A Pseudo Ideal Inter-Concept Semantic Similarity Metric for Semantic Web Services Matching and Composition

(2014 January) By Okhtay Sattari, co-founder of TPS (www.ebtps.com), Tehran, Iran.

Abstract

Similarity/dissimilarity measurement plays a crucial role in information/component/service retrieval and integration. In this paper, we define a pseudo ideal semantic similarity metric which is suitable to be used for semantic web services matching and composition. Because it fulfills the requirements for similarity measurement in the field of web service retrieval that have been investigated and explained in this paper. Our proposed ideal semantic similarity metric considers the direction in comparing two concepts which may be from different ontologies and it measures the similarity/dissimilarity between two concepts by the extent to which the second concept includes instances which are also included by the first concept. Since this ideal similarity metric is not generally actually computable, we call it "pseudo ideal similarity metric", however it can be estimated based on Description Logic (DL) based descriptions of concepts in ontologies. Then, we propose a set of canonization rules for transforming the DL based descriptions of concepts to canonical form that can be used as a part of actually applicable logic based solution to inter-concept similarity measurement which estimates the ideal metric presented in this paper. We also conceptually investigate the behavior of logic-based semantic similarity measures in the field of web service retrieval and finally review the related works in comparison with our proposed ideal similarity metric.

Keywords: Inter-Concept Similarity and Dissimilarity, Overlapping concepts, Description Logics, Ontology, Ideal Similarity Metric, Semantic Matching, Web Service Composition.

1. Introduction

Similarity measures play a crucial role in information/component/service retrieval and integration [18]. While similarity measurement is not restricted to solve a particular problem, most similarity measures have been developed for a specific purpose. Therefore, the question of which measures should be selected depends on the application area [12]. In the presented research, we have focused on a sort of semantic similarity/dissimilarity measures that is suitable to be used for semantic matching of web services in order to automatically compose new web services by discovering and integrating the existing ones.

Web service composition is a crucial operation in creating service oriented applications. Web Services are heterogeneous software components which are encapsulated as standard software services with standard descriptions and interfaces. Hence, planning for integrating the existing web services is made possible by only checking and examining their standard descriptions and interfaces. Finally, the composed web services can be executed by exchanging standard xml-based messages among their constituent web services in an asynchronous way [5, 13, 18, 28, 38]. It should however be noted that by growing the number of web services in the repository, it will be inevitable to automate the processes of service discovery and composition due to time-consuming nature of such processes to be performed by human operators. But, the most important obstacle in automating such processes is that web services may be described using various vocabularies. Humans contrary to computers are able to understand the semantic relations among concepts and properties. To overcome this important obstacle, standard logic-based knowledge representation languages have been emerged making it possible to represent knowledge for software systems and describe entities in a way that they can be understood by them. So, software systems can reason over the knowledge and the semantic descriptions of the entities by exploiting some reasoning engines which implement the required logic-based reasoning rules [2, 5, 16, 17, 21, 23, 27, 28, 37]. Ontology languages are type of knowledge representation languages which can be used for describing the concepts and the conceptual relationships among them. Despite the apparent differences, many of the current ontology languages can be regarded as tractable and decidable subsets of Description Logics (DL). Each concept in an ontology encapsulates a subset of instance data from the domain of discourse [9]. So, it is meaningful to measure the extent to which two concepts overlap or share instances in common.

For automating the process of web service matching and composition, we need to semantically describe web services using ontology languages. Currently, web services besides their syntactic specifications or descriptions by standard syntactic models such as WSDL, are semantically specified or described by standard semantic models such as OWL-S and SAWSDL [16, 34]. Intuitively, an effective matching of web services involves considering all of their functional and non-functional requirements specified in their descriptions and interfaces, but the most crucial part of web services matching, is their signatures matching (i.e. inputs, outputs, preconditions, and effects). We are particularly interested to semantically describe the inputs and outputs of web services, because these two parameters are crucial in the process of matching web services for composing new web services [2, 13, 16, 18, 23, 28].

So far, many similarity measures have been proposed in the literature that can be divided into three major categories: 1) Semantic similarity measures which measure similarity by using and handling the semantic descriptions or relations of concepts in ontologies or taxonomies, 2) Syntactic similarity measures which measure similarity based on the syntax of text strings, and 3) Combined measures which combine semantic and syntactic approaches. In our research, we have only focused on semantic similarity measures. As it is demonstrated in this paper, in order to improve the process of semantic matching of web services, the semantic similarity/dissimilarity measures generally need to measure the extent to which the second concept includes instances which are also included by the first concept through handling the expressivity of the DL based ontology language such as OWL used for describing concepts and roles (properties) in ontologies. As we have conceptually shown in this paper, logic-based semantic similarity measures can be perfect for computing the similarity between concepts in the field of web service retrieval (i.e. web services matching and composition) if and only if they can adequately handle the expressivity of the used ontology language in order to estimate the pseudo ideal similarity metric introduced in Section 4.

The contributions of the presented research work are as follows:

1) The formulation of a pseudo ideal semantic similarity metric in Section 4, which fulfills the requirements for similarity measurement in the field of web services retrieval.
2) A set of Canonization Rules for transforming the concept descriptions to canonical form, presented in Section 5.
3) Theoretical hypotheses for the behavior of logic-based similarity measures, presented in Section 6, that provides a conceptual view for evaluating semantic similarity measures in the field of web service retrieval.

Our research work, presented in this paper, is a qualitative research in essence, because in this paper, we primarily extend the previously proposed theories for logic based matching of web services that were based on simple subsumption reasoning.

In the next section, we review DL-based ontology languages specifically the constructs of OWL ontology language. In Section 3, we present and analyze the requirements for similarity measurement in the field of web service retrieval. In Section 4, we present our proposed pseudo ideal semantic similarity metric. In Section 5, we present our proposed set of canonization rules based on the definitions provided in Section 2. In Section 6, we explain our theoretical hypotheses regarding the behavior of logic based similarity measures in the field of web service retrieval. In Section 7, we review the related works on similarity measurement in comparison with our proposed ideal semantic similarity metric. Finally in Section 8, we conclude this paper with the future works.

2. Ontology Languages and Description Logics

The semantic similarity/dissimilarity between concepts is computed based on their semantic descriptions in ontologies. Thus far many ontology languages have been proposed and standardized such as RDF(S) and OWL for defining concepts and their conceptual relations in ontologies. Despite the apparent differences, many of the current ontology languages can be regarded as tractable and decidable subsets of description logics. ***Description logics (DLs) are a family of knowledge representation languages. They are based on the notion of concepts and roles, and are mainly characterized by constructors that allow complex concepts and roles to be built from atomic (primitive) ones*** [9, 17, 27]**.**

We assume that resources, concepts and their relations are defined in terms of a generic ontology language that can be mapped to some DL language with the standard model-theoretic semantics [12]. In the reference DL framework, a knowledgebase $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ contains a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. $\mathcal{T}$ is a set of concept definitions: $C \equiv D$, where C is the atom denoting the defined concept and D is a DL concept description specified by the application of the language constructors to (primitive) concepts and roles. The complexity of such definitions depends on the specific DL language. $\mathcal{A}$ contains assertions (ground facts) on individuals (domain objects) concerning the current world state, namely $C(x)$

(class-membership) means that x is an instance of concept C, and $R(x, y)$ (relations) means that x is R-related to y. A set-theoretic semantics is generally adopted with these representations, with interpretation I as a couple $(\Delta^{\mathrm{I}}, \varphi^{\mathrm{I}})$ where the nonempty set Δ^{I} is the domain of objects (extension) and the φ^{I} function maps each concept description C to a subset of Δ^{I} i.e., $\mathrm{C}^{\mathrm{I}} \subseteq \Delta^{\mathrm{I}}$, and each role description R to a subset of $\Delta^{\mathrm{I}} \times \Delta^{\mathrm{I}}$ i.e. $\mathrm{R}^{\mathrm{I}} \subseteq \Delta^{\mathrm{I}} \times \Delta^{\mathrm{I}}$ [9, 12]. In this context, given two concept descriptions C and D, C and D are equivalent (denoted by $C \equiv D$) if and only if for every interpretation I it holds that $\mathrm{C}^{\mathrm{I}} = \mathrm{D}^{\mathrm{I}}$, C is disjoint from D (denoted by $C \perp D$) if and only if for every interpretation I it holds that $\mathrm{C}^{\mathrm{I}} \cap \mathrm{D}^{\mathrm{I}} = \emptyset$, D subsumes C (denoted by $C \sqsubseteq D$) if and only if for every interpretation I it holds that $\mathrm{C}^{\mathrm{I}} \subseteq \mathrm{D}^{\mathrm{I}}$, C and D are overlapped (denoted by $C \,\&\, D$) if and only if for every interpretation I it holds that $\mathrm{C}^{\mathrm{I}} \cap \mathrm{D}^{\mathrm{I}} \neq \emptyset$. An interpretation that satisfies all axioms of the knowledge base $\mathcal{K}$ is called a model of $\mathcal{K}$.

In our research, we narrowed our focus to OWL ontology language, because *Description logics* form its formal foundation and OWL has been being endorsed by the semantic web initiative [22]. In 2009, W3 Consortium produced a recommendation for a new version of OWL which adds features to the 2004 version, while remaining compatible. Some of the new features are syntactic sugar while others offer new expressivity, including: keys, property chains, richer datatypes and data ranges, qualified cardinality restrictions, and asymmetric, reflexive, and disjoint properties [22]. While handling the expressivity of more expressive DL-based ontology languages such as OWL 2 is desired for computing the similarity/dissimilarity between concepts, but it can be achieved by long term, ongoing research efforts [11, 12, 27].

In this paper, we present a similarity/dissimilarity measure which tries to handle the expressivity of OWL DL to a considerable extent. OWL DL is one of the three sub-species of OWL 1 and it is based on $SHOIN$ DL. Let CN denotes a concept name, C and D are arbitrary concepts, R is a property, n is a non-negative integer, d and o_i $(1 \leq \mathrm{i} \leq \mathrm{n})$ are instances, and T and $\emptyset$ denote the top (i.e. Thing) and the bottom (i.e. empty class) respectively. Then, a $SHOIN$ concept is [9, 22, 27]:

$$\mathrm{CN} \mid \mathrm{C} \sqcap \mathrm{D} \mid \mathrm{C} \sqcup \mathrm{D} \mid \neg \mathrm{C} \mid \exists R.C \mid \forall R.C \mid \ni R.\mathrm{d} \mid =_n R.T \mid \geq_n R.T \mid \leq_n R.T \mid \{o_1, \ldots, o_n\}$$

In Table 1, the formal representations of the most important constructs in OWL ontology language are shown [9, 27]. In this table, C_1, C_2, and D are concepts (classes), R is a property, T is the top (Thing), d is an instance, and n is a non-negative integer.

OWL Construct	Formal Representation
owl:equivalentClass	$C_1 \equiv C_2$
owl:disjointWith	$C_1 \perp C_2$
owl:complementOf	$C_1 \equiv -C_2$
owl:subClassOf	$C_1 \sqsubseteq C_2$
owl:intersectionOf	$C_1 \sqcap C_2$
owl:unionOf	$C_1 \sqcup C_2$
owl:minCardinality	$\geq_n R.T$
owl:maxCardinality	$\leq_n R.T$
owl:cardinality	$=_n R.T$
owl:allValuesFrom	$\forall R.D$
owl:someValuesFrom	$\exists R.D$
owl:hasValue	$\ni R.\mathrm{d}$

Table 2.1 – OWL constructs and their formal representations

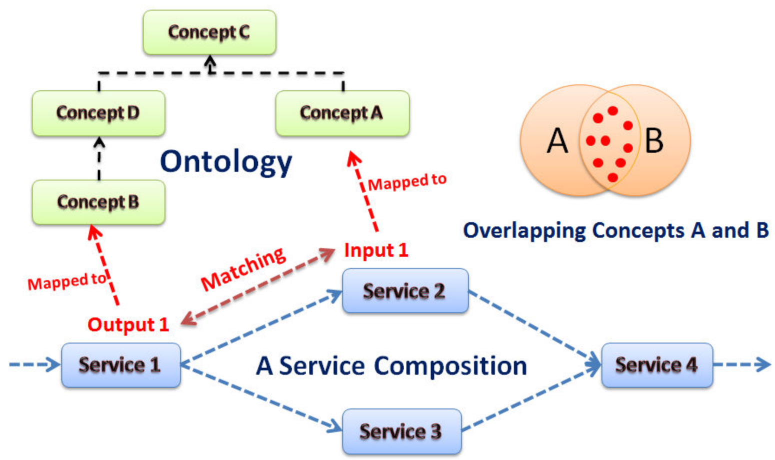
Assume that ontology O is mapped to a DL knowledgebase $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ that contains a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, and the interpretation I = (Δ^{I}, φ^{I}) is a model of $\mathcal{K}$, and for each $x \in \Delta^{\mathrm{I}}$, $\mathcal{L}(x)$ is the set of all concept descriptions like $C \in \mathcal{T}$ for which we have $x \in C^{\mathrm{I}}$. Then for a given concept name CN from O, we define $\mathcal{L}_{\mathrm{GI}}'(x_{CN})$ as follows:

$$\mathcal{L}_{\mathrm{GI}}'(x_{CN}) = \bigcap_{x \in CN^{\mathrm{I}}} \mathcal{L}(x). \qquad (2.1)$$

$\mathcal{L}_{\mathrm{GI}}'(x_{CN})$ is a model of CN based on interpretation I that is a set of all concept descriptions which can be used for describing CN by considering CN as subclass of them. $\mathcal{L}'(x_{CN})$ is also a model of CN if $\mathcal{L}'(x_{CN}) \subseteq \mathcal{L}_{\mathrm{GI}}'(x_{CN})$ and the descriptors of $\mathcal{L}'(x_{CN})$ logically seem necessary and sufficient for defining CN and if so, we refer to $\mathcal{L}'(x_{CN})$ as a canonical form of the description of CN. The canonization process is a process in which we convert the description of concepts to its canonical form by extracting the models of those concepts and standardizing and compressing the semantic descriptions of those models as much as possible based on the semantics of the constructs of the used DL-based ontology language.

3. Analysis of Web Services Matching and Composition

The problem of matching single web services with a web service request has been investigated in many papers. But in our research, we have focused on a much more complicated problem that is the problem of matching web services with each other in order to integrate them and compose a new web service. [5, 13, 18, 28, 38]. The most crucial part of web services matching, is their signatures matching (i.e. inputs, outputs, preconditions, and effects). We are particularly interested to semantically describe the inputs and outputs of web services, because these two parameters are crucial in the process of matching web services for composing new web services [2, 13, 16, 18, 23, 28].



Picture 3.1 – The semantic similarity measure have to be defined based on the extent to which the second concept includes instances which are also included by the first concept, so that it can be effectively applicable in semantic matching of web services for web service composition.

Assume that web service S_1 has an input that its type has been defined as concept A from ontology O_1 in the semantic description of S_1, and web service S_2 has an output that its type has been defined as concept B from ontology O_2 in the semantic description of S_2. We want to determine the degree of match between these two web services since we may compose a new web service using them in a way that the output resulted from executing S_2 will work as an input to S_1. If the output resulted from executing the service S_2 is an instance of A, we can be certain about the successful execution of S_1 without any failure. At least, in this situation, the semantic description of S_1 guarantees it [16, 28]. Therefore, considering the fact that the output of S_2 is always an instance of B, in order to determine the degree of match between the two web services, we need to determine the extent to which A includes instances which are also included by B.

If A subsumes B, we are certain about the successful execution of S_1 with any input provided by S_2 to S_1. But, we do not want to ignore the situations in which A does not subsume B but the two concepts overlap, because in such situations, the successful execution of S_1 is still possible if the input provided by S_2 to S_1 is from the intersection of A and B.

Hence, we need to define a new similarity measure which supports the fact that the more the extent to which the second concept includes instances which are also included by the first concept, the more similar the two concepts are. It is clear that such a similarity measure is asymmetric and considers the direction. Such a matching approach considers the state in which two concepts (classes) overlap, as a level of match higher than the disjoint level even if none of the two concepts subsumes the other. Most of approaches to semantic matching are based on simple subsumption reasoning and do not consider such states as a level of match [2, 5, 10, 16, 21, 23, 37].

Web services preconditions/effects matching is more complicated than inputs/outputs matching. The authors in [2], presents a solution to precondition/effect matching. Considering their work, preconditions/effects are decomposable to atom lists and atom lists are decomposable to atoms. Atom lists are horn clauses like ‘‘Customer {hasCard} VisaCard”, and the atoms are like ‘‘Customer” and ‘‘VisaCard”. Therefore, every precondition/effect is represented as a property predicate (e.g. hasCard) beside a set of arguments (e.g. {Customer, VisaCard}). Then similarity values between arguments of the preconditions/effects in the request and arguments of the preconditions/effects in the service description are computed (please refer to [2] for details). However, their similarity/dissimilarity measure is essentially grounded on subsumption reasoning and network distance based model (please refer to Section 7 for definition) and therefore cannot be perfect, whereas like input/output matching, we propose a similarity/dissimilarity measure which can be perfect since it is based on the extent to which the two concepts overlap and therefore it does not ignore the situations in which none of the two arguments subsumes the other, but they share some instances in common and as a result the web service preconditions might be satisfied with some of the requests sent to it from other web services.

Hence, in order to improve the process of semantic matching of web services based on their semantic descriptions, the semantic similarity/dissimilarity measures need to fulfill three criteria. First, the similarity/dissimilarity between two concepts should be determined based on the extent to which the two concepts overlap. Second, the direction should be considered by the used measure that means the similarity/dissimilarity from the first concept to the second one is not necessarily equal to the similarity/dissimilarity from the second concept to the first one [27, 31]. Third, they have to be able to handle the expressivity of the description logics used for describing web service parameters i.e. web service Inputs, Outputs, Preconditions, and Effects (IOPEs) in ontologies [16, 17, 27]. In general, the sought semantic similarity/dissimilarity measures need to measure the extent to which the second concept includes instances which are also included by the first concept, through handling the expressivity of the DL-based ontology language such as OWL DL used for describing concepts and roles (properties) in ontologies.

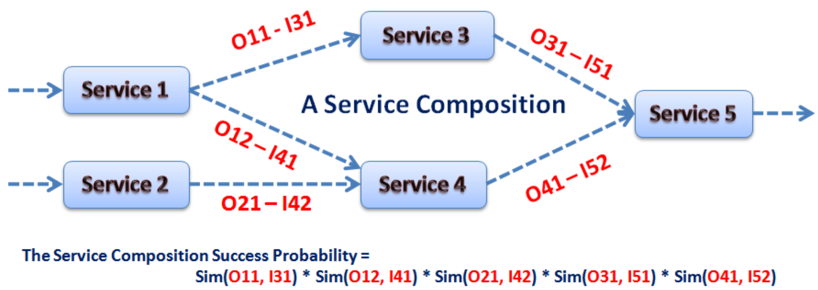
However, a web service may have a number of inputs, outputs, preconditions, and effects and we may compose a complex web service by matching and connecting a number of other web services in a workflow. Hence, the problem of matching the semantic descriptions of web services for web service composition is more complicated than computing the similarity between two concepts although the latter is used as a fundamental operation for solving the former [2, 10, 16, 17, 18, 21, 23, 28, 29, 33, 37, 38]. In our research, we have only focused on the problem of computing the similarity/dissimilarity between two concepts in order to define an ideal semantic similarity metric which fulfills the requirements for similarity measurement in the field of web service retrieval. But here we want to give an intuition about how the target similarity measure can be used in the process of web service composition. So, our purpose is to generally demonstrate the applicability of such a similarity measure in this application area by proposing an aggregation scheme, and clearly we do not intend to give a complete solution to web service composition.

Considering the workflow of a proposed composite web service depicted in Picture 3.2, Iij and Oij respectively represent the jth input and output of the ith service in the proposed web service composition. We need to compute the probability for successful execution of this proposed service composition. We propose the following simple aggregation scheme to aggregate the similarity values computed for each represented output/input pair together into the total probability value for successful execution of the composite web service:

Success_Probability(SC) = $\prod Sim(Oij, Ii'j')$ (3.1)

$0 \leq Sim(C, D) \leq 1$

Where service i and service i' are connected in the proposed service composition SC and jth output of service i namely Oij is provided as input to service i' to satisfy the j' input of service i' namely $Ii'j'$.



Picture 3.2 – This picture shows how the probability for successful execution of the composite service can be theoretically computed using the semantic similarity values computed for the input-output pairs of the services which are connected in the workflow of the composite service.

4. Proposed Pseudo Ideal Semantic Similarity Metric

The authors in [12], presents a framework which explains how (inter-concept) similarity is measured and it consists of the following seven steps:

1. Definition of application area and intended audience
2. Selection of search (query) and target concepts
3. Transformation of concepts to canonical form
4. Definition of an alignment matrix for concept descriptors
5. Application of constructor specific similarity functions
6. Determination of standardized overall similarity
7. Interpretation of the resulting similarity value(s)

The authors in [12] argue that every similarity measure should define in which way it implements these steps and thereby specifies the semantics of similarity (values) as well as its properties. This framework allows for a better separation between the process of measuring similarity (i.e., what is measured) and the used similarity functions (i.e., how it is measured).

In our research, we consider semantic similarity/dissimilarity in the broader sense of any set of criteria which may be used for comparing concepts with respect to their semantics. But, we seek similarity/dissimilarity measures which are suitable to be used for semantic web services matching and composition (i.e. web service retrieval). So, in the presented research, the application area is the field of web service retrieval. Search (query) concepts can be regarded as concepts which define the output types of web services whose outputs are sent as inputs to other services in the workflow of composed web services, and target concepts can be regarded as concepts which define the input types of web services whose inputs are provided by other services in the workflow of composed web services. Since our proposed ideal similarity metric, presented in this section, is asymmetric, distinguishing between search and target concepts is necessary. In addition, search and target concepts may be from different ontologies.

In this section, we define a pseudo ideal semantic similarity metric which ideally expresses or formulates the requirements for similarity/dissimilarity measurement in this application area. We call it "pseudo ideal similarity metric" since it cannot be generally actually computed, but can ideally express what we want to measure. It is defined as follows:

$$\mathrm{DD}(A, B, \mathrm{I}) = \frac{|A^{\mathrm{I}} \cup B^{\mathrm{I}}|}{|A^{\mathrm{I}} \cap B^{\mathrm{I}}|} * \frac{|A^{\mathrm{I}} - B^{\mathrm{I}}|}{|A^{\mathrm{I}}|}; \tag{4.1}$$

$$\mathrm{MHD}(A, B, \mathrm{I}) = \text{Minimum hierarchical distance between } A \text{ and } B; \tag{4.2}$$

$$\mathrm{Dissim}(A, B, \mathrm{I}) = \mathrm{DD}(A, B, \mathrm{I}) + \mathrm{MHD}(A, B, \mathrm{I}); \tag{4.3}$$

$$\mathrm{Sim}(A, B, \mathrm{I}) = \frac{\mu}{\mu + \mathrm{Dissim}(A,\ B,\ I)};\quad \mu > 0 : \text{adjustable factor} \tag{4.4}$$

Where A and B are concepts from the same or different ontologies. Based on an interpretation like $\mathrm{I} = (\Delta^{\mathrm{I}}, \varphi^{\mathrm{I}})$ for the ontology (or ontologies), A is mapped to A^{I} and B is mapped to B^{I}. |A| denotes the size of the set A. In the equation (4.1), if $A^{\mathrm{I}} \equiv \emptyset$ or $B^{\mathrm{I}} \equiv \emptyset$ or $A^{\mathrm{I}} \cap B^{\mathrm{I}} \equiv \emptyset$ ($\emptyset$ means empty), then we consider DD(A, B, I) as infinite (∞). The dissimilarity from A to B represented as Dissim(A, B, I), is sum of DD(A, B, I) and MHD(A, B, I). DD(A, B, I) can be regarded as definition distance between A and B, but here, it is exactly what represented in the equation (4.1). MHD(A, B, I) denotes the minimum hierarchical distance between A and B in the ontological hierarchy after classification. Sim(A, B, I) denotes the semantic similarity from A to B. Dissim(A, B, I) and Sim(A, B, I) can be converted to each other using the equation (4.4). μ is an adjustable factor. While Dissim(A, B, I) is ranged from 0 to ∞ (infinite), Sim(A, B, I) is ranged from 0 to 1. If A and B are from the two different ontologies O_1 and O_2 respectively and T(O_1) and T(O_2) are the roots of these two ontologies, then MHD(A, B, I) can be computed as follows: |MHD(A, T(O_1), I) - MHD(B, T(O_2), I)| or a more perfect computational scheme be used based on the depth or granularity of the respective ontological hierarchies [36]. In our research, we have not focused on such alignment schemes and leave it with the above simple scheme. According to these equations, we have:

$A^{\mathrm{I}} \sqsubseteq -B^{\mathrm{I}}$ (A^{I} and B^{I} are disjoint) ⟹ DD(A, B, I) = ∞ (infinite) ⟹ Dissim(A, B, I) = ∞ ⟹ Sim(A, B, I) = 0;

$A^{\mathrm{I}} \equiv B^{\mathrm{I}}$ (A^{I} and B^{I} are equivalent) ⟹ DD(A, B, I) = 0, MHD(A, B, I) = 0 ⟹ Dissim(A, B, I) = 0 ⟹ Sim(A, B, I) = 1;

$A^{\mathrm{I}} \sqsubset B^{\mathrm{I}}$ (B^{I} subsumes A^{I}) ⟹ DD(A, B, I) = 0;

$B^{\mathrm{I}} \sqsubset A^{\mathrm{I}}$ (A^{I} subsumes B^{I}) ⟹ DD(A, B, I) = $\frac{|A^{\mathrm{I}} - B^{\mathrm{I}}|}{|B^{\mathrm{I}}|}$;

MHD(A, B, I) function complements DD(A, B, I) function by considering the hierarchical distance between two concepts in the ontology. For instance, if B^{I} subsumes A^{I}, DD(A, B, I) is 0, but we intuitively know that the semantic dissimilarity between A^{I} and B^{I} cannot be 0 unless A^{I} and B^{I} are equivalent [27]. The pseudo ideal similarity metric meets this requirement by considering the semantic dissimilarity Dissim(A, B, I) as sum of DD(A, B, I) and MHD(A, B, I).

Considering the equations, the following results are clear:

1) The ideal similarity metric measures the similarity by the extent to which the second concept includes instances which are also included by the first concept and therefore it fulfills the requirements for similarity/dissimilarity measurement in the field of web service retrieval mentioned in Section 3.

2) The ideal similarity metric has not been defined based on a specific interpretation. In other words, we consider each concept as set of instances from the domain of an interpretation which is chosen arbitrarily. So, in order to theoretically get the value of the ideal measure, we first need to choose a specific interpretation that may affect the value resulted from the function.

3) The ideal similarity metric is not generally computable because it is defined using operators of set theory such as union, intersection, difference, and specifically the cardinality or size of sets that are not directly computable since they are applied to ontology concepts when we consider concepts as subsets of the

interpretation domain objects i.e. Δ^{I} which is undetermined. For instance, considering two overlapping concepts ($vertebrate - bird$) and ($vertebrate - reptile$), in order to compute the similarity from the first one to the second one, first we need to compute the value of the following expression:

($|vertebrate^I|$ / $|vertebrate^I$ - $bird^I$ - $reptile^I|$) * ($|reptile^I|$ / $|vertebrate^I$ - $bird^I|$) = ($|vertebrate^I|$ * $|reptile^I|$) / ($|vertebrate^I$ - $bird^I$ - $reptile^I|$ * $|vertebrate^I$ - $bird^I|$)

In fact, we have defined the pseudo ideal similarity metric to ideally express our approach to similarity measurement using mathematical symbols to only make our approach more understandable. So, when we speak about the estimation of such a pseudo ideal measure, we just mean doing computations which lead to the values with the semantics represented by such a pseudo ideal measure considering what it really tries to measure. Hence, comparing a computable logic based similarity measure with this ideal measure in order to investigate how much imprecision we get, is meaningless. In other words, we can only compare computable logic based similarity measures with each other to investigate how well they are able to estimate the ideal measure relative to each other.

5. Transformation of concepts to canonical form

Since logic-based similarity measures compute the similarity/dissimilarity between concepts by comparing their semantic definitions in ontologies, the concept definitions have to be rewritten to a common form to eliminate syntactic influence [9, 12]. It is done in a canonization procedure generally defined in Section 2 and in this section we introduce a specific canonization procedure.

The concept name CN from the OWL DL ontology O, may be defined as subclass or equivalent class of other concepts (i.e. classes) or descriptors, or subclass or equivalent class of intersection or union of a number of other concepts or descriptors. Assume that $\mathrm{S}(CN)$ is the set of all concepts or descriptors like C' which have been explicitly defined as superclass or equivalent class of CN in O (i.e. $CN \sqsubseteq C'$). In addition, CN may be defined by placing some restrictions on some properties. The various types of such property restrictions in OWL DL ontologies are as follows: $\exists R.C$, $\forall R.C$, $\ni \mathrm{R.d}$, $\geq_n R.T$, $\leq_n R.T$, $=_n R.T$ [22]. Assume that $\mathrm{R}(CN)$ is the set of all property restrictions used for defining CN. Also Assume that $\mathrm{D}(CN)$ is the set of all concepts like $-D$ that D has been explicitly stated in the ontology that it is disjoint with CN (i.e. $CN \sqsubseteq -D$). We separate disjoint concepts from others, because the descriptors represented as $-D$ might be handled in a different manner by descriptor-specific functions of some logic-based measures.

We initially consider the $\mathcal{L}^{'}(x_{CN})$ set as the union of $\mathrm{S}(CN)$, $R(CN)$, and $D(CN)$. It is clear that $\mathcal{L}^{'}(x_{CN})$ is the semantic description of CN. Then, we follow the canonization rules presented in this section to extract the canonical form of the description of CN from the initial set. The disjunction (union) construct ($\sqcup$) is handled by generating three possible models from the primary one. One of these models allows an individual for being an instance of both concepts participating in the union, while the two others only allow an individual for being an instance of one of the two concepts participating in the union [12]. Then, for each generated model, the canonization rules will be executed independently. The rules with the less number have higher priority in execution. At the end of this procedure, a number of models may be generated from the description of the given concept CN. Some of these models might have clashes (e.g. C⊓–C) and not be satisfiable, therefore they are ignored within the process of computing similarity/dissimilarity between models of two compared concepts. The canonization rules are as follows:

Rule 0 (start) - Action: $\mathcal{L}^{'}(x_{CN}) \coloneqq \mathrm{S}(CN) \cup R(CN) \cup D(CN)$

Rule 1 (⊓) - Condition: $C_1 \in \mathcal{L}^{'}(x_{CN}) \ \wedge\ C_1 \sqsubseteq C_2$

Action: $\mathcal{L}^{'}(x_{CN}) \coloneqq \mathcal{L}^{'}(x_{CN}) \cup \{C_2\}$

Rule 2 (⊓) - Condition: $C_1 \sqcap C_2 \in \mathcal{L}^{'}(x_{CN})$

Action: $\mathcal{L}^{'}(x_{CN}) \coloneqq (\mathcal{L}^{'}(x_{CN}) - \{C_1 \sqcap C_2\}) \cup \{C_1, C_2\}$

Rule 3 (⊔) - Condition: $C_1 \sqcup C_2 \in \mathcal{L}^{'}(x_{CN})$

Action: $\mathcal{L}^{'}(y_{CN}) \coloneqq (\mathcal{L}^{'}(x_{CN}) - \{C_1 \sqcup C_2\}) \cup \{C_1\}$,

$\mathcal{L}^{'}(w_{CN}) \coloneqq (\mathcal{L}^{'}(x_{CN}) - \{C_1 \sqcup C_2\}) \cup \{C_2\}$,

$\mathcal{L}^{'}(z_{CN}) \coloneqq (\mathcal{L}^{'}(x_{CN}) - \{C_1 \sqcup C_2\}) \cup \{C_1, C_2\}$

Rule 4 (∀) - Condition: $\forall \mathrm{R.G},\ \forall \mathrm{R.G'} \in \mathcal{L}^{'}(x_{CN})$

Action: $\mathcal{L}^{'}(x_{CN}) \coloneqq (\mathcal{L}^{'}(x_{CN}) - \{\forall \mathrm{R.G}, \forall \mathrm{R.G'}\}) \cup \{\forall \mathrm{R.(G \sqcap G')}\}$

Rule 5 (∀) - Condition: $\forall \mathrm{S.G},\ \forall \mathrm{R.G'} \in \mathcal{L}^{'}(x_{CN}) \ \wedge\ S \sqsubset \mathrm{R}$

Action: $\mathcal{L}^{'}(x_{CN}) \coloneqq (\mathcal{L}^{'}(x_{CN}) - \{\forall \mathrm{S.G}\}) \cup \{\forall \mathrm{S.(G \sqcap G')}\}$

Rule 6 (∃) - Condition: $\exists \mathrm{S.G},\ \forall \mathrm{R.G'} \in \mathcal{L}^{'}(x_{CN}) \ \wedge\ S \sqsubseteq \mathrm{R}$

Action: $\mathcal{L}^{'}(x_{CN}) \coloneqq (\mathcal{L}^{'}(x_{CN}) - \{\exists \mathrm{S.G}\}) \cup \{\exists \mathrm{S.(G \sqcap G')}\}$

Rule 7 (∋) - Condition: $\ni \mathrm{R.d},\ \exists \mathrm{R.G} \in \mathcal{L}^{'}(x_{CN}) \ \wedge\ \mathrm{d} \in G$

Action: $\mathcal{L}^{'}(x_{CN}) \coloneqq \mathcal{L}^{'}(x_{CN}) - \{\exists \mathrm{R.G}\}$

Rule 8 ($\geq_n$ - 1) - Condition: $\geq_m S.T \in \mathcal{L}^{'}(x_{CN}) \ \wedge\ S \sqsubseteq \mathrm{R}$

Action: $\mathcal{L}'(x_{CN}) \coloneqq \mathcal{L}'(x_{CN}) \cup \{\geq_m R.T\}$

Rule 9 ($\geq_n$ - 2) - Condition: $\geq_n R.T$, $\geq_m R.T \in \mathcal{L}'(x_{CN}) \wedge$ m $> n$

Action: $\mathcal{L}'(x_{CN}) \coloneqq \mathcal{L}'(x_{CN}) - \{\geq_n R.T\}$

Rule 10 ($\geq_n$ - 3) - Condition: $\geq_n R.T$, $=_m R.T \in \mathcal{L}'(x_{CN})$

Action: $\mathcal{L}'(x_{CN}) \coloneqq \mathcal{L}'(x_{CN}) - \{\geq_n R.T\}$

Rule 11 ($\leq_n$ - 1) - Condition: $\leq_m S.T \in \mathcal{L}'(x_{CN}) \wedge R \sqsubseteq S$

Action: $\mathcal{L}'(x_{CN}) \coloneqq \mathcal{L}'(x_{CN}) \cup \{\leq_m R.T\}$

Rule 12 ($\leq_n$ - 2) - Condition: $\leq_n R.T$, $\leq_m R.T \in \mathcal{L}'(x_{CN}) \wedge$ m < n

Action: $\mathcal{L}'(x_{CN}) \coloneqq \mathcal{L}'(x_{CN}) - \{\leq_n R.T\}$

Rule 13 ($\leq_n$ - 3) - Condition: $\leq_n R.T$, $=_m R.T \in \mathcal{L}'(x_{CN})$

Action: $\mathcal{L}'(x_{CN}) \coloneqq \mathcal{L}'(x_{CN}) - \{\leq_n R.T\}$

Rule 14 ($-$ 1) - Condition: $C \in \mathcal{L}'(x_{CN}) \wedge C \sqsubseteq C' \wedge -D' \in \mathcal{L}'(x_{C'})$

Action: $\mathcal{L}'(x_{CN}) \coloneqq \mathcal{L}'(x_{CN}) \cup \{-D'\}$

Rule 15 ($-$ 2) - Condition: $-D$, $-D' \in \mathcal{L}'(x_{CN})$

Action: $\mathcal{L}'(x_{CN}) \coloneqq (\mathcal{L}'(x_{CN}) - \{-D, -D'\}) \cup \{-(\mathrm{D} \sqcup \mathrm{D}')\}$

Rule 16 ($\{o_1, \ldots, o_n\}$) - Condition: $\{o_1, \ldots, o_n\}$, $\{o'_1, \ldots, o'_{n'}\} \in \mathcal{L}'(x_{CN})$

Action: $\mathcal{L}'(x_{CN}) \coloneqq \mathcal{L}'(x_{CN}) - \{\{o_1, \ldots, o_n\}, \{o'_1, \ldots, o'_{n'}\}\}$

$\mathcal{L}'(x_{CN}) \coloneqq \mathcal{L}'(x_{CN}) \cup \{\{o_1, \ldots, o_n\} \cap \{o'_1, \ldots, o'_{n'}\}\}$

After executing the above rules for the given concept description, each generated model can be represented in a canonical form as follows:

$$\mathcal{L}'(x_{CNi}) = \{ C_{i1}, C_{i2}, \ldots, C_{in_i}, [\forall \mathrm{R}_1.\mathrm{G}_{i1}]_1, [\forall \mathrm{R}_2.\mathrm{G}_{i2}]_1, \ldots, [\forall \mathrm{R}_k.\mathrm{G}_{ik}]_1,$$
$$[\exists \mathrm{R}_1.\mathrm{H}_{i1} \mid \ni \mathrm{R}_1.\mathrm{d}_{i1}]_\infty, [\exists \mathrm{R}_2.\mathrm{H}_{i2} \mid \ni \mathrm{R}_2.\mathrm{d}_{i2}]_\infty, \ldots, [\exists \mathrm{R}_k.\mathrm{H}_{ik} \mid \ni \mathrm{R}_k.\mathrm{d}_{ik}]_\infty,$$
$$\left[\geq_{m_1} \mathrm{R}_1.T \,\middle\|\, =_{s_1} \mathrm{R}_1.T\right]_1, \left[\geq_{m_2} \mathrm{R}_2.T \,\middle\|\, =_{s_2} \mathrm{R}_2.T\right]_1, \ldots, \left[\geq_{m_k} \mathrm{R}_k.T \,\middle\|\, =_{s_k} \mathrm{R}_k.T\right]_1,$$
$$\left[\leq_{r_1} \mathrm{R}_1.T \,\middle\|\, =_{s_1} \mathrm{R}_1.T\right]_1, \left[\leq_{r_2} \mathrm{R}_2.T \,\middle\|\, =_{s_2} \mathrm{R}_2.T\right]_1, \ldots, \left[\leq_{r_k} \mathrm{R}_k.T \,\middle\|\, =_{s_k} \mathrm{R}_k.T\right]_1,$$
$$[-D]_1, [\{o_1, \ldots, o_p\}]_1\} \qquad (5.1)$$

Where $\mathcal{L}'(x_{CNi})$ is a model of CN that is represented as a set of descriptors which define or describe that model. The instances of a model are the instances of the concept which is defined by that model. In the above representation for models, C_{ij} is a primitive concept ($1 \leq j \leq n_i$). $[\forall \mathrm{R}_k.\mathrm{G}_{ik}]_1$ represents the fact that for each property like R_k, there is at most one ∀-type statement in the description of every generated model considering Rule 4. $[\exists \mathrm{R}_k.\mathrm{H}_{ik} \mid \ni \mathrm{R}_k.\mathrm{d}_{ik}]_\infty$ represents the fact that for each property like R_k, there is not any limit for the number of ∃-type and ∋-type statements in the description of every generated model, but the presence of ∋-type statements may affect the existence of ∃-type statements considering Rule 7. $\left[\geq_{m_\mathrm{k}} \mathrm{R_k}.T \,||\!=_{s_\mathrm{k}} \mathrm{R_k}.T\right]_1$ represents the fact that for each property like $\mathrm{R_k}$, there is at most one $\geq_{m_\mathrm{k}}$-type (i.e. minimum cardinality) or $=_{s_\mathrm{k}}$-type (i.e. exact cardinality) statement in the description of every model, and the existence of both is not possible considering Rules 9 and 10. $\left[\leq_{r_\mathrm{k}} \mathrm{R_k}.T \,||\!=_{s_k} \mathrm{R_k}.T\right]_1$ represents the facts similar to the ones mentioned for $\left[\geq_{m_\mathrm{k}} \mathrm{R_k}.T \,||\!=_{s_\mathrm{k}} \mathrm{R_k}.T\right]_1$ but by considering Rules 12 and 13. Finally $[-D]_1$ and $[\{o_1, \dots, o_p\}]_1$ represents that there is at most one negation (i.e. disjoint) and enumeration-type statement in the description of every model considering Rules 15 and 16. By executing the above rules, the logical descriptions of concepts are abstracted and transformed to the same canonical form in a manner that they can be unambiguously compared with each other without any syntactic influence.

6. Theoretical Hypotheses for logic based similarity measures

There are two important dimensions along which the conditions can be changed for logic based similarity measurement and therefore along them the logic based similarity measures have to be evaluated:

1) The complexity of concept definitions in ontologies or in other words, how much the expressivity of description logics has been used for defining or describing concepts in ontologies. We call this dimension “DL Expressivity Usage”. As logic based similarity measures compute the similarity between concepts by handling the expressivity of DLs to an extent, ontologies with proper DL Expressivity Usage are needed for fair and complete evaluation of such measures.

2) The extent to which there are pairs of concepts in ontologies that overlap but none of them subsumes the other. We call this dimension “Proportion of Overlapped Concepts”. Generally, when we compare two concepts, there are three possible situations: 1) The two concepts are disjoint i.e., the intersection of them is not satisfiable, 2) One of the two concepts subsumes the other, and 3) The two concepts overlap i.e., the intersection of them is satisfiable but none of them subsumes the other. Considering all pairs of concepts in ontologies, it seems for many of existing ontologies, the proportion of concept pairs belonging to the third category is much less than the proportion of concept pairs belonging to the first and second categories. As some logic based similarity metrics, might measure the similarity/dissimilarity between two concepts based on the extent to which the two

concepts overlap, concept pairs from the third category are also needed for fair and complete evaluation of those similarity measures.

There are three important quality criteria for evaluating similarity measures in the field of web service retrieval:

1) **Precision** that is the proportion of the relevant web services retrieved by a matchmaker to all the retrieved web services. Precision is a performance criterion.

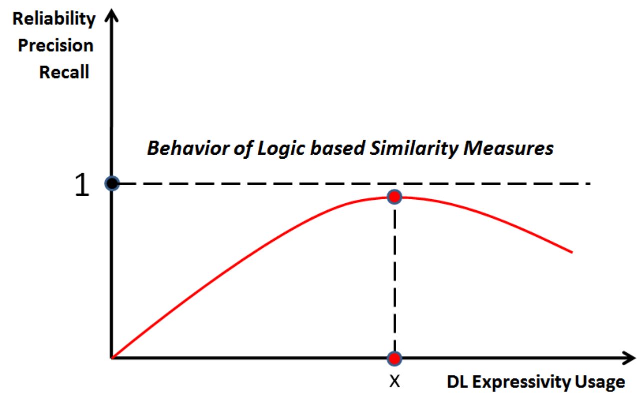
2) **Recall** that is the proportion of the relevant web services retrieved by a matchmaker to all the relevant web services. Recall is also a performance criterion.

3) **Reliability** that shows how much we can rely on the behavior of a matchmaker in various conditions which might be changed along the two aforementioned dimensions or other possible dimensions which have to be also investigated by researchers in this application area.

We consider reliability as an important quality criterion for evaluating similarity measures used in the field of web service retrieval besides the two mostly used quality criteria namely precision and recall. Because when a matchmaker uses a logic based similarity measure, the reliability highly depends on the two aforementioned dimensions namely DL Expressivity Usage and Proportion of Overlapped Concepts. It is important to know how much we can rely on the behavior of a matchmaker when DL Expressivity Usage and Proportion of Overlapped Concepts in ontologies change when the input/output types of web services have been defined in those ontologies. As far as we know, reliability has not been considered as an important quality criterion by researchers in this application area. Most of researchers evaluate their similarity measures or service matchmakers based on some specific web service test collections such as OWLS-TC, SAWSDL-TC2, or Jena Geography Dataset(JGD) which use fixed and specific ontologies, so the performance and reliability of their proposed matchmakers or similarity measures is not evaluated when the characteristics of the underlying ontologies, in which input/output types of web services have been defined, change specifically when DL Expressivity Usage and Proportion of Overlapped Concepts in ontologies change [34].

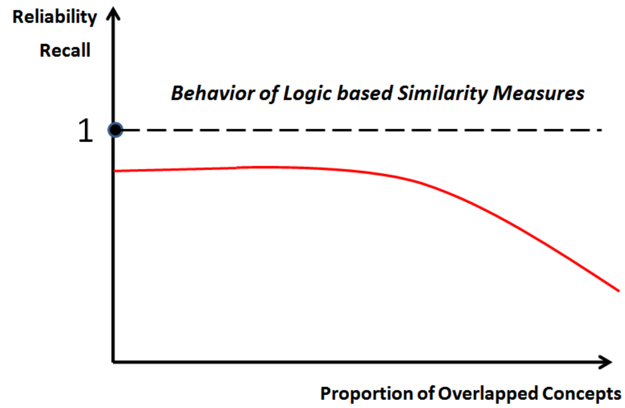
We conceptually predict the behavior of service matchmakers, which use logic based similarity measures, as depicted in Picture 6.1 when DL Expressivity

Usage in ontologies changes. This is a theoretical view which has to be simple to generally express our idea. As shown in Picture 6.1, considering an arbitrary logic based similarity measure used by a service matchmaker for computing the semantic similarity between concepts, by increasing the DL Expressivity Usage in ontologies in which concepts have been defined, first the performance (i.e., Precision and Recall) and reliability of the matchmaker increases until a point like x, but after that point it reduces.



Picture 6.1 – Conceptual prediction for the general behavior of logic based similarity measures when DL Expressivity Usage in ontologies changes.

This behavior can be conceptually explained as follows: When concepts are poorly described in ontologies without an effective usage of the expressivity of Description Logics, the logic based similarity measure fails to effectively compute the similarity between concepts and therefore the reliability and performance of the matchmaker is low, on the other hand since the ability of the used logic based similarity measure in handling the expressivity of Description Logics is limited, when concepts are described in ontologies with high usage of the expressivity of Description Logics, the logic based similarity measure fails to completely handle the description of concepts and therefore the reliability and performance of the matchmaker is also low. Hence, the behavior of the logic based similarity measure has to be similar to the diagram depicted in Picture 6.1, and there is a point like x, along the DL Expressivity Usage dimension, on which the performance and reliability of the matchmaker is at its maximum. This theoretical view, shows that the performance and reliability of logic based similarity measures highly depends on the degree of DL Expressivity Usage in ontologies.



Picture 6.2 – Conceptual prediction for the general behavior of logic based similarity measures in the field of web service retrieval when the Proportion of Overlapped Concepts in ontologies changes.

We also theoretically predict the behavior of service matchmakers, which use logic based similarity measures, as depicted in Picture 6.2 when the Proportion of Overlapped Concepts in ontologies changes. It is a theoretical view which has to be simple to generally express our idea. As shown in Picture 6.2, considering an arbitrary logic based similarity measure used by a web service matchmaker for computing the semantic similarity between concepts, by increasing the Proportion of Overlapped Concepts in ontologies, the Recall based Performance and Reliability of the matchmaker reduce.

This behavior can be conceptually explained as follows: Considering the facts explained in Sections 3 and 4, the Recall based Performance and Reliability of a web service matchmaker is related to its ability to precisely compute the similarity between two concepts by the extent to which the second concept includes instances which are also included by the first concept, as this fact has been conceptually formulated in the ideal similarity measure introduced in Section 4. But, most of logic based web service matchmakers compute the semantic similarity between concepts by simple subsumption reasoning and are not able to recognize the degree of overlap between concepts when none of them subsumes the other. Hence, when the Proportion of Overlapped Concepts in ontologies increases, most of logic based web service matchmakers fail to recall all the potentially relevant web services or all the potentially relevant web service compositions for a request. Also, in the case of semantic similarity measures which tries to estimate our proposed ideal similarity metric, since the ability of those measures in estimating the ideal measure is also limited and there may be situations in which those measures are not able to precisely compute the degree of overlap between concepts, so by increasing the Proportion of Overlapped Concepts in ontologies, after a point, the Reliability and Recall based Performance of the matchmaker which uses those similarity measures also reduce but this point is very far away from the corresponding points of most of other service matchmakers which are not able to recognize the degree of overlap between two concepts when none of the two concepts subsumes the other.

Hence, considering the aforementioned theoretical hypotheses, the experimental evaluation of logic based web service matchmakers on the basis of specific test collections such as OWLS-TC, SAWSDL-TC2, or JGD which are grounded on specific ontologies with specific DL Expressivity Usages and specific Proportions of Overlapped Concepts, is not sufficient for evaluating the performance of the respective similarity measures. In fact, the reliability of logic based web service matchmakers highly depends on the DL Expressivity Usage and the Proportion of Overlapped Concepts in those ontologies, but as far as we know, these important facts have been ignored by research works previously done in this application area [34].

7. Related works on similarity/dissimilarity measurement

Many of the proposed computational solutions for the problem of computing the semantic similarity/dissimilarity between pairs of words rely on existing hierarchical taxonomies. These taxonomies usually represent the lexical knowledge implicit in languages by means of graph structures which reflect concepts of words and their relationships. General purpose efforts to build such structures for the English language yielded hierarchical ontologies such as the well known WordNet [1, 3, 6, 7, 14, 18, 19, 24, 25, 29, 32, 33, 36, 38]. But since these taxonomies are built using simple relation constructs such as is–a, part–whole, cause–effect, and equivalence, one important drawback of the similarity/dissimilarity measures relying on them is their clear inability to handle the expressivity of DL-based ontology languages while using an expressive DL-based ontology language like OWL is necessary in order to satisfactorily define complex concepts and describe complex relationships among them [3, 15, 17, 35].

In some research papers, the existing semantic similarity/dissimilarity measures have been divided into measures for concepts from the same ontology and measures for concepts from different ontologies [36, 39]. Considering previously proposed categorizations of the existing semantic similarity/dissimilarity measures [3, 36], we can categorize them into five general categories: **1) Network distance based model** which measures the semantic similarity/dissimilarity by the geometric network distance between nodes representing concepts in an ontological hierarchical network [1, 3, 7, 14, 18, 19, 25, 29, 32, 33, 36], **2) Information content based model** which measures the semantic similarity/dissimilarity by the extent to which two concepts share information in common. The information that the two concepts share is then evaluated by the information content of their Nearest Common Ancestor (NCA) containing them, and the information content of a concept is inversely proportional to its frequency in a large text corpus [7, 18, 25, 36], **3) Attribute based model** which measures the semantic similarity by the degree to which the attribute-value sets used for describing two concepts overlap [3, 36], **4) Combined model** which combines the aforementioned three models [7, 18, 36], and **5) DL-based model** which measures the semantic similarity/dissimilarity between concepts by comparing the logic-based descriptions of them in DL-based ontologies [9, 11, 12, 17, 27].

It is clear that measures of the first four categories do not model concepts as sets of instances from the domain of discourse. So, they cannot compute the similarity/dissimilarity between two concepts based on the extent to which the two concepts share instances in common. Hence, considering the requirements mentioned in Section 3 for semantic similarity/dissimilarity measurement in the field of web service retrieval, such measures are not essentially suitable and reliable to be used in this field on which our research has been focused.

On the other hand, although the previously proposed DL-based similarity/dissimilarity measures model concepts as sets of instances, but as far as we know, most of them do not particularly go in a way that compute the semantic similarity/dissimilarity between concepts based on the extent to which two concepts overlap. In fact, it seems that most of them try to compute the semantic similarity/dissimilarity between concepts only based on the extent to which the instances of two concepts are similar considering their properties and not based on the extent to which the instances of two concepts are exactly the same. For instance, based on most of these measures, the similarity between bird and reptile is not zero because bird and reptile have some similar or same properties [9, 11, 12, 27], but in our approach the similarity between bird and reptile must be zero, because they do not have any common instance and they are disjoint.

Some of the proposed DL based approaches, such as ones presented in [9], [11] and [12], are completely or partly based on the notion of computing similarity between complex concepts by using the frequencies of primitive concepts and properties in ontology. Such frequencies for the primitives and properties are calculated by the number of their occurrences in the description of complex concepts. For instance, the authors in [11] and [12] remark that two primitives are the more similar if the more complex concepts are defined using both (and not only one) of them. If similarity(A, B) = 1, both primitives always co-occur in complex concepts and cannot be distinguished. Then they present some equations for realizing such a measure and to extend its applicability from primitives to complex concepts. According to the approach presented in [9], in an unfolding process, each compared concept is associated with finite set of signatures in terms of the primitive concepts and properties used in the description of them. Then, the influence of these signatures is evaluated by counting the number of their occurrences in each concept of the ontology, and weights of signatures are computed using a method similar to the inverse document frequency weight scheme from IR with the assumption that signatures appearing in a small number of concepts are more significant for the purpose of discriminating between concepts than those that are frequently referred to by many concepts. By representing concepts as signature vectors, distances (or dissimilarities) between concepts will then equal the distance between vectors in a high dimensional space.

It is clear that the facts about how many times two primitive concepts co-occur in the description of complex concepts or how many times a signature occurs in the descriptions of concepts in an ontology, do not directly say anything about the extent to which two given complex concepts overlap if those primitives or signatures have been used in the description of them. At least, such measurement approaches are not sufficiently straightforward and reliable in estimating the introduced ideal measure as precise as possible. Obviously, such measures rely on the statistics generated from ontologies with regard to the occurrences of some concepts or properties in the description of the others, and therefore their accuracy highly depends on the general design of those ontologies rather than the logical description of the two compared concepts themselves, while measures which directly rely on the logical description of concepts, should be theoretically intuitively much more reliable if they try to estimate the introduced ideal measure.

As a result of our research, we propose that every similarity/dissimilarity measure should specify the semantics of similarity (values) and its properties in order to make possible comparing those measures with other ones, otherwise comparison with other approaches is not possible or is not even meaningful at all. In this paper, we exactly specified the semantics of the target similarity (values) by introducing an ideal similarity metric, but most of the related works have not exactly defined what they try to measure and in which application area their proposed measure will be (more) helpful and why? At least making clear the relation between most approaches to semantic similarity measurement and the ideal measure introduced in this paper does not seem to be an easy task [11]. The authors in [27], present a DL-based approach for semantic matching of web services. It seems their proposed DL-based measure, represented as a pseudo code, tries to estimate our proposed ideal similarity metric although they have not exactly specified the semantics of similarity (values) in their research paper. Some related works may try to measure inter-instance similarity, but as it is conceived of the name inter-instance similarity, the notion behind their measure is completely different from the one of ours. The notion of computing inter-instance similarity is completely different from the notion of computing the extent to which two concepts share instances in common. The two measures might yield similar results in some situations, but there is no guarantee to always yield near results [12].

Most of DL-based measures such as ones presented in [2], [10], [16] [1], [17], [21], [23], [28] and [37] are limited to only infer and use simple subsumption relations between concepts and roles in ontologies. Such approaches are mostly based on the notion of degrees of match organized in a discrete scale. These degrees of match are often named as follows: Exact, Plug in, Subsumes, and Disjoint. Some of these measures such as ones presented in [2] and [10], use a network-distance based model besides subsumption reasoning to enhance their similarity calculation model. Anyway, these measures are not still able to compute the extent to which two concepts share instances in common and therefore they

[1] They may be hybrid approaches that combine the logic-based approach with syntactic ones. In such cases, we only consider the logic-based parts of those measures.

are not able to estimate the introduced ideal measure if none of the two concepts subsumes the other but the two concepts overlap. The authors in [17] add Intersection as a degree of match to the aforementioned ones in order to distinguish it from Disjoint. This degree of match is determined for two concepts if the intersection of them is satisfiable. Anyway, this measure is not still able to compute the extent to which two concepts overlap and therefore to precisely estimate the introduced ideal measure.

8. Conclusion

In our research, we primarily wanted to extend the previously proposed theories for logic based matching of web services that were based on simple subsumption reasoning. To explicitly say, we proposed an ideal semantic similarity metric in Section 4 to extend simple subsumption based similarity metrics in order to include the states in which two compared concepts overlap but none of them subsumes the other. The introduced ideal similarity metric is more perfect than simple subsumption based ones to be used in the field of web service retrieval since it can increase the recall-based performance of web service matchmakers which utilize this similarity metric for web services matching and composition.

Our research work presented in this paper can be considered as the first step in defining a sophisticated computable applicable logic based similarity measure to be used in the field of web service retrieval. Because in order to achieve such a similarity measure, we firstly need to ideally define what should be measured by specifying the semantics of similarity (values) resulted from applying the target similarity measure. Hence in this paper, we defined our ideal semantic similarity metric which fulfills the requirements for similarity measurement in the field of web service retrieval. But since it is not generally actually computable, it has to be estimated based on DL based descriptions of concepts in ontologies. So, defining a suitable computable applicable logic based similarity measure which tries to estimate our proposed ideal similarity metric is left to future works. If such a logic based similarity measure can handle the expressivity of DL based ontology languages to a large extent, then it opens the way for ontology engineers to build sophisticated ontologies with high DL Expressivity Usage and high Proportion of Overlapped Concepts to be used in sophisticated service oriented applications. Because such sophisticated ontologies need to be handled based on the introduced ideal similarity metric to be completely useful in service oriented applications. It is also needed to find a complete solution to the problem of matching the semantic descriptions of web services in which the target semantic similarity measure should be used as a fundamental operation.

References

1. Alvarez, M. and S. Lim (2007) A graph modeling of semantic similarity between words, in Proceedings of the Conference on Semantic Computing, 355–362.
2. Bener A.B., V. Ozadali and E.S. Ilhan (2009) Semantic matchmaker with precondition and effect matching using SWRL, Expert Systems with Applications: An International Journal, 36 n.5, 9371-9377.
3. Borgida, A., T.J. Walsh and H. Hirsh (2005) Towards measuring similarity in description logics, in Proceedings of the 2005 International Workshop on Description Logics (DL2005), Edinburgh, Scotland.
4. Dong H., F. K. Hussain, and E. Chang (2011) A service concept recommendation system for enhancing the dependability of semantic service matchmakers in the service ecosystem environment, in Journal of Network and Computer Applications, 34, 619–631.
5. García-Sánchez F., R. Valencia-García, R. Martínez-Béjar and J.T. Fernández-Breis (2009) An ontology, intelligent agent-based framework for the provision of semantic web services, Expert Systems with Applications: An International Journal, 36 n.2, 3167–3187.
6. Ge J. and Y. Qiu (2008) Concept Similarity Matching Based on Semantic Distance, in Proceedings of the fourth International Conference on Semantics, Knowledge and Grid, 380-383.
7. Guisheng Y. and S. Qiuyan (2008) Research on Ontology-Based Measuring Semantic Similarity, in Proceedings of International Conference on Internet Computing in Science and Engineering, 250-253.
8. Günay A. and P. Yolum (2010) Service matchmaking revisited: An approach based on model checking, in Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 8, 292–309.
9. Hu, B., Y. Kalfoglou, H. Alani, D. Dupplaw, P. Lewis and N. Shadbolt (2006) Semantic metrics, in Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management, 166-181.
10. Ilhan, E.S., G.B. Akkus and A. Bener (2007) SAM: Semantic advanced matchmaker, in International conference on software engineering and knowledge engineering, 698–703.
11. Janowicz, K. (2006) Sim-dl: Towards a semantic similarity measurement theory for the description logic ALCNR in geographic information retrieval, in Meersman, R., Z. Tari and P. Herrero (eds.) SeBGIS 2006, OTM Workshops (2006) LNCS, Heidelberg, Springer, 4278, 1681-1692.
12. Janowicz, K. and M. Wilkes (2009) SIM-DLA: A Novel Semantic Similarity Measure for Description Logics Reducing Inter-concept to Inter-instance Similarity, in Aroyo, L., P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvoenen, R. Mizoguchi, E. Oren, M. Sabou and E.P.B. Simperl, (eds.) 6th Annual European Semantic Web Conference (ESWC2009), Volume 5554 of Lecture Notes in Computer Science, Springer, 353-367.

13. Jeong B., H. Cho and C. Lee (2009) On the functional quality of service (FQoS) to discover and compose interoperable web services, Expert Systems with Applications: An International Journal, 36 n.3, 5411-5418.
14. Jia K., J. Fu, X. Jiang and J. Mao (2008) Semantic Similarity Computation Based on HowNet2008, In International Conference on Natural Language Processing and Knowledge Engineering (IEEE), 1-5.
15. Kim S.K. and H.J. Choi (2008) Decision of Semantic Similarity Using Description Logic and Vector Weight between Concepts, in Proceedings of the fourth International Conference on Networked Computing and Advanced Information Management, 345-350.
16. Klusch M., B. Fries and K. Sycara (2009) OWLS-MX: A hybrid semantic web service matchmaker for OWL-S services, Journal of Web Semantics: Sciences, Services, and Agents on the World Wide Web, 7 n.2, 121-133.
17. Li L. and I. Horrocks (2003) A software framework for matchmaking based on semantic web technology, in Proceedings of the twelfth international world wide web conference, 331-339.
18. Liu M., W. Shen, Q. Hao and J. Yan (2009) An weighted ontology-based semantic similarity algorithm for web service, Expert Systems with Applications: An International Journal, 36 n.10, 12480-12490.
19. Liu X. Y., Y. M. Zhou and R. S. Zheng (2007) Measuring Semantic Similarity in Wordnet, in Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, 3431-3435.
20. Ljubešić N., D. Boras, N. Bakarić and J. Njavro (2008) Comparing Measures of Semantic Similarity, in Proceedings of the ITI 30th International Conference on Information Technology Interfaces, 675-682.
21. Mokhtar S.B., D. Preuveneers, N. Georgantas, V. Issarny and Y. Berbers (2007) EASY: Efficient semAntic Service discoverY in pervasive computing environments with QoS and context support, Journal of Systems and Software, 81 n.5, 785-808.
22. OWL 2 Web Ontology Language Overview, at http://www.w3.org/TR/owl2-overview/. (Accessed 12 July 2011)
23. Paolucci M., T. Kawamura, T.R. Payne and K.P. Sycara (2002) Semantic matching of web services capabilities, in Proceedings of the first international semantic web conference, 333-347.
24. Qiuyan S. and Y. Guisheng (2008) Measuring Semantic Similarity in Ontology and Its Application in Information Retrieval, Congress on Image and Signal Processing, 525-529.
25. Sayed A.E., H. Hacid and D.A. Zighed (2007) A New Context-Aware Measure for Semantic Distance Using a Taxonomy and a Text Corpus, in IEEE International Conference on Information Reuse and Integration, 279-284.
26. Sbodioa M. L., D. Martin, and C. Moulin (2010) Discovering Semantic Web services using SPARQL and intelligent agents, in Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 8, 310–328.
27. Shu G., O.F. Rana, N.J. Avis, C. Dingfang (2007) Ontology-based semantic matchmaking approach, Advances in Engineering Software: An International Journal, 38 n.1, 59-67.
28. Talantikite H.N., D. Aissani and N. Boudjlida (2009) Semantic annotations for web services discovery and composition, Computer Standards & Interfaces: An International Journal, 31 n.6, 1108-1117.

29. Tian Q., L. Lei and L. Pin (2007) Web Service Discovery with UDDI Based on Semantic Similarity of Service Properties, in Proceedings of the third International Conference on Semantics, Knowledge and Grid, 454-457.
30. Tocha E., I. Reinhartz-Bergerb, and D. Doric (2011) Humans, semantic services and similarity: A user study of semantic Web services matching and composition, in Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 9, 16–28.
31. Wang G., D. Xu, Y. Qi and D. Hou (2008) A Semantic Match Algorithm for Web Services Based on Improved Semantic Distance, in Proceedings of the forth International Conference on Next Generation Web Services Practices, 101-106.
32. Wang X. and J. Zhou (2009) An Improvement on the Model of Ontology-Based Semantic Similarity Computation, First International Workshop on Database Technology and Applications, 509-512.
33. Wei C. (2008) Research on the matching algorithm of semantic web service In pervasive environments, in Proceedings of the International Seminar on Future BioMedical Information Engineering, 395-398.
34. Wei D., T. Wang, J. Wang, and A. Bernstein (2011) SAWSDL-iMatcher: A customizable and effective Semantic Web Service Matchmaker Web Semantics: Science, Services and Agents on the World Wide Web, 9, 402–417.
35. Xiao M., L. Zhong and Q. Xiong (2009) Semantic Similarity between Concepts Based on OWL Ontologies, in Proceedings of the Second International Workshop on Knowledge Discovery and Data Mining, 749-752.
36. Xu X., J. Huang, J. Wan and C. Jiang (2008) A Method for Measuring Semantic Similarity of Concepts in the Same Ontology, in Proceedings of the International Multi-symposiums on Computer and Computational Sciences, 207-213.
37. Yang S.J.H., J. Zhang and I.Y.L. Chen (2008) A JESS-enabled context elicitation system for providing context-aware Web services, Expert Systems with Applications: An International Journal, 34 n.4, 2254-2266.
38. You F., Q. Hu, Y. Yao, G. Xu and M. Fang (2008) Study on Web Service Matching and Composition Based on Ontology, in Proceedings of the WRI World Congress on Computer Science and Information Engineering, 542-546.
39. Zhang D. (2008) A Method to Compare Concepts between Ontologies, in Proceedings of the International Conference on Advanced Computer Theory and Engineering, 591-595.