Asian Journal of Information Technology 15 (17): 3296-3305, 2016
ISSN: 1682-3915


# An Ontology Based Framework for Web Service Discovery from Natural Language User Queries

K.V. Enkatachalam, N.K. Karthikeyan and Ajay Basker
Department of Information Technology,
Sri Krishna College of Engineering and Technology, Coimbatore, India

**Abstract:** This study focus on providing a user friendly natural language query interface for web service discovery and maps it with both traditional and semantic web service description languages like WSDL, OWL-S etc and thus aiming to bridge the gap between various service description formats, by using a common registry schema. Our approach considers user requirements regarding input, output and purpose of web service and performs matches across all the three parameters on a web registry. The proposed framework uses an ontology based web service registry which integrates various web service related information like service description, classification, search keywords. The query processor performs basic NLP processing like stemming and tagging on the user query and converts it to an intermediate representation based on description logics which we call WS-QDL. The user query is then classified using classification ontology to limit the search process within applicable domains. The final preprocessing step optimizes the search process by replacing the query terms with the most related terms in the domain ontology. The search algorithm separates input, output and subject specifications from the user query and performs matching based on each criterion. The tests were performed based on OWL-TC3 dataset and the obtained results shows improvement in performance measures.

**Key words:** Semantic web services, ontology, wSDL, web service discovery, natural language processing, web service registry

## INTRODUCTION

Web services are loosely coupled self-descriptive, modular and web-accessible distributed software components which can be published in a web service repository to be discovered by software agents and composed as new value-added web services. The objective of web service discovery is to find an appropriate web service for a service requester. Traditional web services description standards like WSDL, UDDI and SOAP provide only syntactical functionality descriptions which makes human interaction mandatory in the discovery process. This requirement raises the need for more effective methods of web service discovery.

There were several approaches taken by various researchers to improve the discovery process. Some researchers proposed frameworks which takes user query input that is expressed in some web service description language like OWL-S, WSMO WSDL-S (W3C), etc. But, these approaches suffer from the disadvantage that user's knowledge of the standards is mandatory. It is always recommended that the discovery process should be driven by end user and not by technicians. So, a discovery mechanism should be developed in such a way that, it should bridge syntax or semantic gap between user's language and web service descriptions.

Similarly, some approaches developed had the disadvantage of making the discovery scope limited to a particular description language. Some approaches had the requirement that the service description should be semantically tagged. Since, most of the service descriptions exist in WSDL standard, these requirements couldn't be met.

Many a times a navie user may not be aware of technical terminologies which exist in a particular domain. As a result, the user query may not match many services which were actually relevant to the domain. Techniques based on semantics and NLP can be easily overcome such limitations. In order to address the above mentioned disadvantages of the existing approaches, we first propose a discovery framework based on user query expressed in natural language.

Proposed approach accepts user descriptions (sentences) as input, converts them into logical expressions by using the Natural Language Processing

**Corresponding Author:** K.V. Enkatachalam, Department of Information Technology,
Sri Krishna College of Engineering and Technology, Coimbatore, India

(NLP) methods. The user query in intermediate representation form is then classified into one or more domains, similar to web services in registry and thus reducing the search space. The user query is further optimized using a lexical database to increase match between user's terminology and domain terminology. Finally, the input, output and subject requirements of the user query is retrieved and matched with web service descriptions and the results are fetched. The widely used test collection OWLS-TC is applied in the proposed approach to evaluate the performance improvements.

**Literature review:** Semantic web and semantic web services have emerged as an active research area right from the beginning of this decade. Most state-of-the-art service discovery approaches are now based on semantic service description models like OWL-S, WSMO, SAWSDL, etc. Some of the recent works that influenced in the proposed work is given below.

Many of the existing systems pertaining to the web service discovery face the problem that users often have little knowledge about the web service-related technologies and implementation details. Adala *et al*. (2011) answered this issue by proposing a novel approach for automatic discovery of semantic web services which makes use of NLP techniques to match a user query, expressed in natural language with a semantic web service description described in WSDL, OWL-S or WSMO languages. Similarly, Gunasri and Kanagaraj (2014) developed a semantic web service discovery framework which makes use of NLP and clustering techniques. A similar approach was adopted by Sangers *et al*. (2013) which made use of NLP techniques and WordNet based similarity functions for match making. A semantic web service discovery framework was proposed by Sangers *et al*. (2012) which search web service annotations for match with user query consisting of keywords. Our proposed approach applies techniques like stemming and tagging of user query but for the purpose of converting an ambiguous user query to formal description using Description logics. This query form is then optimized with WordNet ontology to increase the match with keywords in service descriptions.

Farrag *et al*. (2013) proposed a Semantic Distance-based Matchmaking Algorithm (SDMA) which uses semantic distance as a measurement to match between user request and web services. SDMA uses a concepts tree to measure the semantic distance between any two concepts. In this method, classification of query is not done. But, it classifies web services. In our approach, we perform a Boolean match on keywords concerning input, output and subject concerning the query and the web service description. Also, we use WordNet based user query optimization to achieve an increased match between the two.

Batra and Bawa proposed the framework for automatic discovery of semantic web services. This framework extracts semantic annotations of web service descriptions to form term-category matrix and also uses Normalized Similarity Score (NSS) to classify the web service to one of the pre-defined categories. In our approach, we classify web services based on keyword match between web service description and a Classification ontology which is formed by training with TC3 dataset user queries.

The hybrid fuzzy match making approach (Fenza *et al*., 2008) exploits the fuzzy multiset mathematical model to discover OWL-S based service description in order to satisfy the user request. User queries are grouped using fuzzy multiset with the help of distance measure and the cluster center with nearest distance value is taken as the service which is closely matched with user query. Similarly, Cassar *et al*. (2014) presented a method using Probabilistic Latent Semantic Analysis (PLSA) like machine-learning techniques to extract latent factors from semantically enriched service descriptions. The latent factors are used to construct a model to represent different types of service descriptions in a vector form. Services are classified and matched based on these vectors. Paliwal *et al*. (2012) proposed a new approach by focusing on semantic based automated service discovery which includes semantic based service categorization based on an ontology and semantic-based service selection using techniques likes ontology linking and Latent Semantic Index. In our approach, classification of user query is done based on a classification ontology and the classification of web services is done based on keyword match between domain ontology and the set of keywords identified from service description.

From the surveys examined, the proposed system has unique features in the following way. The proposed approach accepts user requirements as input, converts them to logical expressions using the NLP processing methods which provides more flexibility for user in specifying giving user requirements whereas early methods accept keywords as user input (Sangers *et al*., 2013). Studied method limits to process web services which are described using Semantic web service description language while the proposed approach extends semantic based discovery features to WSDL. The user query is classified into one or more domains using classification ontology and discovery process is applied only on the selected domains.

**Background:** In this study, we describe some terminologies which we have used in our framework. Here, we give an overview about web service description languages used in the proposed system and then describe various NLP techniques and semantic technologies which we used to process the natural language user queries. Finally, we briefly describe about Description Logics (DL) which serves as base model for the proposed intermediate representation.

## MATERIALS AND METHODS

**Web service description languages:** Web service description languages are interface definition languages which are used for describing the functionalities offered by a web service. Such language usually provides machine readable description of details required for web service invocation and details of its input/output parameters. They can be generally classified as traditional and semantic web services. The traditional web service description standards like WSDL employ keyword based matching techniques. The semantic web services description languages like OWL-S, WSMO, etc., provide richer semantic description of web services, providing increased machine interpretability and better semantic web service discovery.

**WSDL:** The abbreviation ‘WSDL’ stands for ‘web service description language’. WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. Even though, WSDL allows typing of input/output parameters, it does not record pre-conditions and post-conditions for web service invocation.

**OWL-S:** OWL-S, formerly known as DAML-S (Ankolekar *et al*. (2002) is an OWL based semantic web description language. W3C describes OWL-S as ontology, within the OWL-based framework of the Semantic Web for describing Semantic Web Services. It will enable users and software agents to automatically discover, invoke compose and monitor web resources offering services, under specified constraints”. It consists of three parts, viz, service profile, process model and service grounding. A service profile describes the functionality of a web service. The process model describes client interaction with the web service. The service grounding specifies communication protocols, message formats, port numbers etc which are details used by the client to interact with the web service. The OWL-S enables semantic web service discovery and also provides provision to specify preconditions for invoking a web service.

**Natural Language Processing (NLP):** Natural Language Processing (NLP) (Jurafsky and Martin, 2008; Indurkhya and Damerau, 2010) is a field ofcomputer science, artificial intelligence and computational linguisticsconcerned with the interactions betweencomputersand human languages. NLP domain focuses on understanding or manipulating spoken or written human languages. In the proposed work, we employ the following NLP techniques which are listed below.

- Stemming is the process of reducing inflected words to their stem, base or root form
- stop word removing stop words are words which appear irrelevant for a specified purpose. The remover tool eliminates a predefined list of stop words from the given text
- Parts of Speech (PoS) Tagging is the process of markup words in a given text to its corresponding parts of speech like noun, verb, adjective, adverbs and so forth

**Semantic web technologies:** The term “Semantic Web” refers to the W3C’s idea of linked data over the Internet. According to the W3C, it is formally defined as “the Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise and community boundaries”. Semantic web technologiesrefer to a group of technologies which enable people to create data stores on the web, build vocabularies and write rules for handling data. In the following subsections, we give an introduction to the semantic technologies which we have used as part of the framework.

**Ontologies:** An ontology is defined as a formal, explicit specification of a shared conceptualization. An ontology defines the vocabulary used in a domain with its properties and describes relationships among them. Because of this property, ontology is often limited to a particular domain.

**Ontology languages:** There are many ontology languages like F-logic, Racer (Haarslev and Moller, 2003), Cycl (Sireteanu, 2013), RDF, OWL which are used to represent ontology in a machine readable form. Most of them are XML applications and the Web Ontology Language (OWL) seems to be more promising because, it gives more provision for machine interpretability compared to other languages. OWL has three sub languages, namely, OWL-Lite, OWL-DL and OWL-Full which represent increasing levels of expressiveness.

**Word-sense disambiguation:** Word sense disambiguation is the process of identifying the meaning of a word used in a particular sentence. This is relevant when the word used has different meaning in different contexts. WordNet is a commonly used tool for this purpose.

**Wordnet:** Miller (1995) and Fellbaum (1998) WordNet is a large lexical database of English language. In WordNet, words belonging to different parts of speech like nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms known as synsets. These synsets are interlinked by conceptual, lexical and semantic relations. Additional details on WordNet can be found by Fellbaum (1998).

**Description logics (DL):** Description Logics (DLs) refer to a family of languages used for knowledge representation. DLs can be considered as decidable fragments of first order logics and are equipped with a formal semantics which allow unambiguous representation and machine interpretation of content. DLs allow logical deduction, producing reasoning capabilities.

Description logics provide features to model relationships between entities in a domain. Similar to Predicate logic which consists of unary/binary predicates and constants, DLs support three types of entities namely, concepts, roles and individuals. Concepts represent set of individuals, roles represent relationship between individuals and individual names represent instances in a domain.

A DL consists of a set of statements, called axioms, each of which should be true within the domain described. These axioms are generally classified as Assertional (ABox) axioms, Terminological (TBox) axioms and Relational (RBox) axioms. In this study, we first give the basic definitions of terminologies associated with DL.

**Tbox axioms:** TBox contains statements which depict concept hierarchies, i.e., they describe relationship between concepts. TBox axioms include concept inclusion and concept equivalence. concept inclusion declares concept hierarchy by using subsumption. Concept equivalence asserts equivalence between two concepts, resulting that both have same individuals.

**Abox axioms:** ABox capture knowledge about named individuals, i.e., it contains statements regarding properties of individuals and relationship between individuals. The most common ABox axioms are concept assertions and role assertions. Concept assertion declares individual to belong to a concept. Role assertion describes relationships between named individuals. Another assertion called individual inequality asserts disjointness between two named individuals.

**Rbox axioms:** RBox axioms refer to properties of roles the two main supported axioms are role inclusion and role equivalence. Role Inclusion declares hierarchy within roles and role equivalence express equivalence between two roles. Similarly, role disjointness asserts that two roles are unequal. RBox axioms also include role characteristics like reflexivity, symmetry and transitivity.

**Role restrictions:** DL supports feature to link between concepts and roles. There are two role restrictions that are specified by DL. they are existential restriction and universal restriction. For e.g., Parent = ∃parent of Person is a existential restriction which states a Parent is someone who is parent of atleast one person individual. similarly, the expression Human = ∀consists of Person is an universal restriction stating all persons are humans.

**Nominals:** DLs provide features to define concept in terms of other concepts and roles. concepts may be declared as an enumerated set of instances. For e.g., cricket-team = {john} U {paul} U {mark} implies that the concept cricket-team consists of a set of individuals namely, {john}, {paul} and {mark}.

**Proposed approach:** In this study, we present our web service discovery framework which is illustrated by the system architecture given in Fig. 1. The proposed framework works across multiple description formats and extend the semantic based services to syntactic web description approaches too. Our search mechanism incorporates various NLP techniques to process the natural language user query, converts it to a DL based intermediate representation and finally perform searches in an ontology web registry matching between input, output and purpose specification in user query and service description.

**Framework architecture:** Our proposed framework aims to provide an efficient web service search according to a natural language user query. We make use of techniques like conversion to intermediate form, word sense disambiguation and classification of user query to increase the match between user requests and service description. Further, we aim at offering the semantic features to non-semantic descriptions like WSDL which is the most common format available.

We assume that there is a set of web services described in WSDL and OWL-S formats and published in

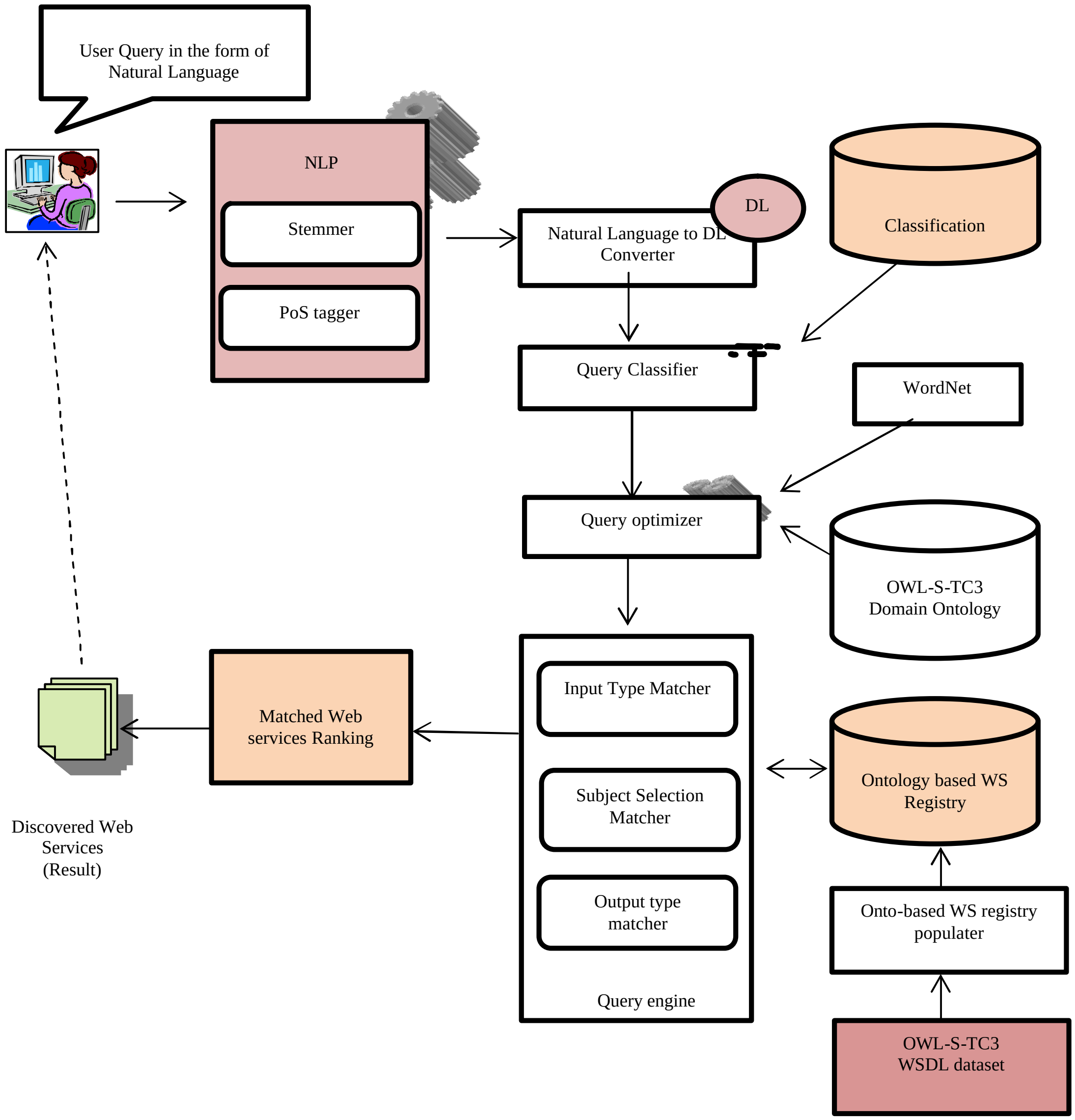


Fig. 1: System architect

ontology service registry. Here, we use WSDL and OWL-S files given in OWL-TC3 dataset. These descriptions are parsed and read by the framework to extract all required information elements for the match making process. Similarly, classification ontology used for user query classification is constructed by training the classifier system with 29 user queries in OWL-TC3 dataset and 500 manually constructed user queries. Similarly, domain ontology across 7 fields, namely, education, medical care, food, travel, communication, economy and weapons, as given in the dataset is manually constructed. The user query is described in our DL based syntax after NLP processing. word sense disambiguation with WordNet lexical database is then used to increase the match between user query and web service description. The search process is performed between the classified optimized user query in our DL based form to documentation in registry ontology.

## RESULTS AND DISCUSSION

**Web service parser and reader:** As shown in the literature survey, there are many web service description languages like WSDL, OWL-S and WSMO. Since, each description language follows different syntax, different parsers are required. For example, Jena is a parser for OWL-S and WSMO4J is a parser for WSML (Haarslev and Moller, 2003). The parser component integrates parsers for WSDL and OWL-S for reading description files in each format, a different reader is needed the work of the reader component is to extract the elements out of the description, knowing its syntax and mapping it to the corresponding concept in the registry ontology.

In case of WSDL descriptions, the service reader extracts all operation parameters and documentation tags where as in OWL-S descriptions, the reader extracts the functional descriptions and also its documentation/annotation tags also.

**Ontology web registry:** The registry ontology provides a common schema to store web service description information irrespective of the syntactic details. The aim is to define a common format which can bridge between syntactic web service descriptions like WSDL and semantic web services description formats like OWL-S, WSMO, etc., document or annotation fields which are associated in large numbers with semantic service descriptions are also stored in the registry ontology. For testing and training, we have added 880 WSDL files in OWLS- TC3 and 1007 OWL-S files. The design details of the registry ontology in Description Logic (DL) syntax is shown in Algorithm 1.

**Nlp processing:** Natural Language Processing (NLP) is a field of computer science and computational linguistics concerned with interaction between computers and human languages. This area is closely associated with the field of human computer interaction. In the proposed work, we use the NLP techniques, namely, Stemming and Parts of Speech (PoS) Tagging additionally, we use stop word removing tools.

Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form. In the proposed system, the user query in processed by applying stemming and every word is reduced to its base form. Parts of Speech Tagging is the process of marking up the words in a text as corresponding to a particular parts of speech like noun, verb, adjective, etc. Our logic on converting user query to DL based description is based on parts of speech.

**Algorithem 1:**
RegistryOntology R = Thing ⊑ (Binding ⊔ Element ⊔ Input ⊔ Output ⊔ Message ⊔ Operation ⊔ Part ⊔ Port ⊔ Service ⊔ WSDL ⊔ Type ⊑ (SimpleType ⊔ ComplexType))
WSDL ⊑wsdlOf.Service
Binding ⊑ bindingOf.Port
Input ⊑ inputOf.Operation
Message ⊑ hasPart.Part
Message ⊑ messageOf.Output
Operation ⊑ hasInput.Input
Operation ⊑ hasOutput.Output
Operation ⊑ operationOf.Port
Output ⊑ outputOf.Operation
Part ⊑ hasPartType.Type
Part ⊑ part of.Message
Port ⊑ hasBining.Binding
Port ⊑ hasOperation.Operation
Port ⊑ portOf.Service
Service ⊑ hasPort.Port
Service ⊑ hasWSDL.WSDL
WSDL ⊑ wsdlOf. Service

The stop word remover tool removes insignificant words from the user query in this phase, especially, certain parts of speech like pronouns, prepositions and interjunctions are fully removed the stemmer reduces the words in the user query to its base or root form for e.g., “ing” forms are removed and plural form is reduced to singular form. Finally, the parts of speech tagger tags the words in the user query with it corresponding Parts of Speech (POS) information the tagging information along with the user query is used by the algorithms given in Fig 2-4 for constructing the unambiguous intermediate representation of the user query.

**Nl-to-dl conversion:** Natural languages queries tend to be very ambiguous in terms of lexical, syntactic and semantic (Bhattacharyya, 2012). For resolving this ambiguity, many use natural language disambiguation methods like dictionary methods (Greenberg, 2014), supervised or unsupervised methods, prepositional phase attachment (Greenberg, 2014), etc.

The proposed method aims at reducing the ambiguity in specifying user requirements incurred while describing requirements using a natural language query expression and specify user requirement in a structured form under three sections, input requirement specification, output requirement specification and subject specification. Here we convert natural language query to a structured logical form which can be easily expressed in our query description language based language which we call Web Service Query Description Language (WS-QDL). The basic syntax of the WS-QDL is given in Fig. 4.

In WS-QDL, the input and output sections consist of a set of keywords retrieved from user query using the algorithms given in Algorithm 2-5. The subject section consists of a set of statements joined by conjunction or disjunction. A query in QDL is marked under three sections, as given in Algorithm 2, namely input, output and subject which specify user requirement for input, output formats and web service task, respectively.

**Algorithm 2:**
Web Service Query Description Language (WS-QDL) Syntax
WS-QDL
INPUT : { KeyWord*}
OUTPUT : {KeyWord*}
SUBJECT : { (axiom) [PREPOSITIONS (axiom)] [AND/OR] (axiom)

The algorithm for retrieving input and output requirements from the natural language user query is given in 3 and 4. Similarly, the algorithm to retrieve the goal of the query is given in Algoritm 6.

**Algorithm 3:**

```
3 Algorithm for extracting the Input Specification
Procedure QueryParserInputSection(Query q)
q = removeStopWords(q)
array=split(q);
for i= inputSectionKeywordsIndex(array) to length(array)
if PoS(array[i])=”NOUN” then
output = output+ array[i]
end if
if PoS(array[i])=”VERB” then
output = output+array[i]+”(“
for until PoS(array[++i])=”NOUN” && i<length(array)
output = output+array[i]
end for
output =output+”)”
end if
if array[i]=[AND/OR/NOT]” then
output = output+array[i]
end if
for until PoS(array[++i])=”NOUN” && i<length(array)
output = output+array[i]
end for
output =output+”)”
end for
end procedure
```

**Algorithm for extracting the output specification 4:**

```
Procedure QueryParserOuputSection(Query q)
q = removeStopWords(q)
array=split(q);
for i= outputSectionKeywordsIndex(array) to length(array)
if PoS(array[i])=”NOUN” then
output = output+ array[i]
end if
if array[i]=[AND/OR/NOT]” then
output = output+array[i]
end if
if PoS(array[i])=”VERB” then
output = output+array[i]+”(“
for until PoS(array[++i])=”NOUN” && i<length(array)
output = output+array[i]
end for
output =output+”)”
end if
end for
end procedure
```

**Algorithm for extracting the subject specification 5:**

```
Procedure QueryParserSubjectSection(Query q)
q = removeStopWords(q)
array=split(q);
for i= subjectSectionKeywordsIndex(array) to length(array)
if PoS(array[i])=”NOUN” then
output = output+ array[i]
end if
if PoS(array[i]=”PREPOSITION” then
output = output+array[i]
end if
if PoS(array[i])=”adjective” then
output = Output = output+ array[i]
end if
if PoS(array[i])=”VERB” then
output = output+array[i]+”(“
for until PoS(array[++i])=”NOUN” && i<length(array)
output = output+array[i]
end for
output =output+”)”
end if
end for
end procedure
```

**Trainer and classification ontology generator:** Classification ontology is used to classify a user query and assign them to a particular domain. It helps to restrict the web service search to a particular domain. It may be noted that a user query may belong to more than one domain. in such case, search is extended to all the related domains. the classification ontology consists of various domains and associated keywords in hierarchical order. for generation of the classification ontology, we use OWL-S files from OWLS-TC3 and a set of manually classified user queries about 500 numbers has been introduced. The nouns from the user query is retrieved and then added to the classification ontology. To associate them with hierarchy, WordNet ontology is used. Since the WSDL files in the OWLS-TC3 does not have documentation fields, the text of documentation fields in corresponding OWL-S files were classified and added to the classification ontology. Procedure for creation and training the classification ontology is shown in Algoritm 6.

**Algorithm 6: Algorithm for creation and training the classification ontology:**

```
//Step 1.Initialize Classification ontology with the domains in the TC3-OWL dataset in level 1.
Classification ontology C = Thing ? (Communication ? Economy ? Education ? Food ? Medical ? Travel ? Weapon)
// Step 2. Processing WSDLs
for each Domain d in DataSet
for each WSDL w in d
Word_list=NULL
Word_list = RetrievePartNames(w)
Word_list = Word_list ? RetrievePartType(w)
Word_list = Word_list ? RetrieveElementNames(w)
Doc_list=RetrieveDocumentation(w)
for each doc in Doc_list
Word_list = Word_list ? Split(doc)
end for
Word_list=PoS_retrieveNouns(Word_list)
for each word wd in Word_list
for each ontology ont in TC3_OWL domain
Root_node= getSubGraphFromAncester(wd,ont)
merge(C,Root_node)
end for
end for
end for
end for
Step 3. //Adding from user queries
for each query q in file
list=split(q)
list=selectNouns(list)
for each word in list
if !found(word,C)
for each domainOntology in TCS-OWL
```

Table 1: Processing the user input query in various stages

| Stages | Description |
|---|---|
| Stage 1: User query | List some services to find the hotels available in a given village or town |
| Stage 2: Stemmer – remove stop words | List some services to find the hotels available in a given village or town |
| Stage 3:PoS Tagger | To [TO] find [VERB] hotels[noun] in [PREPOSITION] GIVEN [VERB] village [NOUN] or [Coordinating conjunction] town [NOUN] |
| Stage 4: Keyword Marker | To [KEYWORD] find [KEYWORD] hotels in [KEYWORD] GIVEN [KEYWORD] village or [KEYWORD] town |
| Stage 5a: SUBJECT MARKER | Hotel(x) IN village(y) OR Hotels(x) IN Town(z) |
| Stage 5b: INPUT MARKER | Village, Town |
| Stage 5c: OUTPUT MARKER | NIL |
| Stafe 6.WS-QDL | WS-QDL [INPUT: {Village, Town} OUTPUT: {None}' SUBJECT {Hotel(x) IN Village(y) OR Hotel(x) IN Town(z) |

```
node=subGraph(word)
if node<>NULL
merge(C,node)
exit loop
end if
end for
end if
end for
end for
```

**Query classifier:** To reduce the search space, the user query is classified into one or more domain names by relating the query terms with the domain ontology and classification ontology.

**Query optimizer:** Sometimes the user search query expression may contain words which are not available in the OWLS-TC3 domain ontology. Hence, searching of services with these words may result in producing poor results. Hence, if there is such list of unidentified words are present, synonyms of the particular word present in the domain ontology helps to find the required services. To optimize the matches, the query optimizer replaces the unmatched words in the user query with WordNet ontology and replaces the query terms with its closest synonym.

**Ws-Qdl query parser:** After the query classification process, the query is parsed to identify input, subject and output requirements, if any, specified by the user as part of the query. One or more of these fields may be empty also. The discovery process matches these requirements with the input message service documentation or port tags and output message respectively. After the discovery process, the ranking process considers the number of matches between user query requirements and service description elements in the corresponding section. The algorithms for retrieval of input, subject and output is given in Algoritm 4-6, respectively.

**Query search engine:** After domain identification, the query is processed to mark the input, output and subject part of the query. The input query part is matched with input message part/documentation in the ontology registry and output query part is matched with output message part/documentation in the ontology registry. The subject part is matched with the documentation as well as the keyword field of the web service which is present in the registry ontology.

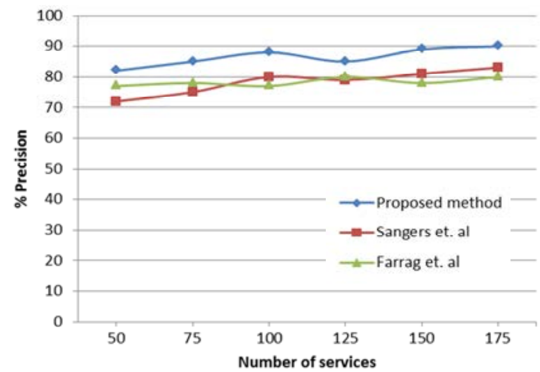


Fig. 2: Comparison of precision rate

The query search engine performs matching between the input-subject-output specifications in the web service registry with the associated specifications in our proposed WS-QDL query. If a match is found regarding the subject, then the web service is listed in results (Table 1).

This study explores the prototype implementation of the proposed system. In order to evaluate the performance of our proposed method, a proper test collection has to be used. There are some publicly available collections to evaluate the service discovery algorithms for OWL-S and SAWSDL services. Our approach is evaluated with the OWL-S Services Retrieval Test Collection (OWLS-TC v3). This collection contains 1007 OWL-S service descriptions from different domains with its equivalent WSDL service descriptions, 29 user requests in the form of queries and their corresponding set of relevant services (Fig. 2).

Farrag *et al*. (2013) to measure the effectiveness of our proposed method, we have compared the precision rate between the proposed approach and the other existing approaches such as (Farrag *et al*., 2013) and

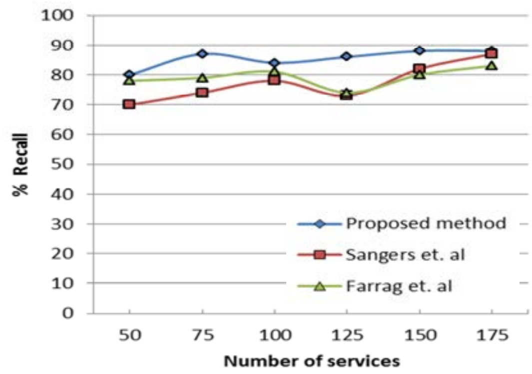


Fig. 3: Comparison of recall rate

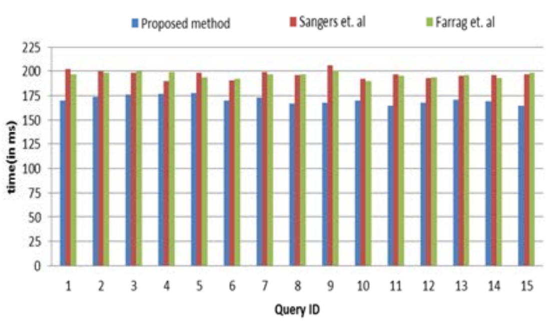


Fig. 4: Comparison of response time of query between proposed and existing methods

(Sangers *et al*., 2013). The comparison of precision rate between proposed approach and the already existing approaches. It explains that our proposed method provide improvement of 10% high precision rate than the other existing methods. Also, comparison of the recall rate between the proposed approach and other existing methods is shown in Fig. 3. The experiment shows that our proposed method ensures an improvement of 10-15% as an average of recall rate than the other existing methods. Since, the researchers have formulated the WS-QDL query from the user's request in natural language format and classified it with the help of classification ontology and WordNet, search space can be reduced to identify the relevant services. Due to this mechanism, high precision and recall can be achieved in the proposed method effectively.

The researchers have evaluated the total response time of each query in both proposed method and existing methods as shown in Fig. 4. The total response time involves the time needed to parse the query, searching services in the registry and list the relevant services. Since, the researchers have identified a set of domain according to the user's request to search the services, the total query response time of the proposed method is very less compared to other existing methods.

## CONCLUSION

Web services discovery has become a primary mechanism for efficiently consuming resources available on the internet. As numbers of services are published in the internet, automated service discovery mechanism is critical for the consumers to identify relevant and reliable services efficiently. The proposed method solves all the criticality since it accepts user query in natural language, converts them into logical expressions by using the NLP method in order to find more relevant semantic service. The user query is classified into one or more domains with the help of classification ontology and further the web service discovery is applied only on the selected domains. This approach increases the efficiency of the system. Experimental analysis shows that the precision and recall calculated by the proposed system is higher compared to the existing approaches in order to evaluate the relevance of the discovered web services.

## REFERENCES


Adala, A., N. Tabbane and S. Tabbane, 2011. A framework for automatic web service discovery based on semantics and NLP techniques. Adv. Multimedia, 2011: 1-7.

Ankolekar, A., M. Burstein, J.R. Hobbs, O. Lassila and D. Martin *et al*., 2002. DAML-S: Web Service Description for the Semantic Web. In: The Semantic Web-ISWC. Horrocks, I. and J. Hendler (Eds). Springer Berlin Heidelberg, Italy, Europe, ISBN: 978-3-540-43760-4, pp: 348-363.

Bhattacharyya, P., 2012. Natural language processing: A perspective from computation in presence of ambiguity resource constraint and multilinguality. CSI. J. Comput., 1: 1-13.

Cassar, G., P. Barnaghi and K. Moessner, 2014. Probabilistic matchmaking methods for automated service discovery. Serv. Comput. IEEE. Trans., 7: 654-666.

Farrag, T.A., A.I. Saleh and H.A. Ali, 2013. Semantic web services matchmaking: Semantic distance-based approach. Comput. Electr. Eng., 39: 497-511.

Fellbaum, C., 1998. WordNet: An Electronic Lexical Database. 2nd Edn., MIT Press, USA., ISBN: 026206197X, pp: 423.

Fenza, G., V. Loia and S. Senatore, 2008. A hybrid approach to semantic web services matchmaking. Int. J. Approximate Reasoning, 48: 808-828.

Greenberg, C., 2014. Disambiguating prepositional phrase attachment sites with sense information captured in contextualized distributional data. Proceedings of the ACL 2014 Student Research Workshop June 22-27, 2014, ACL, Baltimore, Maryland, USA., pp: 71-77.

Gunasri, R. and R. Kanagaraj, 2014. Natural language processing and clustering based service discovery. Intl. J. Technol. Enhancements Emerging Eng. Res., 3: 28-31.

Haarslev, V. and R. Moller, 2003. Racer: An owl reasoning agent for the semantic web. Proceedings of the International Workshop on Applications, Products and Services of Web-based Support Systems, October 13, 2003, IEEE, Halifax, Canada, pp: 91-95.

Indurkhya, N. and F.J. Damerau, 2010. Handbook of Natural Language Processing. 2nd Edn., CRC Press, Boca Raton, Florida, ISBN-13:978-1-4200-5893-8, Pages: 679.

Jurafsky, D. and J.H. Martin, 2008. Speech and Language Processing: An Introduction to Speech Recognition, Computational Linguistics and Natural Language Processing. 2nd Edn., Prentice Hall, New York, pp: 1024.

Miller, G.A., 1995. WordNet: A lexical database for English. Commun. ACM, 38: 39-41.

Paliwal, A.V., B. Shafiq, J. Vaidya, H. Xiong and N. Adam, 2012. Semantics-basedautomated service discovery. Serv. Computing IEEE. Trans., 5: 260-275.

Sangers, J., F. Frasincar, F. Hogenboom and V. Chepegin, 2013. Semantic web service discovery using natural language processing techniques. Expert Syst. Appl., 40: 4660-4671.

Sangers, J., F. Frasincar, F. Hogenboom, A. Hogenboom and V. Chepegin, 2012. A linguistic Approach for Semantic Web Service discovery. In: Management Intelligent Systems. Casillas, J., F.J. Martinez-Lopez and J.M.C. Rodriguez (Eds.). Springer Berlin Heidelberg,Berlin,Germany,ISBN: 978-3-642-30863-5, pp: 131-142.

Sireteanu, A.N., 2013. A survey of web ontology languages and semantic web services. Annal. Alexandru Ioan Cuza Univ. Econ., 60: 42-53.