NRC Publications Archive Archives des publications du CNRC

A weighted ontology-based semantic similarity algorithm for web services

Liu, M.; Shen, W.; Hao, Q.; Yan, J.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. / La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

For the publisher's version, please access the DOI link below. / Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

Publisher's version / Version de l'éditeur:

<https://doi.org/10.1016/j.eswa.2009.04.034>

International Journal of Expert Systems and Applications, 36, 10, pp. 12480-12490, 2009-12-01

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=d5a3cf5c-5bde-410f-b25a-14352daf46c3>

<https://publications-cnrc.canada.ca/fra/voir/objet/?id=d5a3cf5c-5bde-410f-b25a-14352daf46c3>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

<http://www.nrc-cnrc.gc.ca/irc>

A weighted ontology-based semantic similarity algorithm for web services

NRCC-52688

Liu, M.; Shen, W.; Hao, Q.; Yan, J.

January 2010

A version of this document is published in / Une version de ce document se trouve dans:
International Journal of Expert Systems and Applications, 36, (10), pp. 12480–12490, December 01, 2009, DOI: [10.1016/j.eswa.2009.04.034](https://doi.org/10.1016/j.eswa.2009.04.034)

The material in this document is covered by the provisions of the Copyright Act, by Canadian laws, policies, regulations and international agreements. Such provisions serve to identify the information source and, in specific instances, to prohibit reproduction of materials without written permission. For more information visit <http://laws.justice.gc.ca/en/showtdm/cs/C-42>

Les renseignements dans ce document sont protégés par la Loi sur le droit d'auteur, par les lois, les politiques et les règlements du Canada et des accords internationaux. Ces dispositions permettent d'identifier la source de l'information et, dans certains cas, d'interdire la copie de documents sans permission écrite. Pour obtenir de plus amples renseignements : <http://lois.justice.gc.ca/fr/showtdm/cs/C-42>

National Research
Council Canada

Conseil national
de recherches Canada

Canada

A weighted ontology-based semantic similarity algorithm for web service

Min Liu¹, Weiming Shen², Qi Hao², Junwei Yan¹

¹*CIMS Research Center, Tongji University, Shanghai 200092, P.R. China*

²*National Research Council Canada, London, Ontario N6G 4X8, Canada*

Lmin@mail.tongji.edu.cn; weiming.shen@nrc-cnrc.gc.ca

Abstract

A critical step in the process of reusing existing WSDL-specified services for building web-based applications is the discovery of potentially relevant services. However, a category-based service discovery, such as UDDI, is clearly insufficient. Semantic Web Services, augmenting Web service descriptions using Semantic Web technology, were introduced to facilitate the publication, discovery, and execution of Web services at the semantic level. Semantic matchmaker enhances the capability of UDDI service registries in the Semantic Web Services architecture by applying some matching algorithms between advertisements and requests described in OWL-S to recognize various degrees of matching for Web services. Based on the Semantic Web Service framework, semantic matchmaker and a probabilistic matching approach, this paper proposes a weighted ontology-based semantic similarity algorithm for web service to support a more automated and veracious service discovery and rank process in the Semantic Web Service Framework.

Keywords: Semantic Web Services, Semantic Similarity, Degree of Matching, Rank

1. Introduction

A critical step in the process of reusing existing WSDL (Web Service Description Language) - specified services for building web-based applications is the discovery of potentially relevant services [1]. UDDI (Universal Description, Discovery and Integration) servers are essentially

catalogs of published WSDL specifications of available services. These catalogs are organized according to categories of business activities. Providers advertise web services by adding their WSDL specifications to the appropriate UDDI directory category. Through a well-defined API (Application Programming Interface), service requesters can browse the UDDI directory by category to discover existing services potentially relevant to what they want to query.

However, this category-based service discovery is clearly insufficient, because it relies on the shared common-sense understanding of the application domain by the developers who publish and request the specified services, and it is the responsibility of the provider to publish the services in the appropriate UDDI category. Conversely, the requester must browse the “right” categories to discover the potentially relevant services and to assess which of the discovered candidate services are more likely to be useful in their systems.

Semantic Web Services (SWS) [2, 3], augmenting Web service descriptions using Semantic Web technology, were introduced to address the above problem and to facilitate the autonomous publication, discovery, and execution of services at the semantic level. Moreover, semantic Web service description languages, such as OWL-S (Web Ontology Language Schema) [4, 5] and WSMO (Web Service Modeling Ontology) [6], were proposed as abstractions of syntactic Web service description languages such as WSDL. OWL-S describes the categories, the inputs, the outputs and the consequences of Web services in terms of concepts defined in OWL ontology, and it also provides the grounding constructs for specialization into WSDL constructs for compatibility with existing Web services.

Furthermore, to support programmatic service discovery, Semantic matchmaker[7~9], which are the software agents that accept and keep track of the descriptions of available services from providers and match them against the requirements from service requesters [3], enhances the capability of UDDI service registries and web services locations in the Semantic Web Services architecture, and applies some matching algorithms between advertisements and requests

described in OWL-S to recognize and to rank various degrees of matching for web services.

In this paper, we propose a weighted ontology-based semantic similarity algorithm for web service to support a more automated and veracious service discovery and rank process, by distinguishing between the potentially useful and the likely irrelevant services and by ordering the potentially useful ones according to their relevance to the requester's query.

The remainder of the paper is organized as follows: Section 2 discusses the relative work; Section 3 gives the definition of web service ontology and describes the graph-based interface properties of web service; Section 4 designs in detail the weighted ontology-based semantic similarity algorithm; Section 5 explains the matching & ranking implement of semantic similarity and relative matching algorithm in the Semantic Web Service framework; finally, section 6 summarizes the contributions of this work and outlines some issues it has raised for future consideration.

2. Related Work

The problem of service matching and discovery is analogous to the problem of information retrieval and component retrieval [10]. First, a WSDL specification declares 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. Second, a WSDL specification usually includes a set of natural-language descriptions of the service and its elements. Therefore, a semantic information-retrieval method can be used to identify and order the most relevant WSDL specifications based on the similarity of their element descriptions with the current query. The majority of information-retrieval methods are based on the probabilistic vector-space model [11, 12]. In the model, documents are represented as t -dimensional vectors, where each element of the vector corresponds to one of the distinct words contained in the document. The vectors are usually constructed after preprocessing that eliminates stop words (i.e. commonly used words that are unlikely to be characteristic of any document's content) and stems the documents' words so that

related words with a common stem are considered as a single word. Each term in the vector is assigned a weight that reflects the importance of this term in the document. The weight value is proportional to the frequency with which the term appears in the document represented by the vector and inversely proportional to the number of documents that contain this term [12]. A common term importance indicator is the term-frequency inverse-document-frequency $tf - idf$ ranking [11] in probability, according to which, the importance of a word i in document j is

$$w_{ij} = tf_{ij} * idf_i = tf_{ij} * \log_2\left(\frac{N}{df_i}\right)$$

Where, tf_{ij} is the frequency of term i in document j ; idf_i is the inverse document frequency of term i ; N is the total number of documents in the collection; and df_i is the number of documents containing the term i .

Queries are also represented as vectors. The similarity between a document vector d , and a query vector q , can then be computed as the vector inner product:

$$Sim(d, q) = d \bullet q = \sum_{i=1}^t W_{id} * W_{iq}$$

In the above equation, W_{id} and W_{iq} are the weight of term i . A higher similarity score indicates a closer similarity between the query and the retrieved documents.

The major shortcoming of the vector-space model methods arises from the fact that each word does not have a unique, unambiguous meaning - homonyms have the same spelling but different meanings - and that many words have synonyms, i.e. different words that have the same meaning. As a result, two documents may have two similar vector representations and still contain very diverse subject matter.

Signature matching [13] and specification matching [14] can address component discovery. However, signature matching considers only function types and ignores their behaviors and two functions with the same signature can have completely opposite behaviors (consider, for example,

two functions for addition and subtraction of two integers). Although specification matching aims to address this problem by comparing software components based on formal descriptions of the semantics of their behaviors, and employs formal specifications of the components' behavior in terms of invariants, and pre- and post-conditions of their methods, it requires the development of formal specifications to characterize the available components. As these specifications have to be developed independently from the component code, there is no guarantee that they correctly and completely reflect its behavior.

Semantic Web Services (SWS) empowers Web services with semantics. In order to enable the automatic discovery, combination and use of distributed components, two major initiatives aim to realize Semantic Web Services by providing appropriate description means that enable the effective exploitation of semantic annotations with respect to discovery, composition, execution and interoperability of Web Services, namely: OWL-S and WSMO.

OWL-S [4, 5], an effort by BBN Technologies, Carnegie Mellon University, Nokia, Stanford University, SRI International and Yale University, defines an ontology for the semantic markup of Web Services and is intended to enable automation of Web Service discovery, invocation, composition, interoperation and execution monitoring by providing appropriate semantic descriptions of services. WSMO [6], led by the the Semantic Web Services working group which includes more than 50 academic and industrial partners, seeks to create an ontology for describing various aspects related to Semantic Web Services, and aims to solve the integration problem. Both initiatives have the goal of providing a world-wide standard for the semantic description of Web Services, grounding its application in a real world setting. However, OWL-S is more mature in some aspects, such as the definition of the process model and the grounding of Web Services.

As part of the semantic web effort, the two languages are intended as the means for semantically specifying domain specific ontology and the functions and internal behaviors of the services that agents in these domains may provide. As a result, they enable discovery through

specification matching, such as the methods proposed in LARKS [15] and matchmaker in [7]. In a similar way, Gao [16] proposes a formal capability description language that provides primitives for a high level ontological description of what the service is about, and formal specifications of the functionalities it delivers in terms of pre- and post-conditions and constraints. Paolucci [7] and Sycara [8] from Carnegie Mellon University propose DAML-S (DARPA Agent Markup Language Schema) as a prototypical example of ontology for describing Semantic Web services to support capability matching and to manage interaction between Web services. Moreover, based on DAML-S Profiles, they describe the implementation of the DAML-S/UDDI Matchmaker that expands on UDDI by providing semantic capability matching, and present the DAML-S Virtual Machine that uses the DAML-S Process Model to manage the interaction with Web service. Vacharasintopchai [2] presents a semantic web service framework, which describes a semantic matching algorithm including the matching of the input, output and effect parameters between the advertisements and requests in detail, for joint application in the field of computational mechanics.

There also has been some work aimed at bridging the gap between requiring full-fledged semantic specifications and relying simply on currently available WSDL specifications for service matching. Ponnekanti and Fox [17] focus on identifying and bridging incompatibilities of services using a “testing” approach. The Woogole system [18], developed by Dong et al., clusters WSDL descriptions according to the names of the service’s operation parameters, and leverages these clusters to retrieve services that may deliver operations of interest to the requester. The basic idea of the Woogole method is the exploitation of the semantics of the identifiers chosen to describe WSDL elements. Buche [19] use multi-view fuzzy querying, in which fuzzy sets may be defined on a hierarchical domain of values called an ontology (values of the domain are connected using the *a kind of* semantic link), to query incomplete, imprecise and heterogeneously structured data stored in a relational database with fuzzy selection criteria.

The objective of the weighted ontology-based semantic similarity algorithm discussed in this

paper is to support the programmatic discovery of relevant services and to improve the veracity of web service discovery and matching.

3. Definition of Web Service Ontology

In a semantic web service framework, the semantic structure of web service ontology can be described as a 4-tuple-based ontology: $SWS(NP, FP, IP, O)$ as described in figure1.

In $SWS(NP, FP, IP, O)$, SWS is the name of the web service; NP represents the non-functional properties of SWS , including service ID, service name, service type, Quality of service (security, reliability, response time, call cost etc.), economic properties, contact and version. FP represents the function properties of SWS , consisting of syntax properties, static semantics (messages, operation semantics), service mediator and dynamic semantics (behavior, operation logic).

$IP = (P_{in} \cup P_{out} \cup P_{precondition} \cup P_{postcondition} \cup P_{effect} \cup \dots)$ represents a collection of interface properties set for a semantic web service, consisting of input interfaces sets P_{in} , output interfaces sets P_{out} , pre-condition interfaces sets $P_{precondition}$, post-condition interfaces sets $P_{postcondition}$ and effect interfaces sets

P_{effect} . P_{in} , P_{out} , $P_{precondition}$, $P_{postcondition}$ and P_{effect} are of SWS ; O is the ontology which supports the SWS , and the concepts and roles that the parameter set of SWS used are from ontology O . Thus,

$IOP \subseteq O$.

Figure 1 the structure of web service ontology

3.1 Domain Ontology of Web Services

Ontology is a formal explicit specification of concepts in a domain of discourse, the properties of each concept describing various features and attributes of the concept, and restrictions on slots [20, 21]. Concepts denote sets of any individuals, and roles denote binary relationships between individuals. The extension of any concept is a subset of any individuals set, while the extension of any role is a subset of ordered pairs of individuals. In this sense, we give the formal definition of ontology.

Definition 1 An ontology O consists of six elements $\{C, A^C, R, A^R, H, X\}$, where C represents a set of concepts; A^C represents a collection of attribute sets, one for each concept; R represents a set of relationships; A^R represents a collection of attribute sets, one for each relationship; H represents a concept hierarchy; and X represents a set of axioms.

Each concept c_i in C represents a set of objects of the same kind, and can be described by the same set of attributes denoted by $A^C(c_i)$. Each relationship $r_i(c_p, c_q)$ in R represents a binary association between concepts c_p and c_q , and the instances of such a relationship are pairs of (c_p, c_q) concept objects. The attributes of r_i can be denoted by $A^R(r_i)$. H is a concept hierarchy derived from C and it is a set of parent-child (or superclass-subclass) relations between concepts in C . $(c_p, c_q) \in H$ if c_q is a subclass, or sub-concept, of c_p . Each axiom in X is a constraint on the concept's and relationship's attribute values or a constraint on the relationships between concept objects. Each constraint can be expressed like a Prolog-like rule [22]. Figure 2 depicts a simple University ontology O_{univ} [23] in the concept hierarchy.

Figure 2 Part of university ontology

$$O_{univ} = \{C_{univ}, A_{univ}^C, R_{univ}, A_{univ}^R, H_{univ}, X_{univ}\}$$

Where,

$$C_{univ} = \{Student, PhDStudent, AcademicStaff, \\ Professor, Department, Course, Project, \dots\}$$

$$A_{univ}^C = \{A_{univ}^C(Student), A_{univ}^C(PhDStudent), \\ A_{univ}^C(AcademicStaff), A_{univ}^C(Professor), \\ A_{univ}^C(Department), A_{univ}^C(Course), \\ A_{univ}^C(Project), \dots\}$$

$$A_{univ}^C(Student) = \{name, matricnum, email, \dots\}$$

$$A_{univ}^C(PhDStudent) = \{name, matricnum, project, \dots\}$$

$$A_{univ}^C(AcademicStaff) = \{name, staffid, \dots\}$$

$$A_{univ}^C(Professor) = \{name, staffid, email, \dots\}$$

$$A_{univ}^C(Department) = \{name, researcharea, \dots\}$$

$$A_{univ}^C(Course) = \{name, title, period, \dots\}$$

$$A_{univ}^C(Project) = \{name, title, period, \dots\}$$

.....

$$R_{univ} = \{Supervise(Professor, PhDStudent), \\ Teach(Professor, Course), \\ TaughtBy(Course, Professor), \\ Faculty(AcademicStaff, Department), \\ TakeCourse(Student, Course), \dots\}$$

$$A_{univ}^R = \{A_{univ}^R(Supervise), A_{univ}^R(WorkIn), \\ A_{univ}^R(Major), A_{univ}^R(Teach), A_{univ}^R(Take), \dots\}$$

$$A_{univ}^R(Supervise) = \{startdate, enddate, \dots\}$$

$$A_{univ}^R(WorkIn) = \{apptdate, \dots\}$$

$$A_{univ}^R(Major) = \{academicyear, \dots\}$$

$$A_{univ}^R(Teach) = \{semester, year, \dots\}$$

$$A_{univ}^R(Take) = \{semester, year, \dots\}$$

.....

$$H_{univ} = \{(Student, PhDStudent), \\ (AcademicStaff, Professor), \dots\}$$

$$X_{univ} = \{TaughtBy(X, Y) \leftarrow Teach(Y, X), \dots\}$$

This ontology can be represented in any ontology representation language such as OWL-S [25], WSMO [26] and others.

3.2 Graph-based Interface Properties of Web Service Ontology

Interfaces properties sets for a semantic web service consist of P_{in} , P_{out} , $P_{precondition}$, $P_{postcondition}$ and P_{effect} , and can be described as graph-based interface ontology to match the request and advertisement service by concept ontology used in IP. Figure 3 describes the graph-based interface properties of a web service ontology- searchBook service: the output interface is bookName, the output interface is bookDescription and the pre-condition interface is studentID. The arc represents a 3-tuple (s, p, o), where, s is the start node, o is the destination node, and p is the arc.

Figure 3 Graph-based interface properties

4. Weighted Ontology-based Computation of Semantic Similarity for Web Service

According to Interface Properties of Web Service Ontology, as shown in figure 3, the matching process of request and advertisement service can be decomposed into the computing process of weight-based Interface properties matching. Therefore, semantic web service similarity can be addressed through an information theory-based concept similarity algorithm.

The interface properties for a semantic web service, consisting of input interfaces sets, output interfaces sets, pre-condition interfaces sets, post-condition interfaces sets and effect interfaces sets, can be represented as a probabilistic 5-dimensional vector-space model: $IP=(P_{in}, P_{out}, P_{precondition}, P_{postcondition}, P_{effect})$. For example, an OWL-S profile description is a set of OWL-S statements that semantically describes a service, which is either needed by a service consumer or offered from a service provider. From the section on “Service Profiles” in the OWL-S specification, the elements of a profile description that are relevant to the interoperation of Web Services are the taxonomic type of profile, i.e., whether a service belongs to a certain class and “has input,” “has effect,” and “has output” properties.

Web service matching queries are also represented as vectors. Similarity distance between a provider service vector p and a query service vector q , can then be computed as the vector inner product:

$$Sim(p, q) = d \bullet q = \sum_{i=1}^l W_{id} * W_{iq} \quad (1)$$

In the above equation, W_{id} and W_{iq} are the semantic similarity of *interface parameter i*, which can be represented as a concept or a term, i.e., the similarity of web service can be addressed through calculating the vector inner product of the concept vector.

A higher similarity score indicates a closer similarity between the query and the retrieved web services.

4.1. Minimum length of edge-counting path based concept semantic similarity

Semantic similarity measures can be used to calculate the similarity of two concepts organized in an ontology. Concepts are organized into synonym sets (synsets) in the knowledge base, with semantics and relation pointers to other synsets. Therefore, the first class can be found in the hierarchical semantic net that subsumes the compared words. Moreover, one direct method for Similarity calculation is edge-counting [27], which finds the minimum length of the path

connecting the two concepts. For example, the shortest path between *boy* and *girl* in Figure 4 is *boy-male-person-female-girl*, the minimum path length is 4, the synset of person is called the subsumer for words of *boy* and *girl*, while the minimum path length between *boy* and *teacher* is 6. Thus, we could say *girl* is more similar to *boy* than *teacher* to *boy*. Rada et al. [27] demonstrated that this method works well on their much-constrained medical semantic nets (with 15,000 medical terms).

However, this method may be less accurate if it is applied to larger and more general semantic nets such as WordNet [28]. For example, the minimum length from *boy* to *animal* is 4, less than from *boy* to *teacher*, but, intuitively, *boy* is more similar to *teacher* than to *animal* (unless you are cursing the boy). To address this weakness, the direct path length method must be modified by utilizing more information from the hierarchical semantic nets.

Figure 4 Fragment of domain ontology

4.2. Information theory based concept semantic similarity

Following the standard argumentation of information theory [29], the ontology structure defines the function $Parents(c)$ that, given a concept c , returns the set of more generic concepts directly linked to c . In the case of an ontology organized as a tree, $Parents(c)$ always returns a

single concept. On the other hand of multiple inheritance, two distinct ancestor nodes may both be minimal upper bounds, those two nodes might have very different values for information content, so $Parents(c)$ can return more than one term (concept), such as Directed Acyclic Graphs (DAG).

Using the function $Parents(c)$ the set of paths between two concepts c_a and c_b can be defined as:

$$Paths(c_a, c_b) = \{(c_1, \dots, c_n) | (c_a = c_1) \wedge (c_b = c_n) \wedge (\forall i: (1 \leq i < n) \wedge ((c_i \in Parents(c_{i+1}))))\}$$

A concept a is an ancestor of a concept c when there is at least one path from a to c :

$$Ancestors(c) = \{a | Path(a, c) \neq \emptyset\}$$

Note, since $c \in Path(c, c)$, we have $c \in Ancestors(c)$.

The information content of a concept is inversely proportional to its frequency in a corpus, such as the Brown Corpus of American English [30], a large collection (1,000,000 words) of text across genres ranging from news articles to science fiction. The frequency of a concept c , $Freq(c)$, can be defined as the number of times that c and all its descendants occur:

$$Freq(c) = \sum \{occur(c_i) | c \in Ancestors(c_i)\}$$

Note, for each ancestor a of a concept c , we have $Freq(a) \geq Freq(c)$, because the set of descendants of a contains all the descendants of c . An estimate for the likelihood concept probabilities of observing an instance of a concept c is

$$Prob(c) = \frac{Freq(c)}{N}$$

where, N is the total number of all concepts in the corpus.

The information content of concept c can be defined as the negative base 2 logarithm of its probability:

$$IC(c) = -\log(Prob(c))$$

Based on the similarity probability $IC(c)$ of information content, the semantic similarity distance and similarity algorithms are given in the following content.

(1) Semantic similarity distance: $share(c_1, c_2)$ and $wsim(w_1, w_2)$

Note, the information content is monotonic, since it is non-increasing as we descend in the hierarchy.

Semantic similarity measures assume that the similarity between two concepts is related to the extent to which they share information. Given two concepts c_1 and c_2 , their shared information, $Share(c_1, c_2)$, can be defined as the information content of their most informative common ancestor:

$$Share(c_1, c_2) = \max\{IC(a) \mid a \in sub(c_1, c_2)\} \quad (2)$$

where, $sub(c_1, c_2)$ is the concepts that subsume both c_1 and c_2 .

In practice, one often needs to measure word similarity, rather than concept similarity. Using $s(w)$ to represent the set of concepts in the taxonomy that are senses of word w , define

$$wsim(w_1, w_2) = \max_{c_1, c_2} [share(c_1, c_2)] \quad (3)$$

where c_1 ranges over $s(w_1)$ and c_2 ranges over $s(w_2)$. This is consistent with Rada et al.'s (1989) treatment of “disjunctive concepts” using edge-counting; they define the distance between two disjunctive sets of concepts as the minimum path length from any element of the second.

Here, the word similarity is judged by taking the maximal information content over all concepts of which both words could be an instance. To take an example, consider how the word similarity $wsim(doctor, nurse)$ would be computed, using the taxonomic information in Figure 5. (Note that only noun senses are considered here.)

Figure 5 Another fragment of the WordNet taxonomy

By equation $wsim(w_1, w_2)$, we must consider all pairs of concepts c_1 and c_2 , where $c_1 \in \{Doctor1, Doctor2\}$ and $c_2 \in \{Nurse1, Nurse2\}$, and for each such pair we must compute the semantic similarity $share(c_1, c_2)$ according to Equation 1. Table 1 illustrates the computation.

Table 1 Computation of similarity for *Doctor* and *Nurse*

c_1 (Description)	c_2 (Description)	Subsumer	$share(c_1, c_2)$
<i>Doctor1</i> (medical)	<i>Nurse1</i> (medical)	<i>Health_Professional</i>	8.844
<i>Doctor1</i> (medical)	<i>Nurse2</i> (nanny)	<i>Person</i>	2.005
<i>Doctor2</i> (Ph.D.)	<i>Nurse1</i> (medical)	<i>Person</i>	2.005
<i>Doctor2</i> (Ph.D.)	<i>Nurse2</i> (nanny)	<i>Person</i>	2.005

As the table shows, when all the senses for *doctor* are considered against all the senses for *nurse*, the maximum value is 8.844, via *Health_Professional* as a most informative *subsumer*; this is, therefore, the value of word similarity for *doctor* and *nurse*.

(2) $share(c_1, c_2)$ & $wsim(w_1, w_2)$ based semantic similarity algorithms

Based on $share(c_1, c_2)$ and $wsim(w_1, w_2)$, Wu and Palmer, Resnik, Jiang and Conrath, Lin, Li and Bandar proposed some semantic similarity measure algorithm between two concepts.

Wu and Palmer [31] defined their similarity as the information content of their most informative common ancestor over their information content:

$$Sim_{wp}(c_1, c_2) = \frac{2 * N_3}{(N_1 + N_2 + 2 * N_3)} \quad (4)$$

where, N_1 and N_2 , are the number of *IS-A* links from c_1 and c_2 to their most specific common superclass C ; N_3 is the number of *IS-A* links from C to the root of the taxonomy.

Resnik [32] defined their semantic similarity as the information content of their most informative common ancestor:

$$Sim_{Resnik}(c_1, c_2) = Share(c_1, c_2) \quad (5)$$

Jiang and Conrath [33] defined their semantic distance as the difference between their information content and the information content of their most informative common ancestor:

$$dist_{JC}(c_1, c_2) = IC(c_1) + IC(c_2) - 2 * Share(c_1, c_2)$$

Note, Jiang and Conrath's formula measures a distance, the inverse of similarity. A similarity measure, based on Jiang and Conrath distance measure, can be defined as:

$$Sim_{JC}(c_1, c_2) = \frac{1}{dist_{JC}(c_1, c_2) + 1} \quad (6)$$

where, $dist_{JC} + 1$ is used to avoid infinity values, since $dist_{JC}(c, c) = 0$.

Lin [34] defined their similarity as the information content of their most informative common ancestor over their information content:

$$Sim_{Lin}(c_1, c_2) = \frac{2 * Share(c_1, c_2)}{IC(c_1) + IC(c_2)} \quad (7)$$

However, in the absence of a reliable algorithm for choosing the appropriate word senses, the most straightforward way to do so in an information-based setting is to consider all concepts to which both nouns belong, rather than taking just the single maximally informative class. Resnik [35] suggested defining a measure of weighted word similarity as follows:

$$wsim_{\alpha}(w_1, w_2) = \sum_i \alpha(c_i) [-\log p(c_i)] \quad (8)$$

where $\{c_i\}$ is the set of concepts dominating both w_1 and w_2 in any sense of either word, and α is a weighting function over concepts such that $\sum_i \alpha(c_i) = 1$. This measure of similarity takes more information into account than the previous one; rather than relying on the single concept with maximum information content, it allows each class representing shared properties to contribute information content according to the value of $\alpha(c_i)$. Intuitively, these c_i values measure relevance.

Li and Bandar [36] proposed that the similarity $Sim_{LB}(c_1, c_2)$ between concepts c_1 and c_2 was a function of the attributes path length, depth of concepts and density of the semantic nets based on information content as follows:

$$Sim_{LB}(c_1, c_2) = f(l, h, d)$$

where, l is the shortest path length between c_1 and c_2 , h is the depth of subsumer in the hierarchical semantic net, d is the local semantic density of c_1 and c_2 .

Assumed that $Sim_{LB}(c_1, c_2) = f(l, h, d)$ can be rewritten using two independent functions as:

$$Sim_{LB}(c_1, c_2) = f(f_1(l) \cdot f_2(h) \cdot f_3(d)) \quad (9)$$

where, f_1 , f_2 , and f_3 are transfer functions of path length, depth, and local density, respectively.

Taking the weight of the path length according to equation (8), a monotonically decreasing function of l is set as:

$$f_1(l) = e^{-\alpha l} \quad (10)$$

where α is a constant, l is the sum of the weight of all links from c_1 to c_2 under the weighted Equation (8). The selection of the function in exponential form ensures that f_1 satisfies the constraints within the range from 0 to 1.

The depth of the subsumer is derived by counting the levels from the subsumer to the top of the lexical hierarchy. For polysemous words, the subsumer on the shortest path is considered when deriving the depth of the subsumer. Concepts at upper layers of the hierarchical semantic net have more general concepts and less semantic similarity between concepts than concepts at lower layers. This behavior must be taken into account in calculating $Sim_{LB}(c_1, c_2)$. Therefore, we need to scale down $Sim_{LB}(c_1, c_2)$ for subsuming words at upper layers and to scale up $Sim_{LB}(c_1, c_2)$ for subsuming words at lower layers. Moreover, as stated earlier the similarity interval is finite, say $[0, 1]$. As a result, $f_2(h)$ should be a monotonically increasing function with respect to depth h . To satisfy these constraints, we set $f_2(h)$ as:

$$f_2(h) = \frac{(e^{\beta h} - e^{-\beta h})}{(e^{\beta h} + e^{-\beta h})} \quad (11)$$

where, $\beta > 0$ is a smoothing factor. As $\beta \rightarrow \infty$, the depth of a word in the semantic nets is not considered. This function form can be considered as an extension of Shepard's law [21], which claims that exponential-decay functions are a universal law of stimulus generalization for psychological science. The extension is that (4) employs an exponential-growth function of similarity with depth rather than an exponential-decay function because similarity increases with depth. h is the sum of the weight of *IS-A* links from C to the root of the taxonomy under the weighted Equation (8).

Finally, as in setting the function form for transferring depth in (7), local semantic density is defined as a monotonically increasing function of information content- $wsim(w_1, w_2)$,

$$f_3(d) = \frac{(e^{\lambda wsim(w_1, w_2)} - e^{-\lambda wsim(w_1, w_2)})}{(e^{\lambda wsim(w_1, w_2)} + e^{-\lambda wsim(w_1, w_2)})} \quad (12)$$

where, $\lambda > 0$ is a smoothing factor. As $\lambda \rightarrow \infty$, then the information content of words in the semantic nets is not considered.

5. Implement of Semantic Web Service Similarity

Semantic Web Services possess the potential to help unify the computing resources and knowledge scattered on the Internet into a large platform for collaboration. To facilitate the publication and discovery of semantic Web services, a Semantic Web Service Execution Environment (SWSEE), as an extended version of the standard Web Services model, is proposed for the development of collaborative systems as depicted in figure 6.

SWSEE is composed of a web service deploying module and a web service calling module. The service provider, after making the semantic description of a semantic web service (OWL-S or WSMO), deploys the service on his web service server and registers the service into the service registry, using the web service deploying module. Three tools are used in deploying and registering the service. When a company decides to outsource a task through web services,

SWSEE searches, matches and calls the available web service using the web service calling module.

Figure 6 Semantic Web Service Execution Environment

The service requester matches, ranks, selects, and executes web services by using the web service calling module. It consists of the following components: the Semantic matching tool, which matches, ranks, and selects web services by sending a query to the service registry; the Semantic reader, which helps the service user with browsing the service semantics; the WSDL reader, which enables the service caller to understand documents written with WSDL and stored in service registry; the service caller, which calls one of the operations of a chosen web service; the OWL-to-service implementation tool, which encodes the semantic description into the service implementation; the Web service deployment tool, which compiles the service implementation and deploys it onto the web service server; the Web service register tool, which reads WSDL documents and semantic model descriptions from the service provider's workspace, and enrolls the web service to the registry.

5.1 Description Model of Semantic Web Service

SWSEE includes a model for registering and discovering the semantics of services, which should be written with OWL/OWL-S. The service registry model provides functionalities that allow service providers to register service capabilities and users to locate the available services. The service users can reference the semantic information from the service registry and compare the pre- and post-conditions and the input and output specifications with the semantics presented (Figure 7).

Figure 7 OWL based semantic web service

The service information in the service registry includes the following: name, type of service, input and output specifications and the pre- and post-conditions; table 2 gives a simple sample of semantic web service. The service requester must request the service provider guaranteed for certain qualities before calling a service specialized on the component (pre-conditions), and the service provider guarantees certain properties after the service returns outputs to the service requester (post-conditions).

Table 2 an example of semantic web service

```

<!--definition of LogicalService
<!--definition of "ControllerDsg"
<service:LogicalService rdf:id = "ControllerDsg">

```

```

<service:hasInput>
  <service:Spec rdf:resource = "#EngineSpec"/>
  <service:Spec rdf:resource = "#CoolantType"/>
</service:hasInput>
<service:hasOutput>
  <service:Spec rdf:resource = "#WiringSpec"/>
  <service:Spec rdf:resource = "#ControllerSpec"/>
</service:hasOutput>
<service:hasAlternativeChoices>
  <service:AlternativeChoice rdf:parseType = "collection">
    <service:AlternativeChoice/>
  </service:AlternativeChoices>
</service:LogicalService>

```

The service provider posts pre-conditions to inform the service requester of the requirements that the service provider must satisfy to invoke the service. For example, the service provider can restrict the qualifications of the service requester, and specify the quality of inputs to deliver service invocation results. On the other hand, the service provider describes the quality matrix of service execution results in the post-condition entry. The service provider can publish its quality of service level, such as product specs and processing time.

Since the service registry does not provide full-scale semantics, the OWL bounding provides the user with detailed semantics of a service. The full description of a service is given by a separate document, and the link to this document is provided to service searchers as well. The technical information of the service, such as the URL address, the operation list, and type of required parameters, is provided by a separate WSDL document and URL links. Service users can see the technical information and full service description by following the links provided in the individual service information.

5.2 Matching & Ranking algorithm

An OWL-S profile description is a set of OWL-S statements that semantically describes a service, which is either needed by a service consumer or offered from a service provider. From the section on “Service Profiles” in the OWL-S specification, the elements of a profile description that

are relevant to the interoperation of Web Services are the taxonomic type of profile, i.e., whether a service belongs to a certain class and “has input,” “has effect,” and “has output” properties. Examples of such elements are the ‘profileHierarchy:SineFunction-Evaluator’, the “profile:hasInput”, the “profile:hasEffect”, and the “profile:hasOutput”, respectively.

For each pair of service profiles, the degree of match is calculated by using the weighted average of the matching scores between the pairs of the profile types, the input parameters, the effects of service, and the output parameters. Mathematically, the degree of match between a pair of service profiles is

$$D_s = \frac{\sum_i W_i * d_i^p}{\sum_i W_i}$$

where, D_s is the degree of match between two service profiles; W_i and d_i^p is, respectively, weight and the matching scores between the profile types, the input parameters, the effects of services, and the output parameters. By default, equal weights are assigned to the matching scores in typical matchmaking operations. Service consumers may request higher weights to certain pairs of the profile descriptions if compatibility between those pairs is more important.

For each pair of the request concept C_R and advertised concept C_A , the matching score of similarity relation between C_R and C_A with respect to C_R , $SR(C_R, C_A)$ can be computed by **Equation (9)**.

Web services often take several input parameters. In this case the matching score between the two sets of input parameters is the weighted sum of the matching scores between each sequential pair of the input parameters. Mathematically, according to **Equation (1)**, it is defined as

$$d_i^p = \sum_j w_j d_j^i$$

where, d_i^p is the matching score between two sets of input parameters, d_j^i is the degree of match between the j th pair of input parameters, w_j is the weight of the j th pair of input parameters,

and there exists $\sum_j w_j = 1$. To eliminate the case of strong incompatibilities between certain pairs of input parameters, d_i^p must be taken as zero if any of d_j^i equals zero.

Following the above computing process, the degree of match between the query service and the advertisement services from the registry center is calculated. Thus, the semantic similarity between a matching pair of web service can be ranked to manually or programmatically select the objective web service. After that, the selected web service can be bound into the business process.

5.3 Case Study

In our scenario, we consider a case study [43] that describes a die casting process for thermoelectric fan housing. The die casting process for thermoelectric fan housing can be decomposed into several sub-tasks and each task can be assigned to the initial collaborating companies. Four companies are initially collaborating in a process, and each company has its own dedicated jobs, such as defining requirements, material selection, die making, and casting products. Tasks in the process are interdependent: one task in one company affects the other tasks in other companies.

Company C's job is making its product using a casting process with output specifications provided by company A and B. Company C can make a *buy* or *make* decision. We assume C decides to outsource the *casting products* task. C will exploit SWSEE to locate and assign the *casting products* task through web services. Company C tries to outsource the *casting product* and to locate the right service provider. In order to outsource a *casting products* job, the die casting manufacturing capabilities should be published and advertised in an understandable way so that every OEM can find the right provider to satisfy the OEM's customized needs. The manufacturer's capabilities are registered as an entry to the semantic service registry provided in the SWSEE framework. Figure 8 gives the startup user interface to query the advertisement service in the service registry.

Figure 8 Query startup interface

When company C decides to outsource the *casting product* task, the company defines the desired attributes for the product and searches the service registry using such attributes. The attributes include *Materials*, *Finishing*, and *Certifications* as described in the domain ontology database. Moreover, the semantic service defines the pre-condition and post-condition which will be used for locating and evaluating service providers as illustrated in table 3.

Table 3 Semantic Service Definition in registry

```

<entry>
  <entryID> 00001</entryID>
  <serviceType> CastingProducts</serviceType>
  <profile>
    <serviceName> Die Casting</serviceName>
    <Address> Liuix, Zhejiang, China</Address>
    <Telephone> 0571-xxxx-xxxx</Telephone>
    :
  </profile>
  <precondition>
    <Materials> Aluminum</Materials>
    <Machining> CNC</Machining>
    <Machining> Tapping</Machining>
    <Finishing> Powder Coating</Finishing>
    :
  </precondition>
  <postcondition>
    <Certifications> ISO9002</Certifications>
    <EndMarkets> Automotive</EndMarkets>
    <EndMarkets> Furniture</EndMarkets>
    :

```

```

</postcondition>
<inputSpec> TrimDies</inputSpec>
<inputSpec> Dies</inputSpec>
<inputSpec> SelectedMaterial</inputSpec>
<outputSpec> FinishedProduct</outputSpec>
<wsdl_binding> http://www.tongji.edu.cn/cims/wsdl/DieDesign.wsdl</wsdl_binding>
<owl_binding> http://www.tongji.edu.cn/cims/wsdl/services.daml </owl_binding>
</entry>

```

In the above enterprise collaboration case, company C will create a query to the service registry with “*Materials = Aluminum and Certifications = ISO9002 and Finishing = power coating*”. In SWSEE, such queries can either be generated autonomously, based on the pre-defined evaluation functions, or typed manually. Figure 9 gives the user interface to input the Input parameters of the semantic web services.

Figure 9 Interface Parameters

According to the Input parameters of the query web service, SWSEE applies the service matching tool to discover and rank the advertisement services in the service registry center, and gives the list according to the semantic similarity degree between the query service and the advertisement, as shown in figure 10. Therefore, company C can select the service through clicking the service list. The selected service will then be bound into the business process to finish the collaborative outsourcing task. The matching tool is implemented by an agent web service.

Therefore, the manufacturing capability of a casting company in Liuxi, Zhejiang Province, China, will satisfy the query from company C, and then the service registry notifies them that the Liuxi-located casting company can fulfill the *casting product* job. Then company C evaluates other attributes, which is not prepared in the query, starts negotiation if necessary, and outsources ‘casting products’ by invoking services prepared by the casting company in Troy. The service enactment model supports various service invocation types and, hence, facilitates the dynamic process enactment required for the semantic web service in distributed and heterogeneous manufacturing environments.

Figure 10 Ranking list of the matching web service

In the above example, the inference engines that can be used in SWSEE are RACER [38] and the SWI-Prolog Semantic Web package [39], which are respectively based on Description Logics [40] and Prolog language. Protege [41] is used as the ontology modeling tool. RACER is an example of an inference engine that natively understands the semantics of RDF and OWL constructs because its underlying Description Logics framework was used as the basis to design the OWL language. SWI-Prolog, with its Semantic Web package, is an example of an inference engine equipped with a set of rules that enable it to understand the semantics of the RDF and OWL constructs. It should be noted that RDF and OWL are knowledge representation languages. The RDF and OWL specifications do not mandate the use of Description Logics as the only logical framework for the Semantic Web. The knowledge inferred by an inference engine, or the answer that it provides, is limited by its underlying logical framework. Protege, developed by the

Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine [42], is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontology. The matching algorithm is developed according to the equation (1) presented in section 4.

The system includes some other development tools and environments, such as IBM Application Developer Integration Edition, Eclipse, myEclipse, java JDK, Tomcat, MySQL, VOISP (Visitor oriented information services platform, <http://cims.tongji.edu.cn/voisp>) and VOISCP (Visitor oriented information services composite platform).

6. Conclusion

In this paper, a weighted ontology-based semantic similarity algorithm for Web Service is proposed, which can be used to support a more automated and veracity service-discovery process, by distinguishing among potentially useful and likely irrelevant services and by ordering the potentially useful ones according to their relevance to the developer's query.

In term of future research, we are working towards two contents: (1) comparing a weighted ontology-based semantic similarity algorithm in terms of veracity with the traditional matching algorithms, such as a specification matching based probabilistic approach and category-based directory; (2) analyzing the weighted parameters in equation 6 to improve the veracity of the matching algorithm, (3) applying a weighted ontology-based semantic similarity algorithm to cross-enterprise business process collaboration.

Acknowledgements

The research work presented in this paper was partially supported by the National High Technology Research and Development Program of China (Grant No. 2007AA04Z104 and 2007AA10Z206), the China Scholarship Council (through the Visiting Scholarship to the first author), and the National Research Council Canada (to host the first author as a visiting

researcher).

Reference

Formatted: Space Before: 0.5 line

- [1] T. Bellwood, L. Clement, D. Ehnebuske, et al. UDDI Technical Paper, <http://www.uddi.org/pubs/UDDI-V3.00-Published-20020719.doc>, July 19, 2002
- [2] T. Vacharasintopchai, W. Barry, V. Wuwongse, W. Kanok-Nukulchai, "Semantic Web Services Framework for Computational Mechanics", *Journal of Computing in Civil Engineering*, 2007, 21(2), 65-77.
- [3] M. Burstein, C. Bussler, T. Finin, M.N. Huhns, M. Paolucci, A.P. Sheth, S. Williams, M. Zaremba, "A semantic Web services architecture", *IEEE Internet Computing*, 2005, 9(5), 72-81.
- [4] The OWL Services Coalition, The OWL Services Coalition, OWL-S 1.1 beta release. <http://www.daml.org/services/owl-s/1.1B/>, July 2004.
- [5] D. Martin, M. Burstein, J. Hobbs, et al., OWL-S: Semantic markup for Web services, <http://www.w3.org/Submission/OWL-S>, Nov. 22, 2004.
- [6] D. Romana, U. Kellera, H. Lausena, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, D. Fensel, "Web Service Modeling Ontology", *Applied Ontology*, 2005, 1(1), 77-106.
- [7] M. Paolucci, T. Kawamura, T.R. Payne, K. Sycara, "Semantic matching of web services capabilities", *The 6th IEEE International Symposium on Wearable Computers*, Seattle, Washington, USA, Oct. 7-10, 2002.
- [8] K. Sycara, M. Paolucci, A. Ankolekar, N. Srinivasan. "Automated discovery, interaction, and composition of semantic Web services", *Journal of Web semantics*, 2003, 1(1), 27-46
- [9] T. Vacharasintopchai, W. Barry, V. Wuwongse, W. Kanok-Nukulchai, "Semantic Web Services Framework for Computational Mechanics", *Journal of Computing in Civil Engineering*, 2007, 21(2), 65-77.
- [10] E. Stroulia, Y.Q. Wang, "Structural and Semantic Matching for Assessing Web-service Similarity", *International Journal of Cooperative Information Systems*, 2005, 14(4), 407-437
- [11] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval", *Information*

- Processing and Management: An Int. J.* 1988, 24(5): 513–523.
- [12] G. Salton, A. Wong and C. S. Yang, “A vector-space model for information retrieval”, *J. American Society for Information Science*, 1975, (18), pp. 13–620.
 - [13] A. M. Zaremski, J. M. Wing, “Signature matching: A tool for using software libraries”, *ACM Transactions on Software Engineering and Methodology*, 1995, 4(2): 146–170.
 - [14] A. M. Zaremski, J. M. Wing, “Specifications matching of software components”, *ACM Transactions on Software Engineering and Methodology*, 1997, 6(4): 333–369.
 - [15] K. Sycara, S. Widoff, M. Klusch, J. Lu, “LARKS: dynamic matchmaking among heterogeneous software agents in cyberspace”, *Autonomous Agents and Multi-Agent Systems*, 2002, (5): 173–203.
 - [16] X. Gao, J. Yang, M. P. Papazoglou, “The capability matching of web services”, *Fourth IEEE Int. Symp. Multimedia Software Engineering*, Newport Beach, CA, USA (IEEE Press, 11–13 December 2002), pp. 56–63.
 - [17] S. Ponnekanti, A. Fox, “Interoperability among independently evolving web services”, in *Proc. Middleware 2004*, pp. 331–351.
 - [18] X. Dong, A. Y. Halevy, J. Madhavan, E. Nemes, J. Zhang, “Similarity search for web services”, in *Proc. VLDB 2004*, pp. 372–383.
 - [19] P. Buche, C. Dervin, O. Haemmerle, R. Thomopoulos, “Fuzzy Querying of Incomplete, Imprecise, and Heterogeneously Structured Data in the Relational Model Using Ontologies and Rules”, *IEEE Transactions on Fuzzy Systems*, 2005, 13(3), 373–383.
 - [20] S. Staab, A. Maedche, and S. Handschuh. “An Annotation Framework for the Semantic Web”. *Proc. 1st workshop on Multimedia Annotation*, Tokyo, Japan, 2001.
 - [21] S. Staab, A. Mdche, and S. Handschuh. “Creating Metadata for the Semantic Web”. *Computer Networks*, 2003, 42(5):579-598
 - [22] I. Bratko. *PROLOG Programming for Artificial Intelligence*. Pearson Education Limited, third edition, August 2000.
 - [23] M. M. Naing, E. Lim, D. G. Hoe-Lian. "Ontology-based Web Annotation Framework for HyperLink Structures", *Proc. 3rd International Conference on Web Information Systems Engineering Workshops. (WISE 2002)*. Singapore, Dec. 11, 2002, pp.184-193

- [24] D. Brickley and R. Guha. RDF Vocabulary Description Language 1.0: RDF Schema, April 2002.
http://www.w3.org/TR/rdf-schema/#ch_appendix_rdfs.
- [25] The OWL Services Coalition, The OWL Services Coalition, OWL-S 1.1 beta release.
<http://www.daml.org/services/owl-s/1.1B/>, July 2004.
- [26] D. Romana, U. Kellera, H. Lausena, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, D. Fensel, "Web Service Modeling Ontology", *Applied Ontology*, 2005, 1(1), 77–106.
- [27] R. Rada, H. Mili, E. Bichnell, and M. Blettner, "Development and Application of a Metric on Semantic Nets," *IEEE Trans. System, Man, and Cybernetics*, vol. 9, no. 1, pp. 17-30, 1989.
- [28] G.A. Miller, "WordNet: A Lexical Database for English," *Comm. ACM*, vol. 38, no. 11, pp. 39-41, 1995.
- [29] S. Ross. A First Course in Probability. Macmillan, 1976.
- [30] W. N. Francis, H. Kucera. *Frequency Analysis of English Usage: Lexicon and Grammar*. Houghton Mifflin, Boston, 1982.
- [31] Z. Wu, M. Palmer. "Verb semantics and lexical selection". *In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics*, Las Cruces, New Mexico, 1994, pages 133–138
- [32] P. Resnik. "Using information content to evaluate semantic similarity in a taxonomy". *In Proceedings of IJCAI-95*, pages 448–453, Montreal, Canada, 1995.
- [33] J. Jiang, D. Conrath. "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy". *In tenth International Conference on Research on Computational Linguistics (ROCLING X)*, 1997, Taiwan.
- [34] D. Lin. "An information-theoretic definition of similarity". *In Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98)* Madison, Wisconsin, 1998.
- [35] P. Resnik. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, 1999, (11):95-130
- [36] Y.H. Li, Z. A. Bandar, D. McLean. "An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources". *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 2003, 15(4): 871-882
- [37] Y. Liu and C.Q. Zong, "Example-Based Chinese-English MT," *Proc. 2004 IEEE Int'l Conf. Systems, Man, and Cybernetics*, vols. 1-7, pp. 6093-6096, 2004.

- [38] V. Haarslev, R. Moller, "RACER: A core inference engine for the semantic Web." *Proc., 2nd Int. Workshop on Evaluation of Ontology-Based Tools (EON2003)*, Located at the 2nd Int. Semantic Web Conf. (ISWC'03), Sanibel Island, Fla., 27–36, <http://www.sts.tuharburg.de/~r.f.moeller/racer/papers/2003/HaMo03e.pdf>.
- [39] J. Wielemaker, G. Schreiber, B. Wielinga, "Prolog-based infrastructure for RDF: Performance and scalability." *Proc. 2nd Int. Semantic Web Conf. (ISWC'03)*, Sanibel Island, Fla., 644–658, <http://hcs.science.uva.nl/projects/SWI-Prolog/articles/iswc-03.pdf>, April 25, 2006.
- [40] D. Nardi, *The description logic handbook: Theory, implementation, and applications*, Cambridge University Press, Cambridge, U.K.
- [41] M. Horridge, H. Knublauch, A. Rector, R. Stevens, C. Wroe. "A Practical Guide to Building OWL Ontologies Using the Protege-OWL Plugin and CO-ODE Tools Edition 1.0". Available from: <http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf>.
- [42] Protege. <http://protege.stanford.edu/>, 2007
- [43] Steel Founder's Society of America, <http://www.sfsa.org>, 2004.