

Research Article

A Framework for Automatic Web Service Discovery Based on Semantics and NLP Techniques

Asma Adala, Nabil Tabbane, and Sami Tabbane

Multimedia Mobile Radio Networks Research Unit (MEDIATRON), Higher School of Communication of Tunis (Sup'Com), University of Carthage, City of Communication Technologies, 2083 Ariana, Tunisia

Correspondence should be addressed to Asma Adala, asma.adala@gmail.com

Received 4 November 2011; Accepted 30 December 2011

Academic Editor: Tai Hoon Kim

Copyright © 2011 Asma Adala et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As a greater number of Web Services are made available today, automatic discovery is recognized as an important task. To promote the automation of service discovery, different semantic languages have been created that allow describing the functionality of services in a machine interpretable form using Semantic Web technologies. The problem is that users do not have intimate knowledge about semantic Web service languages and related toolkits. In this paper, we propose a discovery framework that enables semantic Web service discovery based on keywords written in natural language. We describe a novel approach for automatic discovery of semantic Web services which employs Natural Language Processing techniques to match a user request, expressed in natural language, with a semantic Web service description. Additionally, we present an efficient semantic matching technique to compute the semantic distance between ontological concepts.

1. Introduction

A lot of web services are being offered nowadays, and this trend is going to continue in the future. A demand increases consequently for an automatic discovery framework of services that are highly relevant to user requirements.

The widespread adoption of Web services is enabled by a set of flexible and extensible XML-based standards such as WSDL [1], UDDI [2], and SOAP [3]. However, these current XML-based specifications provide only syntactical descriptions of the functionality provided by Web services and therefore still require human interaction especially during discovery process. Thus, a more reliable and effective Web service discovery approach, that is suitable for automatic processing, is needed.

The Semantic Web [4] vision has encouraged researchers to enrich existing Web services descriptions with machine-interpretable semantics, called semantic Web services, in order to automate related Web services core tasks such as discovery, composition, selection and invocation. The objective of semantic Web service technology is to minimize the manual discovery and usage of Web services, by allowing software agents and applications to automatically identify, integrate,

and execute these Web resources to achieve the user objectives.

Many approaches for automatic Web service discovery have been proposed, as discussed in Section 4. However, they present several major limitations. First, some proposed discovery frameworks are based on a user request that is expressed in a specific semantic description language like OWL-S [5], WSMO [6], or WSDL-S [7]. As a result, they require the end user to have intimate knowledge of semantic Web services and related description and implementation details which makes their usage difficult for end users. Second, the discovery scope of these approaches is often limited to some Web services that are published in a specific description standard. The most prominent semantic Web services frameworks are based on OWL-S or WSMO standards. This limitation is impractical since it expects all advertised services to have semantic tagged descriptions, especially that descriptions of the vast majority of already existing Web services are specified using Web Services Description Language (WSDL) and do not have associated semantics. Furthermore, it makes assumption about the used service description language which would limit the discovery process to specific advertised services.

Also, from the service requestor's perspective, the requestor may not be aware of all the knowledge that constitutes the domain ontology. Specifically, the service requestor may not be aware of all the terms related to the service request. As a result of which many services relevant to the request may not be considered in the service discovery process.

Another limitation of some proposed framework consists on their semantic matching approaches. In fact, both service provider and service requester use domain ontologies to build semantic service description file. The semantic matchmaker uses the domain ontologies from two sides to determine their degree of semantic match. Most of proposed approaches assume that both service provider and service requester use the same ontology domain to describe service capabilities which is not applicable in real-world scenario. To overcome this ontology heterogeneity, it is needed to utilize ontology mapping techniques to coordinate the differences between these ontologies to support interoperability.

In order to address the cited limitations of existing approaches, we first propose a discovery framework based on a user query expressed in natural language. Then, we perform query preprocessing using Natural Language Processing (NLP) techniques in order to extract keywords from the user query. Compared with formal queries, keyword-based queries have many advantages. They offer a simple syntax in terms of a list of keyword phrases and open vocabularies wherein the users can use their own words to express their information requirement. Also, keyword-based search is more familiar to the user due to its widespread usage (e.g., Search Engines, UDDI registries).

However, creating a semantic Web service discovery engine using a keyword-based approach can be a complex task. In fact, many issues should be considered in order to answer these questions.

- (i) How to extract the most relevant information from a semantic Web service description?
- (ii) How to match keywords from the user query with textual information from a semantic Web service description?
- (iii) How to map English words to ontological concepts in order to perform semantic matching?

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 WSDL or in any semantic Web service description language like OWL-S or WSMO. Finally, to overcome the ontology heterogeneity problem, our proposed framework employs some Natural Language (NLP) techniques to extract senses from user keywords and Web service descriptions. It also contains a mapping module which converts English terms present in WordNet [8, 9] lexical database to Suggested Upper Merged Ontology (SUMO) [10].

The remainder of the paper is structured as follows. We present the related work in Section 2. In Section 3, we provide a background material that is essential to understand the presented approach. Section 4 presents in details our

proposed discovery framework and its different modules. In Section 5, we present some conclusions.

2. Related Work

Many research efforts have been made to present a discovery framework for Web services. They are generally devised into syntactic-based approaches and semantic based approaches. The major differences between these two approaches are summarized in Table 1. The syntactic-based search engines are usually based on WSDL Web services descriptions published in UDDI. One example is the search eSynaps [11] engine. Seekda! [12] tries to go further, by extracting semantics from the WSDL files, which enables runtime exchange of similar services and composition of services. Seekda! has not yet searched through existing semantic Web service description files, but only has made use of the WSDL file of a Web service.

The semantic-based approaches utilize semantic description for Web services to automate the discovery process and employ the Semantic Web techniques. GODO [13], for example, is a Goal-Driven approach for searching WSMO Web services. It consists of a repository with WSMO Goals and lets users state their goal by writing a sentence in plain English. A language analyzer will extract keywords from the user sentence and a WSMO Goal will be searched based on those keywords. The WSMO Goal with the highest match will be sent to WSMX, an execution environment for WSMO service discovery and composition. WSMX will then search for a WSMO Web service that is linked to the given WSMO Goal via some WSMO Mediators and return the WSMO Web service back to the user. This approach makes good use of the capabilities of the WSMO framework, but it cannot be applied for other semantic languages like OWL-S, which do not have such goal representation elements.

Sycara et al. introduced LARKS [14] for describing agent capabilities and requests, and their matchmaking. The discovery/matching engine of the matchmaker agent is based on various filters of different complexity and accuracy which users can choose. However, the model lacks in defining how service requests will be specified by users. Also, LARKS assumes the existence of a common basic vocabulary for all users.

METEOR-S discovery [15] framework addresses the problem of discovering services in a scenario where service providers and requesters may use terms from different ontologies. Their approach relies on annotating service registries (for a particular domain) and exploiting such annotations during discovery.

3. Background

In this section, we describe some concepts definitions and methodologies utilized in our framework. We first present some semantic Web related technologies. Then, we briefly describe some Natural Language Processing (NLP) techniques utilized in our approach in order to process a user query written in natural language and Web services descriptions

TABLE 1: Syntactic versus semantic approaches for Web services discovery.

	Syntactic-based approaches for WS discovery	Semantic-based approaches for WS discovery
Matchmaking technique	(i) A simple keyword-based search (ii) Searching based on functional parameters (ii) Searching based on syntax	(i) Exploit the semantic representation of concepts describing a Web Service and their relations in an ontology (ii) Searching based on both functional and non-functional parameters
Advantages	(i) Simple and widely used technique. (ii) Standards like UDDI exist	(i) Minimize the manual discovery and usage of Web service by allowing software agents to automatically and dynamically discover WSS (ii) Pledge the automation of WS discovery process (iii) Effective and reliable technique
Disadvantages	(i) Do not allow retrieval of Web Services with similar functionality (ii) Not suited for automatic processing (iii) Still requires human interaction	(i) More complex technique (ii) Semantic tagging of Web services may be needed

before performing semantic matchmaking. We finally present an overview about WordNet and SUMO projects.

3.1. Ontology. An ontology is an explicit shared specification of various conceptualization in a particular domain. It plays a vital role in the semantic Web and tries to capture the semantics of a domain by deploying knowledge representation primitives, enabling a machine to understand the relationships between concepts in a domain.

Because some relations and axioms, ontology can be reasoned availablely, therefore we can express the semantics of a concept by establishing the complex relationship among other concepts, attributes, and instances. Domain ontology is a detailed description of the hierarchical concepts of the field. It abstracts and conceptualizes objects, relationship, and class to be expressed as a vocabulary. The sets of glossary in the vocabulary are concepts. Ontology is a detailed description of the world's conceptualization. Domain ontology is the sets of all concepts from the domain. In the actual application, people always build domain ontology in their respective fields (e.g., travel ontology, communication ontology, and medical ontology).

3.2. Ontology Languages. A number of ontology's description languages have been proposed to address the semantic heterogeneity among Web resources and services. However, OWL is considered as a major technology for the future implementation of a Semantic Web since it is based on XML so OWL information can be easily exchanged between different types of computers using different operating systems and application languages.

The Web Ontology Language (OWL) [16] is a language to define and instantiate Web ontologies. It was formerly called DAML+OIL language. OWL ontology may include descriptions of classes, along with their related properties and instances. OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans. It facilitates greater machine interpretability of Web content than that supported by XML, Resource Description Framework (RDF), and RDF Schema by providing additional vocabulary along with a formal semantics [17]. OWL has three sublanguages:

OWL-Lite, OWL-DL, and OWL-Full. These three increasingly expressive sublanguages are designed for use by specific communities of implementers or users [16].

3.3. Web Service Description Languages. Traditional Web services are described using XML-based standards and published into a specific registry standard.

3.3.1. WSDL. WSDL [1] is an XML format for describing network services in abstract terms derived from the concrete data formats and protocols used for implementation. However, WSDL does not support semantic description of services. For example, it does not support the definition of logical constraints between its input and output parameters although it has the concept of input and output types as defined by XSD.

3.3.2. UDDI. UDDI [2] is a well-known Web service repository. The UDDI specification consists of a programmer's API along with an XML Schema definition of supporting data structures and messages. UDDI repositories contain information about *businesses*, *services*, and *service bindings* as well as additional metadata for categorization purposes. However, UDDI does not represent service capabilities. It uses *tModels* to provide a tagging mechanism. Searching for a service in an UDDI is performed by string matching on some defined fields. Thus, it is unsuitable for locating services on the basis of a semantic specification of their functionality.

3.4. Semantic Web Service Description Languages. Semantic Web services are services that have been enriched with machine-interpretable semantics. Semantic description aims to enhance the integration and Web service discovery by utilizing the machine readable constructs of the representation.

Several standards have been proposed for creating semantic Web services. Each one of them is having their own strength and can be used in a specific situation. Some of the popular languages are described as follows.

3.4.1. OWL-S. OWL-S [5] is an OWL-based Web service ontology, which supplies Web service providers with a core set of markup language, constructs for describing the properties,

and capabilities of their Web services in unambiguous and computer interpretable form. An OWL-S description is composed of three parts which are Service Profile, Service Model, and Service Grounding. The Service profile describes service capabilities and it is the part used in the discovery process. The Service Model describes how the service works (internal processes), and the Service Grounding specifies the details of how the service can be accessed.

3.4.2. *WSMO*. WSMO [6] provides a conceptual framework and a formal language to describe all relevant aspects of Web services to facilitate the automation of service discovery using semantics. The overall structure of WSMO is divided into four main elements [6].

- (i) Ontologies: provides the terminology used by other WSMO elements.
- (ii) Web service descriptions: describes the functional and behavioral aspects of a Web service.
- (iii) Goals: represents user desires.
- (iv) Mediators: aims to automatically handle interoperability problems between different WSMO elements.

3.4.3. *WSDL-S*. Current WSDL standard operates at the syntactic level and lacks the semantic expressivity needed to represent the requirements and capabilities of Web Services [18]. WSDL-S [7] is a lightweight approach for adding semantics to Web services. In WSDL-S, the semantic models are maintained outside of WSDL documents and are referenced from the WSDL document via WSDL extensibility elements.

3.5. *NLP*. Natural Language processing (NLP) [19, 20] is a field of computer science and linguistics concerned with the interactions between computers and human (natural) languages. NLP is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things. In theory, natural-language processing is a very attractive method of human-computer interaction. NLP has significant overlap with the field of computational linguistics and is often considered a subfield of artificial intelligence.

In our work, we employ some NLP techniques which are presented as follows.

- (i) Word splitting: is the process of parsing concatenated text (i.e., text that contains no spaces or other word separators) to infer where word breaks exist.
- (ii) Stemming: is the process for reducing inflected (or sometimes derived) words to their stem, base, or root form. For example, a stemming algorithm reduces the words “fishing”, “fished”, “fish,” and “fisher” to the root word, “fish”.
- (iii) Part Of Speech (POS) tagging: is the process of marking up the words in a text (corpus) as corresponding to a particular part of speech, based on both its definition as well as its context. A POS tagger enables the identification of words as nouns, verbs, adjectives, adverbs, and so forth.

- (iv) Word Sense Disambiguation (WSD): the process of identifying which sense of a word (i.e., meaning) is used in a sentence, when the word has multiple meanings (polysemy).

3.6. *WordNet*. WordNet [8, 9] is an electronic lexical database for the English language realized at Princeton University by George Miller’s team and based on psycholinguistic theories. In WordNet, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (*synsets*), each expressing a distinct concept. Synsets are interlinked by means of conceptual semantic and lexical relations.

WordNet is of interest not only because it is a vast repository of lexical data, but also because it is so widely used. It has been leveraged for automated sense-disambiguation, term expansion in IR systems, and the construction of structure representations of document content. In fact, WordNet is so popular that it is almost considered a de facto standard in the NLP community.

3.7. *SUMO*. SUMO (Suggested Upper Merged Ontology) [10] is an ontology that was created at Teknowledge Corporation with extensive input from the SUO (Standard Upper Ontology) [21] mailing list and it has been proposed as a starter document for the IEEE-sanctioned SUO Working Group.

The SUMO was created by merging publicly available ontological content into a single, comprehensive, and cohesive structure.

4. Proposed Framework

In this section, we present our discovery framework presented in Figure 1. We give detailed description about our proposed keyword-based discovery approach for searching Web services which are described using a syntactic or a semantic language and advertised in a Web service registry. This search mechanism incorporates natural language processing techniques to establish a match between a user search query, containing English keywords, and a Web service description. The overall process is modeled in a sequence diagram expressed in UML (Unified Modeling Language) standard and presented in Figure 2.

4.1. *Framework Architecture*. Our discovery process aims to enable efficient search for appropriate Web services according to a user query. In our proposed discovery framework, we suppose that there is a set of Web services described in WSDL, OWL-S, or WSMO languages and published by service providers in a Web service registry. These descriptions are parsed and read by our system in order to extract all useful information elements for the matchmaking process. Some NLP techniques are then applied to extracted information to find useful words for next steps. As words could have different senses, a sense disambiguation is performed. In order to map each word to its corresponding concept in SUMO ontology, a WordNet/SUMO mapping is carried out. The final process in the framework is semantic matchmaking.

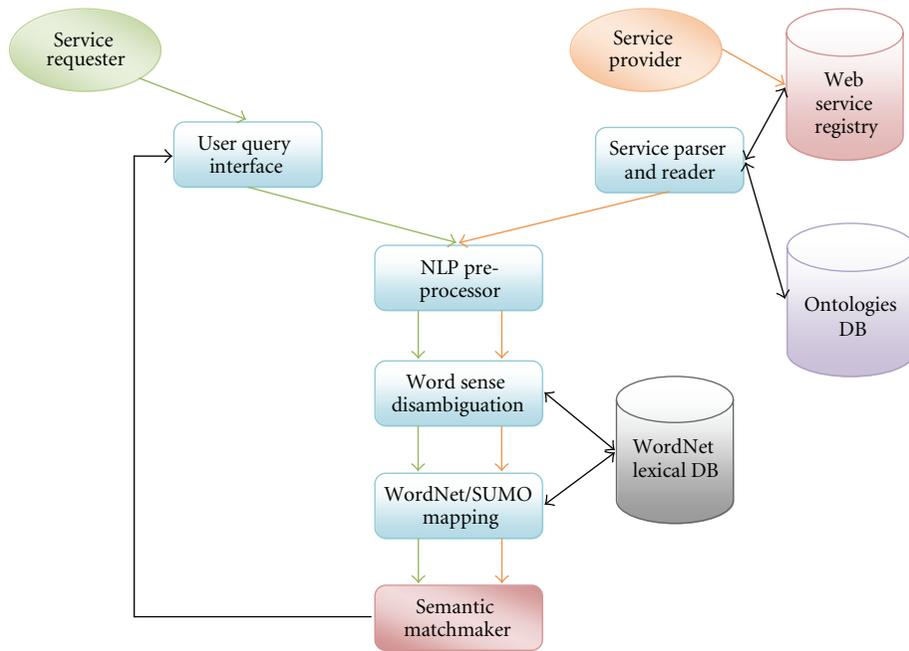


FIGURE 1: Automatic Web service discovery framework architecture.

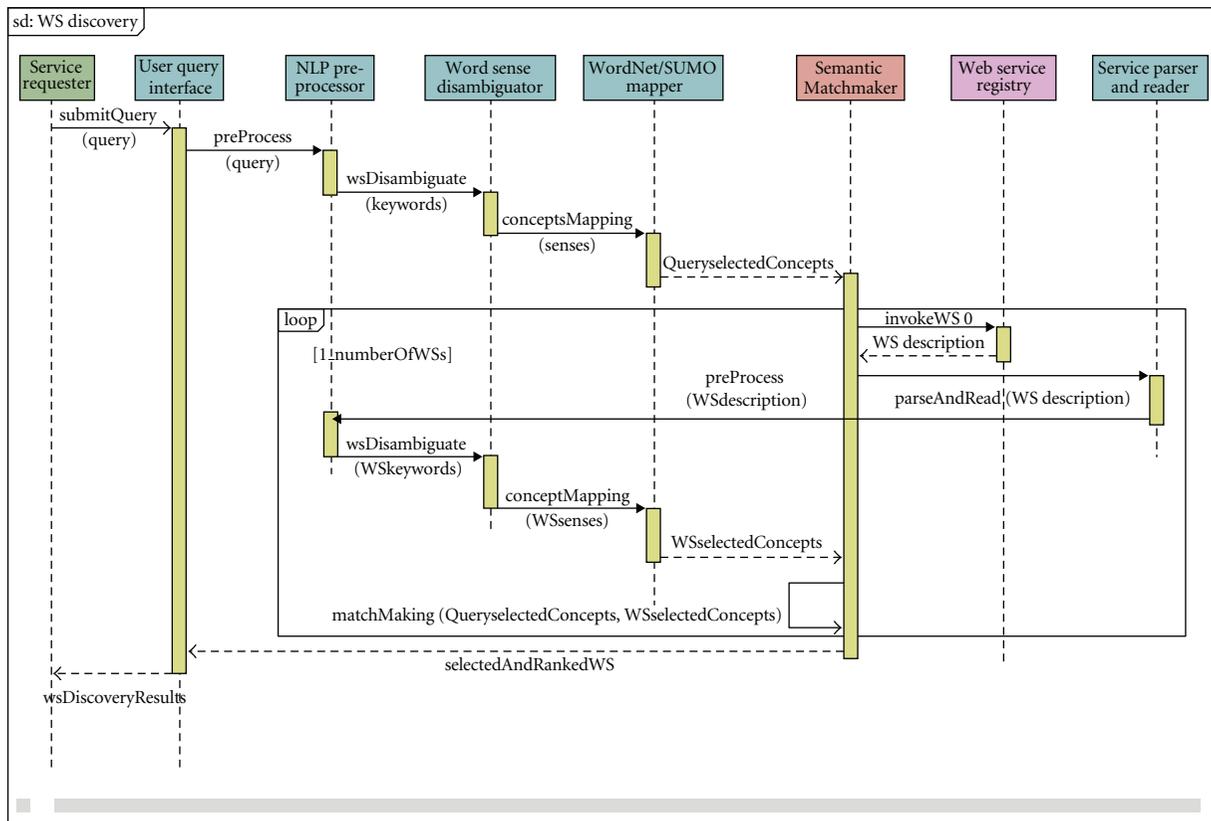


FIGURE 2: Sequence diagram for use case "discovery of Web services."

It is based on calculating semantic distance between concepts defined in ontology.

From the service requester point of view, our system offers a simple graphical user query interface to facilitate the discovery process. Therefore, the framework has as input a query expressed in natural language, from which useful keywords are extracted. Consequently, the overall architecture and implemented technologies are transparent to user. The user query must be also preprocessed to be matched to service description using the same processes as service description. Finally, the concepts mapped to the senses disambiguated from the search query are matched with the concepts mapped to the senses disambiguated from Web service description.

4.2. Service Parser and Reader. As presented in Sections 3.3 and 3.4, there exist many Web service description languages. For each service annotation, a different reader is needed. A reader must be able to extract elements out of a Web service description and of its used ontologies in case of semantic annotations.

In the case of OWL-S or WSMO Web service, names and nonfunctional descriptions of elements such as the capabilities (inputs/outputs), conditions, and effects of the Web service should be extracted by the service reader. After extracting concepts out of those elements, the service reader searches for their nonfunctional descriptions in the relevant ontology which is extracted from the ontologies database.

In the case of WSDL description, the service reader extracts operations parameters (all terms under <element name> and <documentation> tag).

Before extracting words from a Web service description, the description has to be parsed. Different languages can represent different syntaxes and therefore different parsers are needed. For example, for WSMO a WSML parser like WSMO4J [22] can be used. Sesame [23] and Jena [24] are examples of parsers for OWL-S.

4.3. Service and Query Preprocessor. Web service description must be preprocessed in order to transform extracted elements into useful words that could be processed later. User query must be also pre-processed to extract useful keywords from a query written in natural language. For pre-processing, some NLP techniques are utilized.

First, word segmentation is performed if needed to split a string of written language into its component words. The white space is a good approximation of a word delimiter. In the case of element names, simply splitting the words when a case transition has occurred is enough, since in most cases they are written as camel words (e.g., TravelCheckingService). To find useful words for WSD, each word in the sentences found must be tagged with the right Part-of-Speech (PoS) such as noun, verb, and adjective. Markups and punctuations are then removed. Translation of uppercase characters into lowercase is also needed. Second, all stop words are removed from extracted elements. Stemming is finally processed to transform obtained words to root words.

4.4. Word Sense Disambiguation. The Word Sense Disambiguation module establishes the context of words received from the preprocessor by extracting relevant senses. This will result in a set of senses, each representing a single meaning of a word. In general terms, WSD involves the association of a given word in service description or in user request with a definition or meaning (*sense*) which is distinguishable from other meanings potentially attributable to that word. The task therefore necessarily involves two steps: (1) the determination of all the different senses for every word and (2) a means to assign each occurrence of a word to the appropriate sense.

In our approach, we use a variant of the SSI algorithm [25] to get the senses out of a set of words as it is shown by (1). The algorithm disambiguates a word (*word*) based on a previously disambiguated set of words and their related senses. Per sense of the word (*sj*), a similarity with the senses from the context (*sci*) is calculated and the sense with the highest similarity is chosen. After that, the word and its chosen sense will be added to the context (*I*) and iteration will be done. This process continues until there are no ambiguous words left

$$\text{selected Sense (lex)} = \arg \underset{sj \in \text{senses (word)}}{\text{Max}} \sum_{sci \in I} \text{sim}(sj, sci). \quad (1)$$

At the start of the process, a context is not yet established. In order to disambiguate meanings of the words that can have multiple senses, one first has to find the words that have only one sense (monosemous words) to initialize the context. If all the words in the set have multiple senses (polysemous words), the least ambiguous word is chosen and for each of its senses, the algorithm is simulated as if the sense was used as the starting context. Each time a new sense is added to the context, the similarity between the new sense and the context is stored. The sense which creates the highest sum of similarity measures during its simulation is used for the context initialization.

The similarity function (*sim*) is defined in Section 3.6.

4.5. WordNet/SUMO Mapping. The mappings between WordNet and the SUMO can be regarded as a natural language index to the SUMO. It presents a tool which permits the user to enter English terms and which returns SUMO concepts that are associated with the input terms via WordNet synsets. The WordNet/SUMO mapping module offers the capability to assign the structured meanings of the SUMO to free text. In fact, all extracted senses from WSD module are matched to the equivalent concept in SUMO ontology. Thus, semantic matchmaking could be applied to user query-related concepts with service-description-related concepts.

4.6. Semantic Matchmaker. A basic step toward semantic matchmaking is to calculate the semantic distance between concepts that are defined in an ontology. In the semantic matchmaking module, we utilize a novel edge-based approach to measure the semantic distance between two ontological concepts which is presented in details in our previous

work [26]. The edge is the direct semantic relation between two concepts in the ontology. In our proposed approach, the semantic distance between two concepts is a function σ of edges weights values along the path between two concepts. An edge's weight depends on two parameters that are the depth of the parent node (super concept) in the hierarchy ($d(p)$) and the local density of the parent node ($E(p)$). This semantic distance function is defined in

$$\sigma(c_1, c_2) = \sum_{c \in \{\text{path}(c_1, c_2) \setminus \text{LS}(c_1, c_2)\}} \text{wt}(c, p(c)). \quad (2)$$

The calculation of an edge weight is expressed by

$$\text{wt}(c, p) = \left(\beta + (1 - \beta) \frac{\bar{E}}{E(p)} \right) \left(\frac{d(p) + 1}{d(p)} \right)^\alpha. \quad (3)$$

The semantic matchmaker has in input two sets of SUMO concepts. One set represents the user query and the other represents the service description.

Equation (4) is applied to calculate the final semantic matching degree between the two sets of concepts

$$\text{MatchD}(S_1, S_2) = \begin{cases} \text{Max}_{c_1 \in S_1} \left\{ \text{Min}_{c_2 \in S_2} \{ \sigma(c_1, c_2) \} \right\}, & S_1 \neq \varphi \wedge S_2 \neq \varphi, \\ 0, & S_1 = \varphi \vee S_2 = \varphi. \end{cases} \quad (4)$$

5. Conclusion

The work proposed in this paper provides an approach for automatic discovery of Web services.

We lay stress on the fact that, since users often have little knowledge about Web-service-related technologies and implementation details, a discovery framework that has a user query expressed in natural language as input is needed. Our proposed framework presents a discovery mechanism that enables Web-service-discovery-based on keywords written in natural language with no constraints about the used Web service description language. We presented a novel approach which takes advantages from keyword-based search simplicity and from Semantic web emergent technologies to automate the discovery process of Web services.

Some of our work in progress is aimed at extending our approach to service discovery, to support service invocation and workflow composition.

References

- [1] W3C Web Services Description Language, <http://www.w3.org/TR/wsdl>.
- [2] Universal Description Discovery and Integration, <http://uddi.xml.org/uddi-org>.
- [3] SOAP Version 1.2. W3C Recommendation, 2007, <http://www.w3.org/TR/soap12-part0/>.
- [4] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, 2001, <http://www.w3.org/2001/sw>.
- [5] OWL-S: Semantic Markup for web services, OWL-white paper, <http://www.ai.sri.com/daml/services/owl-s/1.2/overview/>.
- [6] J. D. Bruijn, C. Bussler, J. Domingue et al., "Web Service Modeling Ontology (WSMO)," <http://www.w3.org/Submission/WSMO/>.
- [7] K. Sivashanmugam, K. Verma, A. Sheth, and J. Miller, "Adding semantics to web services standards," in *Proceedings of the International Conference on Web Services*, pp. 395–401, Las Vegas, Nev, USA, June 2003.
- [8] G. A. Miller, "WordNet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [9] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, Mass, USA, 1998.
- [10] I. Niles and A. Pease, "Linking lexicons and ontologies: mapping wordnet to the suggested upper merged ontology," in *Proceedings of the International Conference on Information and Knowledge Engineering (IKE '03)*, pp. 412–416, Las Vegas, Nev, USA, June 2003.
- [11] eSynaps: eSynaps Web Service Search, 2009.
- [12] Semantic Technology Institute: Seekda!, 2009.
- [13] J. M. Gomez, M. Rico, F. Garcia-Sanchez, R. M. Bejar, and C. Bussler, "GODO: goal driven orchestration for semantic web services," in *Proceedings of the Workshop on Web Services Modeling Ontology Implementations*, vol. 113, CEUR Workshop Proceedings, 2004.
- [14] K. Sycara, S. Widoff, M. Klusch, and J. Lu, "LARKS: dynamic matchmaking among heterogeneous software agents in cyberspace," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 2002.
- [15] S. Oundhakar, K. Verma, K. Sivashanmugam, A. Sheth, and J. Miller, "Discovery of web services in a multi-ontology and federated registry environment," *International Journal of Web Services Research*, vol. 2, no. 3, pp. 1–32, 2005.
- [16] Y. Alsafadi, J.-F. B. J. Barnette, S. Bechhofer et al., "OWL Web Ontology Language Guide," 2004, <http://www.w3.org/TR/owl-guide/>.
- [17] D. L. McGuinness and F. V. Harmelen, "OWL Web Ontology Language Overview," 2004, <http://www.w3.org/TR/owl-features/>.
- [18] M. Herrmann, M. A. Aslam, and O. Dalferth, "Applying semantics (WSDL, WSDL-S, OWL) in service oriented architectures (SOA)," in *Proceedings of the 10th International Protégé Conference*, Budapest, Hungary, 2007.
- [19] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2008.
- [20] N. Indurkha and F. Damerau, *Handbook of Natural Language Processing*, CRC Press, Taylor and Francis Group, 2nd edition, 2010.
- [21] Standard Upper Ontology, <http://suo.ieee.org/>.
- [22] EU IST, FIT-IT: WSMO4J API, 2008, <http://wsmo4j.sourceforge.net/>.
- [23] Sesame, 2009, <http://www.openrdf.org/>.
- [24] HP Labs Semantic Web: Jena, 2008, <http://jena.sourceforge.net>.
- [25] R. Navigli and P. Velardi, "Structural semantic interconnections: a knowledge-based approach to word sense disambiguation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1075–1086, 2005.
- [26] A. Adala, N. Tabbane, and S. Tabbane, "An edge-based approach for semantic matchmaking of service capabilities," *Journal of Computer Technology and Application*, vol. 2, no. 8, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

