Southeast-Asian J. of Sciences: Vol 1, No 1(2012), pp. 32-45

A MATHEMATICAL MODEL FOR SEMANTIC SIMILARITY MEASURES

Dinh Que Tran*, Manh Hung Nguyen†

Department of Information Technology
Post and Telecommunication Institute of Technology (PTIT)
Hanoi, Vietnam
*e-mail: * tdque@yahoo.com, † nmhufng@yahoo.com*

Abstract

Semantic similarity between words, concepts or sets of concepts has been a fundamental theme and widely studied in various areas including natural language processing, document semantic comparison, artificial intelligence, semantic web, semantic web service, and semantic search engines. Several similarity measures have been proposed but they are usually tied to special application domains or information representation of various application domains.

In this paper, we present a mathematical model for distance-based semantic similarity estimation in domains that are represented with ontology - an explicit specification of conceptualization of such domains. Based on this model, we construct algorithms to calculate the semantic similarity between two concepts and one between two sets of concepts. The significance of the proposed mathematical model is that it offers a generalization that enables to maintain flexibility and thus supports various computational measures.

1. Introduction

Determining the semantic relatedness between words, concepts or sets of concepts refers to computing a measure of similarity between those ones. Such computation has played an important role in distributed systems to enable to

Key words: mathematical model, semantic similarity, semantic matching, ontology, semantic web, semantic web service.
2000 AMS Mathematics Subject Classification: Applied Mathematics

be understandable among interacting autonomous components. Semantic similarity, which is the form of semantic relatedness, has become one of important research areas in computation. It has been widely used in applications including natural language processing, document comparison, artificial intelligence, semantic web, semantic web service and semantic search engines. Several similarity measures have been proposed ([1] [2] [3] [4] [5] [6] [7] [8] [9]) such as ones based on information content, cosine coefficient, Dice coefficient, measure based on distance and so on. Such measures are usually tied to some special application domain or information representation of application domains. Especially, in Semantic Web or Semantic Web Service ([1] [6]), description of an object is represented in the standard Ontology of Web Language (OWL) or OWL-based Web Service Ontology (OWL-S), which is the type of knowledge representation with semantic network in Artificial Intelligence. Methods addressing similarity in these domains is based on the structure OWL or OWL-S. Although several techniques of computing similarity in various domains has been proposed, a general computational model of semantic relatedness remains a challenging task.

In this paper, we introduce a mathematical model for distance-based semantic similarity estimation in domains that are represented with various ontologies. First of all, we investigate a mathematical representation of semantic distance between concepts in an ontology. Then, we examine a mathematical model for similarity of two concepts as well as similarity between a concept and a set of concepts. Based on this model, we develop algorithms to calculate the semantic similarity between two concepts and one between two sets of concepts. The significance of the proposed mathematical model is that it offers a generalization that enables to maintain flexibility and thus supports various computational measures. The remainder of this paper is organized as follows. Section 2 is devoted to our mathematical model for semantic similarity measure between two concepts. Section 3 presents a mathematical model for semantic similarity measure between two sets of concepts. Section 4 is the discussion of our model and then compares it with some related works as well. Conclusion is given in Section 5.

2. Semantic Similarity between Two Concepts

2.1. Semantic Similarity between Concepts in an Ontology

Definition 1. *An ontology is a 2-tuple* $\mathcal{G} =< \mathcal{C}, \mathcal{V} >$*, in which* $\mathcal{C}$ *is a set of nodes corresponding to concepts defined in the ontology and* $\mathcal{V}$ *is a set of arcs representing relationships of couples of nodes in* $\mathcal{C}$*.*

In this paper, rather than considering the properties of nodes, we focus on relationship between concepts. A relationship in $\mathcal{V}$ is defined as follows: If

$x, y \in \mathcal{C}$ and $< x, y > \in \mathcal{V}$, then x is called the parent of y, and y is the child of x. An ontology is of the tree form, in which each node has a unique parent, but may have several child nodes.

Definition 2. *Let $\mathcal{C}$ be a set of concepts. A similarity measure $sim : \mathcal{C} \times \mathcal{C} \rightarrow [0, 1]$ is a function from a pair of concepts to a real number between zero and one such that:*

(i) $\forall x \in \mathcal{C}\ sim(x, x) = 1$;

(ii) $\forall x, y \in \mathcal{C}\ sim(x, y) = sim(y, x)$.

Definition 3. *The path length $L(c_1, c_2)$ between concepts c_1 and c_2 in an ontology is the length of the shortest path from node c_1 to node c_2 on the ontology.*

In order to compare the semantic similarity between concepts on ontology, the following assumptions are accepted:

Assumption 1. *Let c_1 and c_2 be two concepts defined in an ontology whose root node is root, then:*

(i) If two concepts are identical, their path length is 0: $L(c_i, c_i) = 0$ with $\forall i$. Then their semantic similar is maximal and then could be normalised as 1;

(ii) The path length between any concept to the general root of the ontology is maximal $L(c_i, root) = \infty$ with $\forall i$;

(iii) If two concepts are independent - they have no common root concept (their common root concept is the general root of the ontology), then their similar is minimal and could be normalised as 0;

(iv) The longer the path from each of them to the other is, the less semantic similar they have;

(v) If two concepts have a common root concept, then their semantic relation is defined as follows:

- *If this one is parent of the another one, then their similar is bigger than if the common root concept is not in two them;*
- *The longer the path from each of them to the common root concept is, the less semantic similar they are.*

Let c_0 be the nearest common ancestor concept of two concepts c_1 and c_2, we have $L(c_1, c_2) = L(c_1, c_0) + L(c_0, c_2)$. Based on the above assumption, we can define a pre-similar function as follows:

Definition 4. *A function $f : \Re^+ \times \Re^+ \to [0,1]$ is pre-similar, denoted pre-sim, iff it satisfies the following conditions:*

(i) $f(0,0) = 1$;

(ii) $f(\infty, l) = f(l, \infty) = 0$;

(iii) $f(l_1, l_2) = f(l_2, l_1)$;

(iv) $f(l_1, l_2) \geqslant f(l_3, l_4)$ *if* $l_1 + l_2 \leqslant l_3 + l_4$;

(v) $f(l_1, l_0) \geqslant f(l_2, l_0)$ *if* $l_1 \leqslant l_2$;

(vi) $f(l_0, l_1) \geqslant f(l_0, l_2)$ *if* $l_1 \leqslant l_2$.

It is easy to prove the following propositions:

Proposition 1. *The functions f, g determined by the following formulas*

(i) $f(x,y) = \frac{1}{(x+y+1)^n} \quad n = 1, 2...$

(ii) $g(x,y) = \frac{1}{e^{x+y}}$

are pre-sim functions.

Proposition 2. *Given a pre-sim function $f_{ont} : \Re^+ \times \Re^+ \to [0,1]$. The function $s_{ont} : \mathcal{C} \times \mathcal{C} \to [0,1]$ between concepts c_1 and c_2 with the nearest common ancestor c_0 on an ontology determined by the formula*

$$s_{ont}(c_1, c_2) = f_{ont}(L(c_1, c_0), L(c_0, c_2))$$

is a similar measure.

Estimating the ontology-based semantic similarity is presented in Algorithm 1. First of all, searching the common parent node of the two given concepts (Step 1), then calculating the path length between each one to their common parent node (Steps 2-3). Applying the mapping f_{ont}, which is defined in Definition 4, to calculate the similarity of two given concepts (Step 4).

2.2. Syntax Similarity between Words with the Same Core

In reality, there are several words with the same original core word, but not all of them are always included in an ontology. In order to measure the semantic similarity between these words (called the core semantic similarity), we include an additional concept.

Definition 5. *The syntax distance between a word w_1 and its original core word w_0, denoted as $d(w_1, w_0)$, is the total number of characters that may be added (or deleted) from the word w_1 to become the original core word w_0.*

Algorithm 1 Ontology-based semantic similarity

Input: two concepts c_1 andc_2 on an ontology
Output: the ontology-based semantic similarity between c_1 and c_2: $OntS(c_1, c_2)$

1: $c_0 \leftarrow$ the common parent node of c_1 and c_2
2: $l_1 \leftarrow L(c_1, c_0)$
3: $l_2 \leftarrow L(c_2, c_0)$
4: $OntS(c_1, c_2) \leftarrow f_{ont}(l_1, l_2)$
5: **return** $OntS(c_1, c_2)$

As a consequence, the syntax distance between two words w_1 and w_2, which have the same original core word $w_0 \notin \{w_1, w_2\}$, is the total distance from each of them to the common core word: $d(w_1, w_2) = d(w_1, w_0) + d(w_2, w_0)$. We assume that:

Assumption 2. *Let w_i and w_j be two words, then:*

(i) If two words are identical, their distance is 0. It means that $d(w_i, w_i) = 0$ with $\forall i$. Then their syntax similarity is maximal, which could be normalised as 1;

(ii) If two words have no any original core word, their syntax distance is maximal $d(w_i, w_j) = \infty$. If w_i is totally different from w_j, then their syntax similarity is minimal, which could be normalized as 0;

(iii) The longer the syntax distance from each of them to the original core word is, the less syntax similarity they have.

Let w_0 be the original core word of two words w_1 and w_2, we define a syntax similarity between w_1 and w_2 as follows:

Proposition 3. *Let $f_{syn} : \Re^+ \times \Re^+ \rightarrow [0, 1]$ be a pre-similar function. The syntax similarity between words w_1 and w_2 determined by the formula*

$$s_{syn}(w_1, w_2) = f_{syn}(d(w_1, w_0), d(w_2, w_0))$$

is a similar measure.

Estimating the syntax similarity is presented in Algorithm 2. First of all, finding the original core word of the two given words (Step 1), then calculating the distance between each one to their original core word (Steps 2-3). Lastly, applying the mapping f_{syn}, defined in Proposition 3, to calculate the similarity of two given words (Step 4).

Algorithm 2 Syntax similarity

Input: two words w_1 and w_2 having an original core word
Output: the syntax similarity between w_1 and w_2: $SynS(w_1, w_2)$

1: $w_0 \leftarrow$ the original core word of (w_1, w_2)
2: $d_1 \leftarrow d(w_1, w_0)$
3: $d_2 \leftarrow d(w_2, w_0)$
4: $SynS(w_1, w_2) \leftarrow f_{syn}(d_1, d_2)$
5: **return** $SynS(w_1, w_2)$

2.3. Transitive Semantic Similarity

Let c_1, c_2 and c_3 be concepts, in which only c_2 and c_3 belong to the same ontology and c_1 and c_2 shares the same core word. Then the relatedness relation between c_1 and c_3 is called a *transitive semantic relation*. We assume that:

Assumption 3. *Let c_1, c_2 and c_3 be three concepts or words in which only c_2 and c_3 are defined in an ontology, c_1 has the same core original syntax with c_2, then:*

(i) While there is no definition of c_1 in ontology, this means that there is no semantic relation between c_1 and c_3 on the ontology, so the transitive semantic matching of c_1 and c_3 must be not bigger than the semantic matching between c_2 and c_3;

(ii) The higher the core word similarity between c_1 and c_2 is, the higher the transitive semantic similarity between c_1 and c_3 via c_2 is;

(iii) The higher the semantic similarity between c_2 and c_3 is, the higher the transitive semantic similarity between c_1 and c_3 via c_2 is;

Definition 6. *A function $f_{tran} : \Re^+ \times \Re^+ \to [0, 1]$ is a transitive similar function, denoted tran-sim, iff it satisfies the following conditions:*

(i) $0 \leqslant f_{tran}(u, v) \leqslant v$;

(ii) $f_{tran}(u_1, v) \leqslant f_{tran}(u_2, v)$ *if* $u_1 \leqslant u_2$;

(iii) $f_{tran}(u, v_1) \leqslant f_{tran}(u, v_2)$ *if* $v_1 \leqslant v_2$.

It is easy to prove the following proposition:

Proposition 4. *The following functions are tran-sim functions:*

(i) $f(x, y) = y$;

(ii) $g(x, y) = x * y$;

(iii) $h(x,y) = \frac{a*min(x,y)+b*y}{a+b}$ $a, b \in N^*$;

And the transitive semantic distance is defined as follows:

Definition 7. *Let c_1, c_2 and c_3 be concepts, in which only c_2 and c_3 belong to the same ontology and c_1 and c_2 shares the same core word. Suppose that $f_{tran} : \Re \times \Re \to [0,1]$ is a tran-sim function, $s_{syn}(c_1, c_2)$ is the syntax similarity on the same core word between c_1 and c_2, $s_{ont}(c_2, c_3)$ is the semantic similarity on ontology between c_2 and c_3. The transitive semantic similarity between concepts c_1 and c_3 via concept c_2 is determined by the following formula:*

$$s_{tran}(c_1, c_2, c_3) = f_{tran}(s_{syn}(c_1, c_2), s_{ont}(c_2, c_3))$$

It is easy to prove the following proposition.

Proposition 5. *Suppose that c_1 has many concepts in core word relations $C = \{c'_1, c'_2, ..., c'_n\}$ and all $c'_i \in C$ have semantic similarity on an ontology with c_3. The transitive semantic similarity between c_1 and c_3 defined by the following formula:*

$$s_{tran}(c_1, c_3) = \ Max_{c'_i \in C}\{f_{tran}(s_{syn}(c_1, c'_i), s_{ont}(c'_i, c_3))\} \quad (1)$$

is a similar measure.

The algorithm of estimating the transitive semantic similarity is presented in Algorithm 3. For all concepts c_i having the same original core word with w and it is also defined in the same ontology with c, we calculate the syntax similarity between w and c_i (Step 2), then calculate the ontology-based semantic similarity between c_i and c (Steps 3), and then apply the mapping f_{tran}, defined in Definition 7, to calculate the transitive semantic similarity of w and c via c_i (Step 4). The final transitive semantic similarity between w and c is determined as the maximal value in all the intermediate values via c_i (Step 6)

2.4. General Semantic Similarity between Two Concepts

Let c_1 and c_2 be two words or concepts. In order to measure their semantic similarity in general, we consider the following cases:

- If c_1 and c_2 are both in the same ontology, then their general semantic similarity is their ontology-based semantic similarity defined in Definition 4;
- If either c_1 or c_2 is in an ontology, other is not, their general semantic similarity is their transitive semantic similarity defined in Definition 7;
- If neither c_1 nor c_2 is in an ontology, we consider as they have not any semantic relation;

The algorithm to estimate the general semantic similarity between two words or concepts is presented in Algorithm 4.

Algorithm 3 Transitive semantic similarity

Input: a word w and a concept c in an ontology
Output: the transitive semantic similarity between w and c: $TranS(w, c)$

1: **for all** concept c_i in the same ontology with c, and c_i has an original core word with w **do**
2: $s_{syn} \leftarrow SynS(w, c_i)$
3: $s_{ont} \leftarrow OntS(c_i, c)$
4: $s(w, c, c_i) \leftarrow f_{tran}(s_{syn}, s_{ont})$
5: **end for**
6: $TranS(w, c) \leftarrow Max\{s(w, c, c_i)\}\ \forall c_i$
7: **return** $TranS(w, c)$

Algorithm 4 General semantic similarity

Input: two words or concepts c_1 and c_2
Output: the general semantic similarity between c_1 and c_2: $GeneralS(c_1, c_2)$

1: **if** c_1 and c_2 are in the same ontology **then**
2: $GeneralS(c_1, c_2) \leftarrow OntS(c_1, c_2)$
3: **else**
4: **if** c_1 is in an ontology **then**
5: $GeneralS(c_1, c_2) \leftarrow TranS(c_2, c_1)$
6: **else**
7: **if** c_2 is in an ontology **then**
8: $GeneralS(c_1, c_2) \leftarrow TranS(c_1, c_2)$
9: **else**
10: $GeneralS(c_1, c_2) \leftarrow 0$
11: **end if**
12: **end if**
13: **end if**
14: **return** $GeneralS(c_1, c_2)$

3. Semantic Similarity between Two Sets of Concepts

3.1. Semantic Similarity between a Concept and a Set of Concepts

Let c and $C = \{c_1, c_2, ..., c_n\}$ be a concept and a set of concepts, respectively. In order to measure the semantic similarity between the concept c and set C, we assume that:

Assumption 4. *Let c and $C = \{c_1, c_2, ..., c_n\}$ be a concept and the set of concepts, respectively. Then:*

(i) If C has only one element which is identical with c, i.e., $C = \{c\}$, then the similarity between c and C is maximal, which could be normalised as 1;

(ii) The semantic similarity between c and C must not lower than the minimal semantic similarity between c and each concept $c_i \in C$, and it also must not higher than the maximal semantic similarity between c and each concept $c_i \in C$;

(iii) The higher the semantic similarity between c and each concept $c_i \in C$ is, the higher the semantic similarity between c and C is;

Definition 8. *$f : \Re^n \to [0, 1]$ is a semantic similar function between a concept and a set of concepts , denoted single-sim, iff it satisfies the following conditions:*

(i) $f(1^n) = 1$;

(ii) $min(x_1, x_2, ..., x_n) \leqslant f(x_1, x_2, ..., x_n) \leqslant max(x_1, x_2, ..., x_n)$

(iii) $f(x_1, .., x_i, ..., x_n) \leqslant f(x_1, ..., x'_i, ..., x_n)$ *if* $x_i \leqslant x'_i$ $i = 1, ..., n$

It is easy to prove the following proposition.

Proposition 6. *The following functions are single-sim functions:*

(i) $f(x_1, x_2, ..., x_n) = min(x_1, x_2, ..., x_n)$

(ii) $g(x_1, x_2, ..., x_n) = max(x_1, x_2, ..., x_n)$

(iii) $h(x_1, x_2, ..., x_n) = average(x_1, x_2, ..., x_n)$

(iv) $t(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} w_i * x_i$, *where* $w_i \in [0, 1]$ *is the weight of* x_i

Let $s(c, c_i)$ be the general semantic similarity between c and $c_i \in C$, defined in Algorithm 4. Then the semantic similarity between single concept c and a set of concepts $C = \{c_1, c_2, ..., c_n\}$ is defined as follows:

Definition 9. *Given a concept c and a set of concepts $C = \{c_1, c_2, ..., c_n\}$ and a single-sim function $f_{single} : \Re^n \to [0, 1]$. The semantic similarity between a concept c and a set of concepts $C = \{c_1, c_2, ..., c_n\}$ is determined by the formula:*

$$S_{single-sim}(c, C) = f_{single}(s(c, c_1), s(c, c_2), ..., s(c, c_n)).$$

The algorithm of estimating the semantic similarity between a concept c and a set of concepts C is presented in Algorithm 5. For all concepts $c_i \in C$, we calculate the general semantic between c and c_i (Step 2), then apply the mapping f_{single}, defined in Definition 9, to calculate the semantic similarity c and set C (Step 4)

Algorithm 5 Semantic similarity between a single concept and a set of concepts

Input: a word or concepts c and a set of concepts $C = \{c_1, c_2, ..., c_n\}$
Output: the semantic similarity between c and set C: $SingleS(c, C)$

1: **for all** concept c_i in the C **do**
2: $s(c, c_i) \leftarrow GeneralS(c, c_i)$
3: **end for**
4: $SingleS(c, C) \leftarrow f_{single}(s(c, c_1), s(c, c_2), ..., s(c, c_n))$
5: **return** $SingleS(c, C)$

3.2. Semantic Similarity between Two Sets of Concepts

Suppose that $C = \{c_1, c_2, ..., c_n\}$ and $C' = \{c'_1, c'_2, ..., c'_m\}$ are two sets of concepts and $C^* = C \cap C' = \{c^*_1, c^*_2, ..., c^*_k\}$, $0 \leqslant k \leqslant Min(n, m)$ is the intersection set of C and C'. In order to measure the semantic similarity between two sets C and C', we make use of the following assumptions:

Assumption 5. *Assumptions of two set similarity*

(i) If C and C' are identical, then their semantic similarity is maximal, which could be normalised as 1;

(ii) The semantic similarity between C and C' is equal to those between C' and C. It means that the relation is symmetric;

(iii) The more the size of C^* is big, the more the semantic similarity between C and C' is high;

(iv) The higher the semantic similarity between each element $c_i \in C, c_i \notin C^*$ and set C' is, the higher the semantic similarity between C and C' is;

(v) The higher the semantic similarity between each element $c'_i \in C', c'_i \notin C^*$ and set C is, the higher the semantic similarity between C and C' is;

Definition 10. *$f_{set} : \Re^n \rightarrow [0, 1]$ is a semantic similar function between two sets of concepts iff it satisfies the following conditions:*

(i) $f_{set}(k) = 1$ for all k, $0 \leqslant k \leqslant min(n, m)$

(ii) $f_{set}(k_1, x_1, .., x_i, ..., x_n) \leqslant f_{set}(k_2, x_1, .., x'_i, ..., x_n)$ if $k_1 \leqslant k_2$

(iii) $f_{set}(k, x_1, .., x_i, ..., x_n) \leqslant f_{set}(k, x_1, .., x'_i, ..., x_n)$ if $x_i \leqslant x'_i$ for all $i = 1, .., n$

Proposition 7. *The following functions are set-sim functions:*

(i) $f_{set}(k, x_1, x_2, ..., x_n) = \frac{k+min(x_1,x_2,...,x_n)}{k+1}$

(ii) $f_{set}(k, x_1, x_2, ..., x_n) = \frac{k+max(x_1,x_2,...,x_n)}{k+1}$

(iii) $f_{set}(k, x_1, x_2, ..., x_n) = \frac{k+average(x_1,x_2,...,x_n)}{k+1}$

(iv) $f_{set}(k, x_1, x_2, ..., x_n) = \frac{k+\sum_{i=1}^{n} w_i * x_i}{k+1}$, *where* w_i *is the weight of* x_i

Let $S_{single-sim}(c_i, C')$ be the semantic similarity between each single concept $c_i \in C$ and set C', defined in Definition 9. Then the semantic similarity between two sets of concepts $C = \{c_1, c_2, ..., c_n\}$ and $C' = \{c'_1, c'_2, ..., c'_m\}$ is defined as follows:

Definition 11. *Suppose* $C = \{c_1, c_2, ..., c_n\}$ *and* $C' = \{c'_1, c'_2, ..., c'_m\}$ *are two sets of concepts,* $C^* = C \cap C' = \{c^*_1, c^*_2, ..., c^*_k\}$, $0 \leqslant k \leqslant Min(n, m)$ *is the intersection set of* C *and* C' *and* $f_{set} : \Re^{n+1} \rightarrow [0, 1]$ *is a set-sim function. The semantic similarity between two sets* C *and* C' *is determined by the formula:*

$$S_{set}(C, C') = f_{set}(k, S_{single-sim}(c_1, C'), S_{single-sim}(c_2, C'), ..., S_{single-sim}(c_n, C'))$$

The algorithm of estimating the semantic similarity between two sets of concepts C and C' is presented in Algorithm 6. First, constructing the intersection set C^* of two given sets (Step 1) and calculating its size (Step 2), and then for each element c_i of C, calculating the semantic similarity between concept c_i and set C' by applying the formulas in Definition 9 (Step 4). Lastly, applying the mapping f_{set}, defined in Definition 11 to calculate the semantic similarity of two given sets (Step 6).

Algorithm 6 Semantic similarity between two sets of concepts

Input: two sets of concepts $C = \{c_1, c_2, ..., c_n\}$ and $C' = \{c'_1, c'_2, ..., c'_m\}$
Output: the semantic similarity between C and set C': $SetS(C, C')$

1: $C^* \leftarrow C \cap C'$
2: $k \leftarrow \|C^*\|$
3: **for all** concept c_i in the C **do**
4: $\quad S(c_i, C') \leftarrow SingleS(c_i, C')$
5: **end for**
6: $SetS(C, C') \leftarrow f_{set}(k, S(c_1, C'), S(c_2, C'), ..., S(c_n, C'))$
7: **return** $SetS(C, C')$

4. Discussion & Related works

Paolucci et al. [5] proposed a method for estimating the semantic similarity between descriptions of advertised and requested Web services. This algorithm distinguishes four types of semantic matching based on DAML-S - a DAML-based language for service description: *exact* if required concept is equal advertised concept; *plug-in* if advertised concept subsumes required concept; *subsumes* if required concept subsumes advertised concept; and *fail* if there is no semantic relation between required and advertised concepts. It is clear that this method is compatible with our model: These four kinds of matching are completely included in our function to estimate the ontology-based semantic similarity which is defined in Definition 4.

Although this method is good in distinguishing the main kinds of matching, it is inconvenient in applying. First, it does not make clear in the case that two concepts are *plug-in* or *subsumes*; intuitively, if they are direct parent - child, then their similarity must be higher than in the case they are grandparent - grandchildren. Second, this method does not consider the case that two concepts have the same parent or grandparent; they must have a semantic relation, but in this method, they are in the kind of *fail*. Third, this method enables only to compare two concepts in an ontology, if one of them is not defined in the ontology, we cannot compare them. Fourth, this method does not enable to compare the semantic similarity of two set of concepts, which intuitively appears very frequent in the reality.

Ludwig et al. [4] distinguished three types of matching based on the work of Paolucci et al. [5] in the context of semantic Web service matching: *Precise match*, the service provides the requested functionality or more; *Partial match*, the service is capable of providing part of the requested functionality; *Mismatch*, the service is not capable of providing the requested functionality. And the match scores are 0 represents a *mismatch*, 1 represents a *precise match* and a value in-between represents a *partial match*. This method is clearly compatible with our model, but it is much more simple than ours because of the same reasons with method of Paolucci et al. [5].

Wang et al. [7] has a big improvement from the method of Paolucci et al. [5]. They also considered the distance between concepts on an ontology (called *semantic distance*), and the other kind of semantic relation between concepts that missed in method of Paolucci, such as the case that two concepts have the same parent or the same root. Their model is considered as a particular instance of ours. Operators given in their model are concrete, while ours is a general model with only the constraints on them. Their model is still limited in one ontology since it cannot measure the *semantic distance* between two concepts if one of them is not defined in the same ontology. Moreover, this model does not enable to measure the *semantic distance* between two sets of concepts yet.

Another approach is that of D. Lin [2] whose idea is to measure the similarity between any two objects based on information-theoretic approach. Their model enables to measure the similarity between two any objects: values, vectors, words, taxonomy objects (this is applied only on a taxonomy structure such as WordNet, not on ontology) and most interesting is to measure also similarity between ordinal values. This model is very closed and compatible with ours except that this model is strictly based on information-theoretic definition with probability principle, our model is flexible with any kind of operators as long as they satisfy the constraints defined in our general model.

5. Conclusions

In this paper, we present a mathematical model for estimating or calculating the semantic similarity at two levels. First, it enables to estimate the semantic similarity between two concepts which are either defined in an ontology, or only one of them is defined in an ontology. The estimation is based on their semantic relation on ontology, or their syntax relation or both of them. Second, it enables to estimate the semantic similarity between two sets of concepts, which is also based on the semantic similarity between the individual concepts of the two given sets.

Our model is considered as a generalization of the proposed similarity computational models. At each step of estimation, instead of applying a particular function, we generate them as some series of functions satisfying the constraints defined by the model. This makes our model more flexible in developing. It means that the developers could choose their own operators and functions from their special domain as long as they satisfy the constraints defined in our approach.

However, this model is currently limited at estimating the semantic similarity between two sets of concepts which have not any constraints on the order of elements. In the future work, we will consider of estimating the semantic similarity of two ordered sets of concepts and/or those between two sentences.

References

[1] J. Hau, W. Lee, J. Darlington, *A semantic similarity measure for semantic web services*, The Fourteenth International World Wide Web Conference (WWW 2005), 10-14.

[2] Delang Lin, *An information-theoretic definition of similarity*, In Proc. 15th International Conf. on Machine Learning,Morgan Kaufmann (1998), 296-304.

[3] Min Liu, Weiming Shen, Qi Hao, Junwei Yan, *An weighted ontology-based semantic similarity algorithm for web service*, Journal of Expert Systems with Applications, Vol. 36 (2009), 12480-12490.

[4] Simone A. Ludwig and S. M. S. Reyhani, *Semantic approach to service discovery in a grid environment*, Journal of Web Semantic, Vol. 4, No. 1, (2006), 1-13.

[5] Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, and Katia P. Sycara, *Semantic matching of web services capabilities*, In Proceedings of the First International Conference on The Semantic Web, ISWC '02, Springer-Verlag (2002), 333-347.

[6] James Z. Wang, Farha Ali and Pradip K. Srimani, *An efficient method to measure the semantic similarity of ontologies*, Journal of Pervasive Computing and Communications Vol. 6, No. 1 (2010), 88-103.

[7] Gongzhen Wang, Donghong Xu, Yong Qi, and DiHou, *A semantic match algorithm for web services based on improved semantic distance*, In Proceedings of 4th International Conference on Next Generation Web Services Practices, USA, (2008), 101-106.

[8] S. Wan and R. A. Angryk, *Measuring Semantic Similarity Using WordNet-based Context Vectors*, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Montreal, Canada (2007), 908-913.

[9] A. Zohali and D. Zamanifar, ıMatching model for semantic web services discovery, Journal of Theoretical and Applied Information Technology, Vol. 07 (2009), 139-144.