object, Query class objects for local queries and SparqlHttpSource for remote ones.

Once the data is in a triple store, SPARQL allows running all sorts of queries against the data set. This can be performed by configuring the"web.config" which enables to configure the ASP.NET server that will run the SPARQL endpoint.

The figure 2 displays the strict ontology based searching.

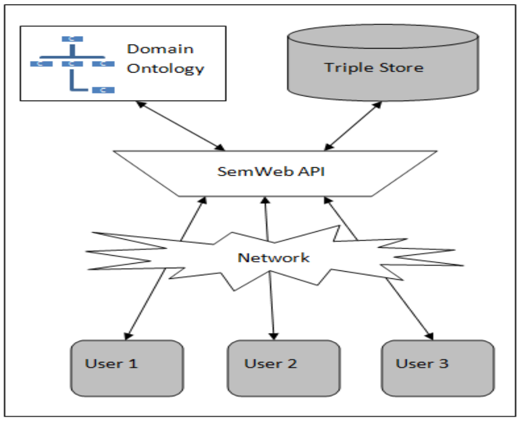

Fig. 2. The Outline of Strict Ontology Based Web Service Discovery

As it can be seen, the typical architecture of ontology based search is similar to relational database model. The semantic service request is received from the user, query methods are called to be executed which runs the queries and extracts results which meets the user's requirements.

Different studies have already presented a series of matchmaking mechanisms for Semantic Web service discovery. Massimo Paolucci [10] has presented an algorithm for advertisement matching with a request, where the matchmaker compares the input and output concepts of user request with all the services in registry and calculates the similarity.

This algorithm is adopted in our hybrid service searching, where each advertisement (adv) stored in the triple store will be submitted as parameter to the match function.

V. Hybrid Searching

Hybrid Web service discovery combines the previous explained keyword based searching with semantic searching. In this case, the traditional matching on input and output called relaxed algorithm presented by Paolucci [10], will be extended by using the Generalized Cosine Similarity [11]. This algorithm calculates final values of inputs and outputs by calculating matching levels for each input and output.

We will follow the main idea behind this algorithm which is: an advertisement matches a request when all the outputs of the request are matched by the outputs of the advertisement, and all the inputs of the advertisement are matched by the inputs of the request.

Formally, let ξ be the set of all advertisements in advertisement repository. For a given Query *Q*, the matchmaking algorithm returns the set of all advertisements which are compatible*, Match (Q).*

$$Match(q)=\{A\in \xi \mid compatible(A,Q)\} \tag{2}$$

In this approach Generalized Cosine Similarity Measure is used in order to calculate the similarity between a query and advertisement, which is an extension of relax algorithm:

$$Sim(A\ ,Q)=\frac{\vec{A}\ x\vec{Q}}{\left|\vec{A}\right|^{2}\ x\left|\vec{Q}\right|^{2}} \tag{3}$$

For given to terms or collection of elements C1and C2, the semantic distance is defined as similarity of concepts in relation *subClassOf.*

Algorithm3: Semantic Matching
Input: Service request SR_e
Output: $OverallScore$, J

```
begin
foreach Service Input $SWInp_i$ € $SWInp$
  foreach Service Request $SRInp_j$ € $SRInp$
      sim (SWInp SRInp)
  end for
  $Score_i \leftarrow Max(\text{sim (SWInp, SRInp)})$
  if $Score > \varphi$
    $J = J \cup Score$
  end if
end for
$Score_i \leftarrow C_{1=1..n} I_k \in I$
foreach Service Output $SWOup_i$ € $SWOup$
  foreach Service Request $SROup_j$ € $SROup$
  sim (SWOup SROu)
  end for
    $Score_o \leftarrow Max(\text{sim (SWOup, SROup)})$
  if $Score > \varphi$
    $J = J \cup Score$
  end if
end for
$Score_o \leftarrow C_{1=1..n} I_k \in I$
$OverallScore \leftarrow (Score_i + Score_o)/2$
return $OverallScore$, $J$
end
```

In this manner, we should take in consideration the depth of the node and lowest common ancestor which is the node of greatest depth that is an ancestor of both C1 and C2, and the semantic similarity can be defined as follow:

$$Sim(C_1,C_2)=\frac{2*depth(LCA(C_1,C_2))}{depth(C_1)+depth(C_2)} \tag{4}$$

This similarity information will be used during the matching process in rating the input and output matching.

VI. Syntactic Similarity and Semantic Similarity Aggregation

Let Sim_{synt} represent the syntactic based similarity defined in keyword searching and Sim_{sem} represent the semantic based similarity defined in hybrid searching above.

The service similarity metric in our proposed hybrid Web service discovery is defines as:

$$Sim(S_1,S_2)=(\alpha * Sim_{synt}(S_1,S_2)+\beta * Sim_{sem}(S_1,S_2))/2 \qquad (5)$$

$$\alpha+\beta=1 \qquad (6)$$

Where α and β are two parameters for adjusting search performance.

VII. Evaluation Results

To evaluate the accuracy of the proposed approach and to measure the overall performance, precision, recall and F-Score curves [8] are used.

The aim of this experiment was to demonstrate that Hybrid search can show better performance compared to Keyword based and Semantic based searching methods.

The comparative evaluation was performed on 20 queries. Queries are generated in such way that includes two groups of documents, either WSDL files or OWL-S files.

The analyses of the results of F-score curves for each query indicated that:

• In the case of Q2, Q3, Q4, Q18 which are chosen in a way to return only WSDL files, that are relevant results to them. Semantic search returns 0 results, while Hybrid method returns the same results with the Keyword based method.

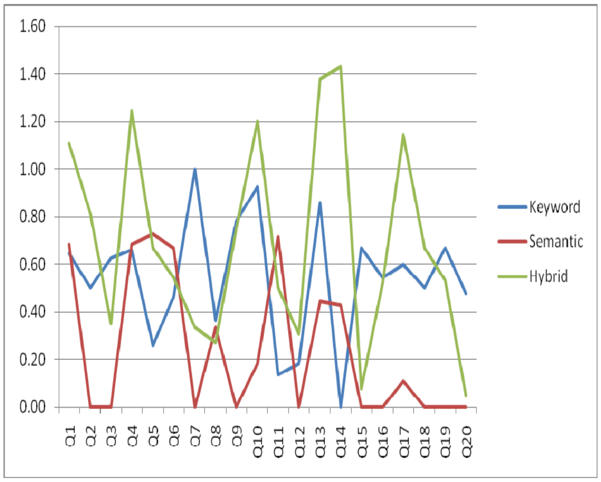

Fig. 3. F-Score Results for each Query for Three Modalities

• In the case of Q14, key words are selected in such manner to return OWL-S files only. For this query, Web service collection file, contains 5 OWL-S relevant files but no WSDL files. In this case, pure Semantic method retrieves only 3 services from 5 possible services as result, while Hybrid method retrieves the all possible services as result. The indicated results are consequence of the pure Semantic method which returns results only when the key words used in the query, are defined as input or output in an existing OWL-S file. Hybrid method on the other hand, does not take into consideration only terms defined as input and output parameters, but also other parameters as well, such as service name, description, etc., which enable it to return all services that deal with key words used in query .

• In other cases, the queries are selected in a way to return both OWL-S and WSDL files as possible results. In that case, all of the methods can return relevant services. However, Hybrid method returns better results compared with two other methods. It outperforms the Keyword based method because it takes into account the semantic similarity of keywords used in the query, and outperforms the pure Semantic method, for the same reasons mentioned for query Q14.

From the results it is obvious that the Hybrid Web service discovery approach perform significantly better in comparison to syntactic as well as to pure semantic based service selection.

VIII. Conclusion

In this paper a new Web service discovery approach which is based on hybrid semantic matching algorithm is proposed. The method extends the reasoning of semantic search paradigm combining it with the flexibility of keyword-based retrieval. Furthermore, we carried out a series of experiments to evaluate the correctness and performance of the proposed Web service discovery method, and show that it outperforms both keyword-based search and pure semantic search in terms of precision and recall.

References

[1] Papazoglou, M. P., Traverso, P., Dustdar, S., and Leymann, F. 2007. *Service-Oriented Computing: State of the Art and Research Challenges*. Computer 40, 11 (Nov. 2007), 38-45.

[2] W3C, OWL-S: Semantic Markup for Web Services, http://www.w3.org/Submission/OWL-S/

[3] F.Ismaili, B.Sisediev, *Web Services Research Challenges, Limitations and Opportunities*, WSEAS TRANSACTIONS on INFORMATION SCIENCE & APPLICATIONS, ISSN: 1790-0832, Issue 10, Volume 5, October 2008

[4] F.Ismaili, B.Shishedjiev,Xh.Zenuni, B.Raufi, *GHSOM-based Web Service Discovery,* Proceedings of the European Computing Conference, ISSN: 1790-5117, 2010

[5] Mi IsLita. http://www.miislita.com/term-vector/term-vector-3.html

[6] The SOMLib Digital Library Project. http://www.ifs.tuwien.ac.at/~andi/somlib/

[7] SemWeb.NET: Semantic Web/RDF Library for C#/.NET http://razor.occams.info/code/semweb/

[8] Jiawei, H, Micheline Kamber, *Data Mining: Concepts and Techniques*, 2nd Edition, ISBN: 13: 978-1-55860-901-3 , Elsevier, 2006

[9] WordNet, A lexical Database for English, http://wordnet.princeton.edu/

[10] M. Paolucci, T. Kawamura, T. Payne and K. Sycara. *Semantic Matching of Web Services Capabilities*. In *Proceedings of the 1st International Semantic Web Conference (ISWC2002)*. 2002

[11] Information Retrieval,http://www.dcs.gla.ac.uk/Keith/Chapter.7/Ch.7.html