Comparison of Web Service Similarity-Assessment Methods

J.Uma Maheswari
Assistant Professor(Sr.Gr)
Dept of CSE,
PSG College of Technology,
Coimbatore – 641004.

G.R.Karpagam, PhD
Professor,
Dept of CSE,
PSG College of Technology,
Coimbatore – 641004.

S. Indhumathy
PG Student
Dept of CSE,
PSG College of Technology,
Coimbatore -641004.

ABSTRACT
Due to the advent of service oriented architecture, web services have gained popularity. The need for efficient web service discovery increases because of the enormous growth of the web services. The main concern of this paper is to addresses the challenge of automated web service discovery and service similarity assessment. It utilizes the WordNet and a traditional information retrieval method, combined with structure matching to identify potentially useful services and estimating their relevance. The objective of this paper is to find the best suitable web service assessment method by comparing the three web service similarity assessment methods namely WordNet-powered vector space model, Structure matching and Semantic structure matching.

Keywords
Web Service, Semantic matching, Structure matching, Vector space, wordnet

1. INTRODUCTION
The corporate world is moving towards Service oriented architecture, web service technology gaining popularity today. According to [1], "web service is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. It is an executable component which can be invoked remotely to perform business operations using the protocols like XML, SOAP, UDDI and WSDL. The web populated components used in the web services are: EXtensible Markup Language (XML) [2] is a protocol for containing and managing data with structured documents could be used on the WEB. Web services are XML based application components. SOAP [3] is an XML based protocol for accessing web services. The goal is to allow for a machine readable document to be passed over any multiple connection protocols to create a decentralized, distributed system. UDDI [4] means Universal Description, Discovery and Integration. UDDI is a directory for storing information about web services which could be described by WSDL. The core of UDDI is the UDDI Business Registry, a global, pubic, online directory. It is used to classify and publish available web services to a repository and enable discovery by potential users. SOAP is a protocol for communicating with a UDDI service. WSDL [5] is Web Service, Definition Language. WSDL is the piece of Web services framework that describes how to connect to web service providers. A WSDL document is just a simple XML document. It contains set of definitions to describe a web service such as definition, Data types, Message, Operation, Port type, Binding, Port, and Service. Because of increasing number of web services, discovery of user required service from a pool of web service is a challenging task. Traditional web service discovery method is based on syntactic approach using UDDI, in which it retrieves the service descriptions that contain particular keywords from the user's query. This procedure leads to irrelevant discovery, because the keywords in the query can be semantically similar but syntactically different, or syntactically similar but semantically different from the terms in a service description. So, in this paper both semantic and syntactic approaches to retrieve the most appropriate web services from registry are considered.

2. RELATED WORK
Component retrieval is used to locate and identify appropriate components. A semantic component is one or more segments of text that contains information about a particular aspect of the concept [6]. The two components cannot be linked directly with respect to their interface. In order to interconnect these two software components, the programmer could rewrite one of the modules to meet the interface of the other. The problem of web-service discovery is similar to the problems of component retrieval and information retrieval. A WSDL specification is the specification of a software component including a specification of its interface signature and a specification of where the actual implementation exists and how it can be used. There are two categories of methods for component discovery: signature matching and specification matching.

Signature matching is a method for organizing, navigating through, and retrieving from software libraries [7]. There are two kinds of matching-function matching and module matching. The signature of a function is simply its type; the signature of a module is a multi set of user-defined types and a multi set of function signatures. Signature matching is the process of determining which library components "match" a query signature. Signature matching is an efficient means for component retrieval. Function signatures can be automatically generated from the function code. Furthermore, signature matching efficiently prunes down the functions and/or modules that do not match the query, so that more expensive and precise techniques can be used on the smaller set of remaining candidate components. The disadvantages of these methods are it considers only function types and ignores their behaviours and two functions with the same signature can have completely opposite behaviors.

Specification matching aims at addressing the above problem by comparing software components based on formal descriptions of the semantics of their behaviours. So, signature-matching is extended with a specification-matching scheme [8].

Traditional information-retrieval methods rely on textual descriptions of artifacts to assess their similarity. Full text retrieval methods search all documents for the specified string. These methods are most straightforward and require minimum effort to maintain, but the response time is bad when files are large [9]. There are two reasons for that knowledge of these methods is useful for the newer developments; they are Traditional text retrieval and Semantic Information retrieval.

The traditional text retrieval method is one of the fundamental models. It is the vector model [12]where each document is represented as a t-dimensional vector where t is the number of distinct words in the document. Similarity between two web services can be computed based on their representing vectors. Representing documents and queries as vectors allows for relevance feedback, and increase effectiveness of the search. It is based on some of the techniques for efficient retrieval. The methods for text retrieval are: Full text scanning, Signature Files, Inversion and Vector Model and Clustering.

The Full text scanning method is used to locate the documents that contain a certain search string term is to search all documents for the specified string. String is a sequence of characters.

In Signature Files method [13] each document yields a bit string signature using hashing on its words and superimposed coding. The resulting document signatures are stored sequentially in a separate file which is much smaller than the original file and can be searched much faster. They used a stop list to discard the common words and an automatic procedure to reduce each non-common word to its stem.

In Inversion each document can be represented by a list of keywords which describe the contents of the document for retrieval purposes. Fast retrieval [13] can be achieved if those keywords are inverted. The keywords are stored e.g. alphabetically in the index file for each keyword they maintain a list of pointers to the qualifying documents in the "postings file" .The advantages are that it is relatively easy to implement it is fast and it supports synonyms easily. The disadvantages of this method are the storage overhead, the cost of updating and reorganizing the index.

The basic idea in clustering[13] is that similar documents are grouped together to form clusters in Vector Model and Clustering. Document clustering involves two procedures, the cluster generation and the cluster search.

In traditional Information retrieval techniques uses only a small amount of the information associated with a document as the basis for relevance decisions [7]. But semantic information retrieval method tries to capture more information about each document to achieve better performance. In signature file approach, each document generates a bit string as its signature, and searches are done on these signature files. The advantages are that it is easy to implement and it is robust. The disadvantages of this method are, its response time is bad when used on large files and fast retrieval but storage overhead and cost of maintaining is increased.

From the literature survey made, it is clear that once the web service are created,

- The category-based service-discovery method is clearly insufficient.
- It is the responsibility of the provider/developer to publish the services in the appropriate UDDI category.
- They must, browse the "right" category to discover the relevant services. More importantly, these methods do not provide any support for selecting among competing alternative services that could potentially be reused.
- Prioritization of the candidates is again the responsibility of the consumer.

3. WEB SIMILARITY ASSESMENT METHODS

The following methods aimed towards addressing the challenge of automated web service discovery and service similarity assessment. It represents a suite of methods that utilizes WordNet, an on-line lexical database for the English language, combined with a traditional information-retrieval method and structure and identifier matching for identifying potentially useful services and estimating their relevance to the task. The three methods can be used severally or can be combined to retrieve the most similar services to a given task.

a. WordNet-Powered Vector-Space Model
b. WSDL Structure Matching
c. Semantic WSDL Structure Matching.

3.1 Wordnet-Powered Vector-Space Model

The vector space model combines both information retrieval method and WordNet to retrieve the similar services. In vector-space model, documents and queries are represented as T-dimensional vectors, where T is the total number of distinct words in a document collection after the pre processing step shown in Fig.1.

Preprocessing

For all services, WSDL text description is extracted and it is stemmed by using the reduced version of porter stemmer algorithm. The Porter Stemmer [11] is a conflation Stemmer developed by Martin Porter at the University of Cambridge in 1980. The Stemmer is based on the idea that the suffixes in the English language (approximately 1200) are mostly made up of a combination of smaller and simpler suffixes. The Porter Stemmer is a very widely used and available Stemmer, and is used in many applications. The pseudo code for the stemmer is given in Fig.2

```
For all words in the service description, repeat the below
If (suffix of a word==”sses”) replace it by “ss”. (caresses -> caress
Else If (suffix of a word==”ies”) replace it by ”i”.( ties -> tie)
Else If (suffix of a word==”s”) remove. (cats -> cat)
Else if (suffix of a word==”ed”) remove.( plastered-> plaster)
Else if (suffix of a word==”ness”) remove.( goodness -> good)
Else if (suffix of a word==”ing”) remove.(singing->sing)
Else if (suffix of a word==”ful”) remove.(hopeful -> hope)
Else if (suffix of a word==”icate”) replace it by “ic”.(triplicate -> triplic)
Else if (suffix of a word==”alize”) replace it by “al”.(formalize -> formal)
Else if (suffix of a word==”ation”) replace it by “ate”. predication -> predicate
```

Fig 2: Stemming Algorithm

After stemming the unwanted words are removed from the list of words obtained from the WSDL description. The pseudo code for stop word removal is given below.

```
stop_word[][10]={"i", "a", "about","an","are","as","at","be",
                 "by","com","for","from","how","in", "is","it",
                 "of","on","or","that","the","this","to","was",
                 "what","when","where","who", "will","with" }
        for all word_in in v[]
        for all word_out in a stop_word list
        if(word_in of a query == stop word_out)
         remove the word
```

Fig 3: Stop word removal

After removing the stop word ,Each term in the vector is assigned a weight that reflects the importance of a word in the document. This value is proportional to the frequency a word appears in a document and inversely proportional to number of documents.

$$w_{ij} = tf_{ij}\, idf_i = tf_{ij} log2\ (N/df_i) \ldots\ldots\ldots\ldots\ldots \quad (1)$$

Where,

tf_{ij} = frequency of term i in document j, normalized across a document,
idf_i = the inverse document frequency of term i,
N = total number of documents in the collection,
df_i = document frequency of term i

Vector space model with WordNet is used to find semantically similar words to textual descriptions extracted from WSDL service specification files. The WordNet-powered vector-space model extension thus involves maintaining three sub-vectors for each document and query. The three groups of words for each service description are:

Group 1: Original Words: Original textual descriptions extracted from WSDL specification files. Group 2: Words' Synonyms: Synonyms of original words.Group 3:Words' Family: Hypernyms, hyponyms, and siblings of original document terms.

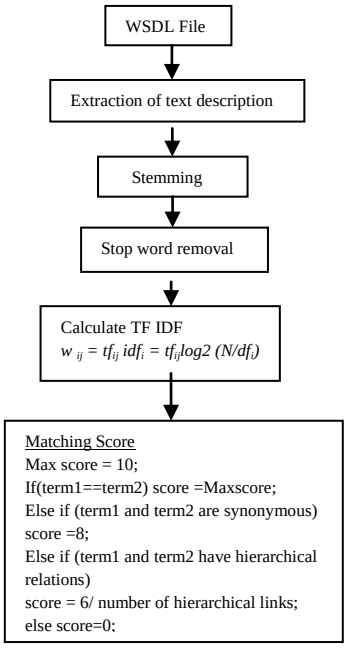


Fig 1: WorldNet powered vector space model

Different weights are assigned to different sub-vector matching scores. Then the similarity scores are calculated from documents and queries matched with the corresponding sub-vectors.

3.2 WSDL Structure Matching

The structure-matching method of WSDL specifications is a natural extension of the signature matching method for component retrieval shown in fig.4

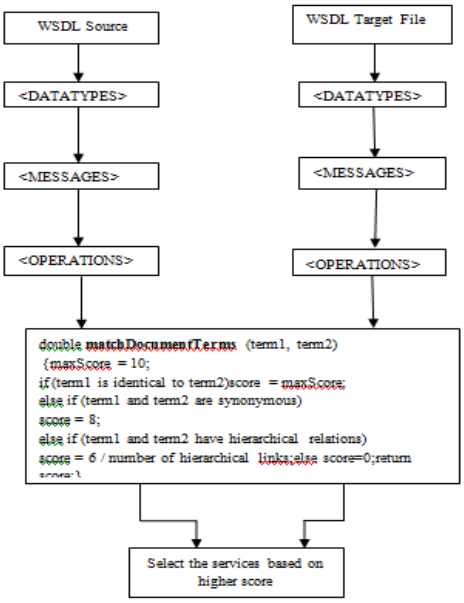


Fig 4: Structure matching

It involves the comparison of the operations' set offered by the services, which is based on the comparison of the structures of the operations' input and output messages, which, in turn, is based on the comparison of the data types communicated by these messages.

3.2.1 Structure Matching Data Types

The data types involved in the two WSDL specifications are compared. If the source and the target data types are matched then it gets the highest score.

3.2.2 Structure Matching Message

After evaluating the data-type matching scores, the structures of the source-service messages against the target-service messages are matched. The objective of this step is to identify the parameter correspondence that maximizes the sum of their individual data-type matching scores.

3.2.3 Structure Matching Operations

The third step of the process is the matching operations are based on the process of matching messages. The matching score between two operations is the sum of the matching scores of their input and output messages.

Finally, the overall score is computed by identifying the pair-wise correspondence of their operations. After all target WSDL have been matched against the source WSDL specification, they are ordered according to their "overall matching scores": a higher score indicates a closer similarity between the target and source specifications.

3.3 Semantic WSDL Structure Matching

Semantic WSDL structure matching is similar to WSDL structure matching. Both tries to match the similar components in the source and target service. Instead of assessing structure similarities between the two services, the WordNet-powered identifier matcher calculates the semantic distances between identifiers of data types and names of services and operations to assess service similarities shown in fig.5.

3.3.1 Identifier Matching Data Types

The process of matching service identifiers is very similar to the process of matching data types in WSDL structure matching. Instead of matching types of parameters, the identifier matcher uses WordNet to calculate semantic distances between the names of data types (identifiers).

3.3.2 Identifier Matching Operations

Unlike in WSDL structure matching, messages are not matched in identifier matching process. This is because message names are not necessarily always programmer-defined names. After evaluating data-type identifier matching scores, the source and target services' operations are matched. Given a source and a target operation, there are many possible correspondences between their parameter lists.

Fig.6 lists the algorithm matchDocumentTerms that explains the WordNet based "cost structure" for assessing the similarity of two identifiers. If two words are identical or synonymous, they are assigned a maximum score of 10 and 8 respectively. Otherwise, if two words are in a hierarchically semantic relation, i.e. they are hypernyms, hyponyms or siblings to each other, then count the number of semantic links between these words along their shortest path in WordNet hierarchy.

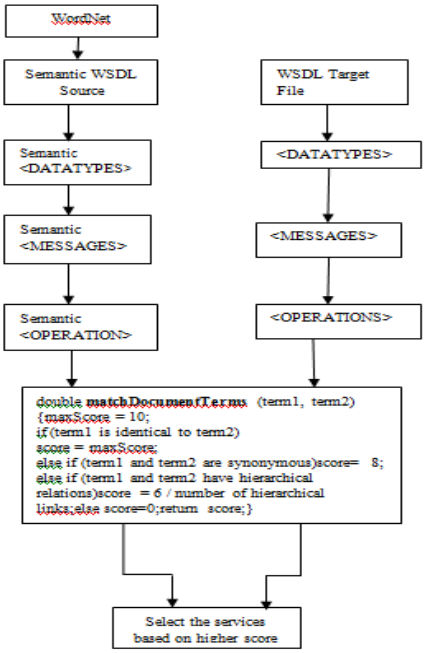


Fig 5: Semantic structure matching

```
double matchDocumentTerms (term1, term2) {
  maxScore = 10;
  if (term1 is identical to term2)
    score = maxScore;
  else if (term1 and term2 are synonymous)
    score = 8;
  else if (term1 and term2 have hierarchical relations)
    score = 6 / number of hierarchical links;
  else score =0;
  return score; }
```

Fig 6: Algorithm for Matching Two Document Terms

4. EXPERIMENTAL RESULTS AND DISCUSSION

The service-discovery method as a whole and the effectiveness of its components of the retrieved methods are evaluated using Precision and recall by the number of relevant and irrelevant results. Precision and recall are the basic measures used in evaluating search strategies. The performances for the service discovery with Word Net-powered vector space model, discovery with structure matching, discovery with semantic structure matching, are measured.Precision and Recall values for the methods are recorded and shown in Fig.7 and Table 1.

RECALL

RECALL is the ratio of the number of relevant records that are retrieved. It is usually expressed as a percentage.

$$Recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

PRECISION

PRECISION is the ratio of the retrieved documents that are relevant. It is usually expressed as a percentage.

$$Precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

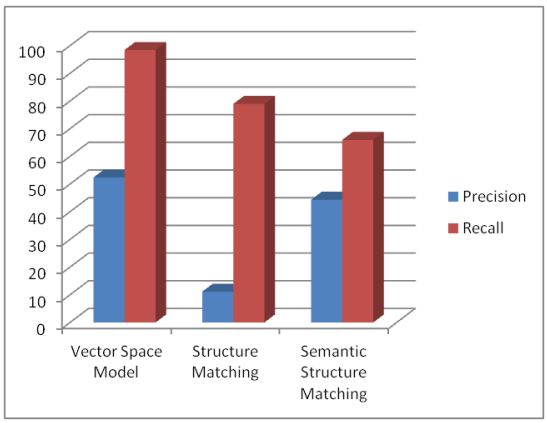


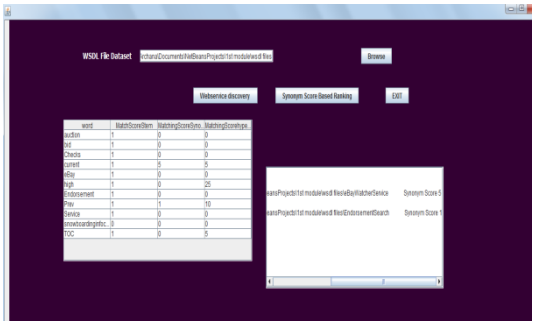
Fig.7.Precision and Recall Graph

Table 1: Precision and Recall measure

Algorithm	Precision	Recall
Vector Space Model	52.35	98.42
Structure Matching	11.12	78.95
Semantic structure Matching	44.32	65.94

4.1 WordNet-Powered Vector-Space Model

In WordNet vector space model , the service descriptions specified in natural language of each service from each category (requests) is matched against the text descriptions of all other services from all categories is shown in Fig.8. The WordNet-powered vector space model achieves a precision of 52% at 98% recall on average on this set of web service experiments.



.Fig. 8 Word Net-Powered Vector-Space

4.2 WSDL Structure Matching

In structure matching algorithm, the structures of each service from each request is matched against the structures of all other services from all categories. Averages were calculated between service requests from each category and all candidate services. The candidate web services were ranked according to their similarity scores to the requests, and the top 50% of the list were considered to be relevant to the requests and were returned to the users.

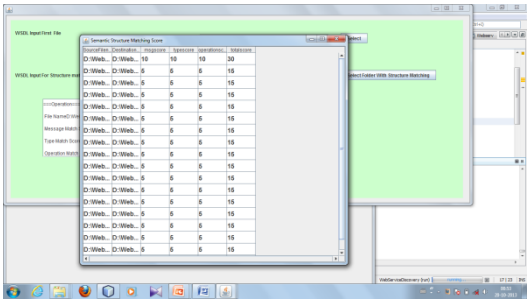


Fig. 9 Structure Matching Score Values

The precision is rather low in this set of experiments because some related services have considerably different structures and some irrelevant services can often have higher matching scores because they have many false substructures that happen to match the query structure. In structure matching algorithm it returns the low precision value of 11% and recall value of 78%.

4.3 Semantic WSDL Structure Matching

The effectiveness of this method would be improved by combining the two approaches that utilize both service structure and semantic information. First, WordNet-matching method was used to evaluate the semantic identifier similarity of the queries with the available services. Candidate services were ranked according to their relevance to the queries. Then from the set of likely candidates it was further refined by the structure-matching step assessing the structure similarity of the desired versus the retrieved services.

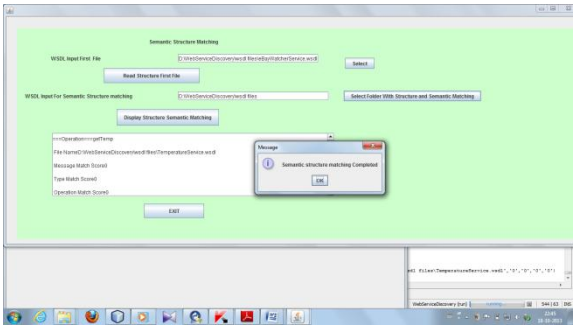


Fig. 10 Semantic WSDL Structure Matching

Candidate services were re-ranked according to their structure similarities to the queries and the services were returned as final results to the queries. Semantic structure matching method achieves a precision of 44% at 65% recall on average. Compare to performance of pure structure matching method, precision is improved by 33% from 11% and recalled is decreased by 13% from 78%.

The performances of Vector Space Model with WordNet-Powered having high precision values. Semantic Structure matching combined with WordNet returns better results than structure matching method in terms of identifying relevant services to the desired services.

5. CONCLUSION

This paper reports a web service discovery model that combines traditional information retrieval techniques with a structure-matching algorithm. Then it designed to calculate semantic and structural similarity between a desired service and a set of advertised services. It includes semantically similar words retrieved from WordNet database for all documents and queries. The performances of Vector Space Model with WordNet-Powered having high precision values. Semantic Structure matching combined with WordNet returns better results than structure matching method in terms of identifying relevant services to the desired services.

The WordNet Powered VectorSpace model combines with semantic structure matching algorithm can be used to retrieve the most efficient web services. Then, Structure matching algorithm can be extended to manipulate the full WSDL syntax. The inclusion of this structure information of services should help improve structure matching method's accuracy in discovering relevant services and also the proposed matching method can also be applied to OWLS in order to improve the semantics of web service description.

6. REFERENCES

[1] D. Booth, M. Champion, C. Ferris, F. McCabe, E. Newcomer, and D. Orchard."W3C Web Service Architecture". http://www.w3.org/TR/ws-arch/

[2] XML Introduction - What is XML? - W3Schools: http://www.w3schools.com/xml/xml_whatis.asp

[3] Simple Object Access Protocol (SOAP) http://www.w3.org/TR/2003/REC-soap12-part0-20030624

[4] Universal Description Discovery and Integration (UDDI). http://uddi.org/.

[5] Web Services Description Language (WSDL: http://www.w3.org/TR/wsdl)

[6] J. Purtilo and J.M. Atlee."Module Reuse by Interface Adaptation". Software Practice and Experience, Vol. 21, No. 6, 1991, 539-556.

[7] M. Zaremski and J. M. Wing."Signature Matching: a Tool for Using Software Libraries". ACM Transactions on Software Engineering and Methodology, Vol. 4 No. 2, 146-170, Apr. 1995.

[8] M. Zaremski and J. M. Wing. "Specifications Matching of Software Components". ACM Transactions on Software Engineering and Methodology, Vol. 6, No. 4, 333- 369, Oct. 1997.

[9] Faloutsos,D.W.Oard."survey of Information Retrieval and Filtering Methods", University of Maryland. Technical Report CS-TR-3514, August 1995.

[10] Wang and Eleni Stroulia Yiqiao "Semantic Structure Matching for Assessing Web-Service Similarity", Computer Science Department, University of Alberta, Edmonton, AB, T6G 2E8, Canada {yiqiao,stroulia}@cs.ualberta.ca.

[11] Noraida Haji Ali , Noor Syakirah Ibrahim, Porter Stemming Algorithm for Semantic Checking, ICCIT 2012.

[12] Yiqiao Wang, "Information Retrieval and Semantic Structure Matching for Assessing Web-Service Similarity", Thesis, Master of science 2003.

[13] Manish Sharma, Rahul Patel, "A Survey on Information Retrieval Models, Techniques And Applications", International Journal of Emerging Technology and Advanced Engineering ,ISSN 2250-2459, Volume 3, Issue 11, November 2013.