ISSN:
2734-9918

TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC SƯ PHẠM TP HỒ CHÍ MINH

Tập 19, Số 10 (2022): 1735-1748

Website: https://journal.hcmue.edu.vn

HO CHI MINH CITY UNIVERSITY OF EDUCATION
JOURNAL OF SCIENCE

Vol. 19, No. 10 (2022): 1735-1748

https://doi.org/10.54607/hcmue.js.19.10.3648(2022)

Research Article

ENHANCING OWL ONTOLOGIES MATCHING BASED ON SEMANTIC SIMILARITY MEASUREMENT

Pham Thi Thu Thuy

Nha Trang University, Vietnam

Corresponding author: Pham Thi Thu Thuy – Email: thuythuy@ntu.edu.vn

Received: October 18, 2022; Revised: October 26, 2022; Accepted: October 28, 2022

ABSTRACT

Recently, Web Ontology Language (OWL) has become a widely-used language for providing a source of precisely defined concepts. The number of OWL documents, increasing with the growth of the Semantic Web, leads to the heterogeneous problem. The same concepts may be defined differently, using different terms and positions in the documental structure. Therefore, identifying the element similarity in different ontologies becomes crucial for the success of web mining and information integration systems. In this paper, we propose a new semantic similarity measure for comparing elements in different OWL ontologies. This measure is designed to enable the extraction of information encoded in OWL element descriptions and to take into account the element relationships with its ancestors, brothers, and children. We evaluate the proposed metrics in the context of matching two OWL documents to determine the number of matches between them. The experimental results show better accuracy over other approaches.

Keywords: matching; measure; ontology; OWL; semantic similarity

1. Introduction

OWL is a powerful ontology language using RDF/XML syntax. OWL inherits the advantages of its predecessor, OWLS, and adds many elements to help overcome the limitations of OWLS. The main purpose of OWL is to provide standards for creating a platform for resource management for sharing and reusing data on the Web.

However, the increasing number of OWL ontologies leads to the heterogeneity problem. The same entities may be modeled differently using different terms or placed in different positions in the entity hierarchy. This heterogeneous problem causes a great challenge to integrating the OWL ontologies. Measuring the entity similarity between two OWL ontologies is the core of the success of the information integration.

Several approaches have been proposed to measure the term similarity between different ontologies. In general, they can be divided into three groups: structure, lexical, and hybrid.

Cite this article as: Pham Thi Thu Thuy (2022). Enhancing OWL ontologies matching based on semantic similarity measurement. *Ho Chi Minh City University of Education Journal of Science*, *19*(10), 1735-1748.

Structure-based measures (Resnik, 1999; Lin, 1998; Jiang & Conrath, 1997; Akbari & Fathian, 2010; Cheng et al., 2018; Jean-Mary et al., 2009) rely mainly on the Information Content of the terms to represent their semantic values. Resnik's (1999) method concentrates only on the MICA of the compared terms. Still, it ignores the locations of these terms in the graph, e.g., a term's distance from the root of the ontology and the semantic impact of other ancestor terms. A term's distance to the root of the ontology shows the specialization level of this term in human perception. If a term is far from the root in the ontology, researchers know more information about it, and the meaning of the term is more specific. On the other hand, if a term is closer to the root of the ontology, it means the term is a more general term, such as cellular process or metabolic process, which does not provide too many details about the related entities.

For lexical-based approaches (Zhao & Wang, 2018; Preeti & Sanjay, 2020; Mingxin, Xue & Rui, 2013; Stoilos, Stamou & Kollias, 2005; Sánchez et al., 2010; Fayez & Althobaiti, 2017), each concept node in an ontology has its own property set, which reflects the characteristics of the concept. The higher the degree of attribute coincidence of concepts, the more similar they are. The advantage of this approach is that it can solve the problem of semantic similarity across ontology. However, the disadvantage is that it is more suitable for processing large ontology with rich semantic knowledge and not suitable for small ontology.

The hybrid method (Nguyen & Conrad, 2015; Xu et al., 2020; Sun, Wei & Wang, 2021; Han et al., 2017) considers both the structure and the lexical similarity of terms at different ontological levels. The hybrid method considers more factors than the single method. Still, it mainly relies on expert experience and adopts the method of manual weight assignment to formulate the weight factors of each element.

Our method is similar to the hybrid approach, although our computation focuses on the similarity between concepts in different OWL. However, the important difference between these approaches and our approach is that the description, the name, and the data type similarity values are derived from our proposed measures without any user intervention.

The remainder of the paper is organized as follows. Section 2 describes our approach to measuring OWL similarity. The experiment evaluation is given in Section 3. Finally, Section 4 concludes the paper.

2. O2Sim Method

The framework of O2Sim includes the input, the O2Sim computation, and the output. The input is two OWL ontologies. The main component of this framework is the O2Sim computation, composed of the description and structure similarity measures. The outputs are the similarity values of concepts between OWL ontologies. The O2Sim framework is depicted in Figure 1.

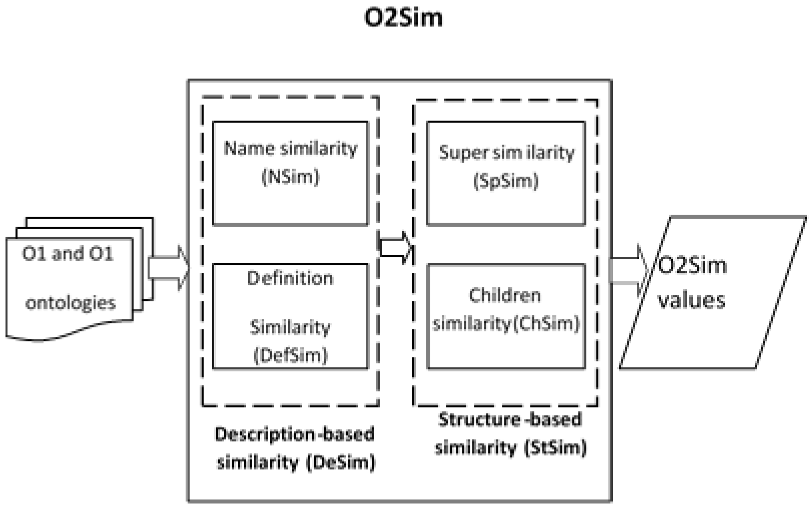

O2Sim
O1 and O1 ontologies
Name similarity (NSim)
Definition Similarity (DefSim)
Super sim ilarity (SpSim)
Children similarity(ChSim)
Description-based similarity (DeSim)
Structure -based similarity (StSim)
O2Sim values

Figure 1. *The framework of the O2Sim method*

The description similarity (DeSim) in Figure 1 comprises the similarity of the element name (NaSim.) and the definition similarity (DefSim). The structure similarity encompasses two individual measures: the ancestor element similarity (AnSim.) and the children element similarity (ChSim.). The final O2Sim similarity combines all the partial results using a weighted sum function.

The semantic similarity between concepts C1 and C2 is defined as the weighted sum of the description similarity (DeSim) and the structure similarity (StSim):

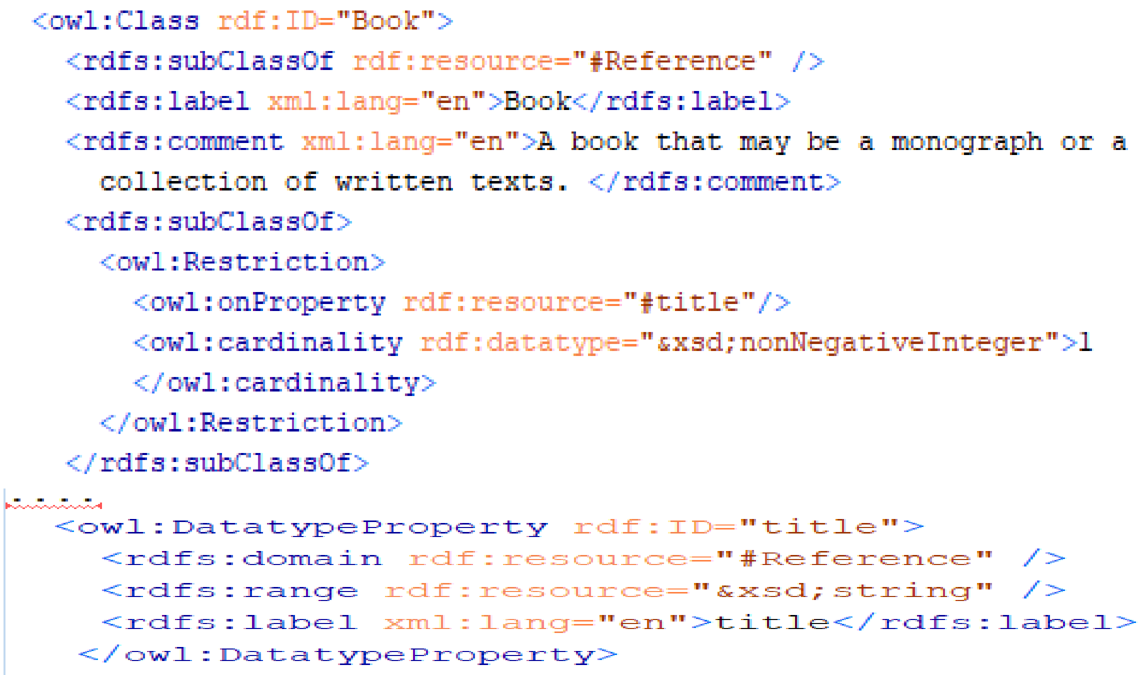
$$O2Sim(C_1, C_2) = \frac{\alpha_1 * DeSim(C_1, C_2) + \alpha_2 * StSim(C_1, C_2)}{\alpha_1 + \alpha_2} \tag{1}$$

where α_1 and α_2 are the weight parameters between 0 and 1. In this paper, we assume that DeSim and StSim have an equivalent role, so 0.5 is assigned to both α_1 and α_2. These weight factors are used to scale the O2Sim results to 0 and 1. Higher O2Sim values represent a greater similarity between elements of two OWL ontologies.

2.1. Description Similarity (DeSim)

The OWL ontology comprises the vocabulary, the data model, and the data type. The vocabulary allows us to determine the name similarity between nodes of two OWL ontologies. The data model, which represents the relationship of the entities, is used to compute the structural similarity. The data type helps us to improve the similarity quality between properties. For instance, consider a part of the 101 ontology in Benchmark[1] dataset described by OWL shown in Figure 2.

[1] http://oaei.ontologymatching.org/2010/benchmarks/index.html



```
<owl:Class rdf:ID="Book">
  <rdfs:subClassOf rdf:resource="#Reference" />
  <rdfs:label xml:lang="en">Book</rdfs:label>
  <rdfs:comment xml:lang="en">A book that may be a monograph or a
    collection of written texts. </rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#title"/>
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
      </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>

 <owl:DatatypeProperty rdf:ID="title">
   <rdfs:domain rdf:resource="#Reference" />
   <rdfs:range rdf:resource="&xsd;string" />
   <rdfs:label xml:lang="en">title</rdfs:label>
  </owl:DatatypeProperty>
```

Figure 2. *A part of 101 ontology described by OWL*

In Figure 2, the node named *Book* is defined by owl:Class, rdfs:subClassOf, rdfs:label, rdfs:comment. The node *Book* also has properties, such as *title* and *volume*. Those properties have their domain, range, and label. In our approach, the description similarity between concepts is included the similarity of its name and the similarity of its definition. There are two types of concepts, class and property. The name similarity (NSim) of the class and the property is the same, but the definition similarity (DefSim) of the class includes the definitions of the subclass, label, and comment, meanwhile the DefSim of the property computes the similarity of the domain, range, and label.

The description similarity (DeSim) between two concepts C_1 in the ontology 1 (O_1) and C_2 in the ontology 2 (O_2) is as the following:

$$DeSim(C_1, C_2) = \frac{\beta_1 * NSim(C_1,C_2) + \beta_2 * DefSim(C_1,C_2)}{\beta_1 + \beta_2} \tag{2}$$

where β1 and β2 are the weight parameters between 0 and 1. In this paper, we assume that NSim and DefSim have an equivalent role, so 0.5 is assigned to both β_1 and β_2. Each similarity measure is presented in the following subsections.

2.1.1. Name Similarity (NSim)

The name similarity computes the linguistic and semantic similarity between concepts in two OWL ontologies. Concept names in the OWL file are often declared as a word or a set of words. Moreover, since OWL tags are created freely, similar semantic notions can be represented by different words (e.g., title and name), or different elements can have linguistic similarities (e.g., book and paperback).

The name similarity between elements is computed by three main steps. The first step normalizes each element name to remove genitives, punctuation, capitalization, stop words (such as, of, and, with, for, to, in, by, on, and the), and inflection (plurals and verb conjugations).

The second step finds the synonyms for each compared element name by looking them up in the WordNet[2] thesaurus and then computes the name similarity between elements. To obtain a high quality of name similarity, we measure both linguistic and semantic similarities. The linguistic step computes the string similarity of the entity names by matching two string names. The linguistic similarity metric between two entities C1 and C2 is:

$$LingSim(C_1, C_2) = \frac{n_{C_1 \cap C_2}}{\max(n_{C_1}, n_{C_2})} \tag{3}$$

where $n_{C_1 \cap C_2}$ is the number of matching characters between elements C_1 and C_2; max is the maximum value; n_{C_1} and n_{C_2} are the lengths of the elements C_1 and C_2, respectively. For example,

$$LingSim(MasterThesis, PhdThesis) = \frac{n_{MasterThesis \cap PhdThesis}}{max(n_{MasterThesis}, n_{PhdThesis})} = \frac{6}{12} = 0.5$$

The proposed linguistic similarity measurement (3) works effectively when two entities are not entirely identical in their names. Specifically, when two element names are not found in WordNet, the LingSim value is their final name similarity result.

When one of the two compared elements is found in WordNet, we compute the semantic similarity for two synonym sets of the two elements. The metric for measuring the semantic similarity between two elements, C_1 and C_2 is:

$$SeSim(C_1, C_2) = \frac{2 * \sum_{i=1}^{n_{sc_1}} \sum_{j=1}^{n_{sc_2}} LingSim(C_1.sc_1[i], C_2.sc_2[j])}{n_{sc_1} + n_{sc_2}} \tag{4}$$

where sc_1 and sc_2 are the synonym sets of the elements C_1 and C_2, respectively; n_{sc_1} and n_{sc_2} are the numbers of entities in sc_1 and sc_2, respectively.

Using linguistic computation in semantic analysis improves the quality of the name similarity measurement when entities in each synonym set are not entirely identical. If two compared elements are not found in the WordNet, the name similarity (NSim) is the linguistic similarity, NSim = LingSim; otherwise, NSim=SeSim.

The third step computes the name similarity for tokenized elements in the first step. Since each combined element is split into token lists, the similarity of elements C_1 and C_2 equals two token lists T_1 and T_2. The metric for computing the name similarity between T_1 and T_2 is:

[2] http://wordnet.princeton.edu/wordnet

$$NSim(T_1,T_2)=\frac{\sum_{C_1\in T_1}\max_{C_2\in T_2}(SeSim(C_1,C_2))+\sum_{C_2\in T_2}\max_{C_1\in T_1}(SeSim(C_1,C_2))}{n_{T_1}+n_{T_2}} \quad (5)$$

where n_{T_1} and n_{T_2} are the numbers of words in the token sets of the concepts C_1 and C_2, respectively. Two elements are considered to be similar if their name similarity exceeds a given threshold.

2.1.2. Definition Similarity (DefSim)

As we discussed, there are two types of definition similarity, the first for the class concept and the second for the property concept. For the class concept, we compute the linguistic similarity between three definitions, including rdfs:subClassOf (su), rdfs:label (la) and rdfs:comment (co).

The definition similarity (DefSim) of two classes C1 and C2 in different OWL ontologies is determined by the following equation:

$$DefSim(C_1,C_2)=\gamma_1 * LingSim(su.C_1,su.C_2)+\gamma_2 * LingSim(la.C_1,la.C_2)+(1-\gamma_1-\gamma_2) * LingSim(co.C_1,co.C_2) \quad (6)$$

where γ_1 and γ_2 are weight parameters. Since subClassOf (su) plays an important role in class definitions, the definition of the label is usually the same as the declaration of the name of the class. It also plays an important role. Whereas the definition of a comment is a different explanation for the class name, sometimes some classes do not have a comment. Therefore, we assign weights γ_1 and γ_2 to 0.4, leaving 0.2 for comment similarity (co).

For the similarity between properties, we compute the similarity of the property's domain, label, and range. For the domain (do) and label (lab), we use linguistic similarity (equation number 3). However, values of the range are the datatype. Therefore, we propose the DtSim to measure the similarity between range values. The definition similarity (DefSim) of two properties C_1 and C_2 in different OWL ontologies is determined by the following equation:

$$DefSim(C_1,C_2)=\delta_1 * LingSim(do.C_1,su.C_2)+\delta_2 * LingSim(lab.C_1,la.C_2)+(1-\delta_1-\delta_2) * DtSim(C_1,C_2) \quad (7)$$

where δ_1 and δ_2 are weight parameters. Because domain (do) indicates the class to which the property belongs, it is more important than the other two properties (lab and DtSim), so we assign 0.4 to δ_1 and 0.3 to the other two parameters.

To compute the range similarity of properties, we propose a novel metric as in equation number 10. Since most of OWL's data types are similar to those of XML Schema, we explore the constraining facets of XML Schema data type[3], and then define the metric for measuring the similarity among the data types based on their constraining similarity:

[3] https://appletree.or.kr/quick_reference_cards/XML-XSLT-UML/XML%20Schema%20-%20Data%20Types.pdf

$$DSim1(C_1, C_2) = \frac{\sum_i \left| \left\{ cf_i \mid C_1[cf_i] = C_2[cf_i], 1 \le i \le n_{cf} \right\} \right|}{max(n_{C_1.cf}, n_{C_2.cf})} \quad (8)$$

where DSim1 is the data type similarity based on the resemblance of constraining facets; cf is one of the constraining facets described in [6], $max(n_{C_1.cf}, n_{C_2.cf})$ is the maximum number of constraining facets of the data type of the elements C_1 and C_2.

The results of equation (8) are quite acceptable except for some illogical values. For instance, the resemblance of date and float is 1.0, and the similarity between decimal and integer is also 1.0, although the number of constraining facets between date and decimal is different. Instead, we expect that those similarity values are less than 1.0, and the similarity between decimal and integer is higher than that of date and float.

Thus, we insert another metric to measure the data type similarity based on the number of constraining facets of each data type over the total number of constraining facets. This technique is names DSim2, and it is determined by the following equation:

$$DSim2(C_1, C_2) = \frac{max(n_{C_1.cf}, n_{C_2.cf})}{n_{cf}} \quad (9)$$

where $max(n_{C_1.cf}, n_{C_2.cf})$ is the maximum number of constraining facets of the data type of the element C_1 and C_2; ncf is the number of constraining facets, in this case ncf =12.

The combination of DSim1 and DSim2 produces the data type similarity (DtSim) of two elements C_1 and C_2. DtSim is measured by the following definition:

$$DtSim(C_1, C_2) = \frac{\phi_1 * DSim1(C_1, C_2) + \phi_2 * DSim2(C_1, C_2)}{\phi_1 + \phi_2} \quad (10)$$

where ϕ1 and ϕ2 are weight parameters between 0 and 1. In this paper, we assign 0.5 to ϕ1 and ϕ2 since we assume that DSim1 and DSim2 have similar roles. With equation (9), we can moderate the results of data type similarity. The final data type similarity (DtSim) among some common OWL data types is presented in Table 1.

Table 1. *OWL data type compatibility by equation (10)*

	string	**decimal**	**float**	**integer**	**long**	**date**	**time**
string	1.000	0.542	0.506	0.542	0.542	0.506	0.506
decimal	0.542	1.000	0.764	0.875	0.875	0.764	0.764
float	0.506	0.764	1.000	0.764	0.764	0.792	0.792
integer	0.542	0.875	0.764	1.000	0.875	0.764	0.764
long	0.542	0.875	0.764	0.875	1.000	0.764	0.764
date	0.506	0.764	0.792	0.764	0.764	1.000	0.792
time	0.506	0.764	0.792	0.764	0.764	0.792	1.000

In Table 1, if two elements have the same data type, their compatible value is 1.000. Otherwise, this value is assigned by equation (10).

2.2. Structure Similarity (StSim)

The structure similarity (StSim) between two concepts, C_1 in OWL1 and C_2 in OWL2, is computed based on the assumption that two elements are similar if their ancestor elements and their children are similar. Therefore, we compute the structure similarity by including these two factors. The structure similarity (StSim) of two concepts C_1 and C_2 determined by the following equation (11):

$$StSim(C_1, C_2) = \varepsilon * SpSim(C_1, C_2) + (1 - \varepsilon) * ChSim(C_1, C_2) \quad (11)$$

where SpSim is the super (ancestor) similarity; ChSim is the children similarity; ε is the weight parameter. Since the roles of SpSim and ChSim are assumed to be equivalent, we assign 0.5 to ε.

2.2.1. Super Similarity (SpSim)

The super concepts are the set of super classes defined from the rdfs:subClassOf and the rdfs:domain of those concepts. For instance, the super entities of the element SportCar in Fig. 3 are Vehicle, power, and registeredTo. Usually, the super entity of each element within a OWL Schema document contains several elements. Therefore, the super similarity between two elements C_1 and C_2 is the average similarity of two super element lists.

For instance, the super element of an element C_1 is SC1 = [C11, C12, …, C1k], and the super element of an element C_2 is SC2 = [C21, C22, …, C2t], where k and t are the numbers of super elements of the elements C_1 and C_2, respectively. If k ≥ t, we take each element in SC1 to compare with each element in SC2. Otherwise, if k < t, we compare each element in SC2 with each element in SC1. The highest value of the measurement is chosen. The super similarity (SpSim) of two concepts C_1 and C_2 is presented as following matrices (12) and (13):

$$SpSim(C_1,C_2) = \begin{bmatrix} DcSim(C_{11},C_{21}) \cdots DcSim(C_{11},C_{2t}) \\ \vdots \quad\quad \ddots \quad\quad \vdots \\ DcSim(C_{1k},C_{21}) \cdots DcSim(C_{1k},C_{2t}) \end{bmatrix}, \text{ k≥t} \quad (12)$$

$$SpSim(C_2,C_1) = \begin{bmatrix} DcSim(C_{21},C_{11}) \cdots DcSim(C_{21},C_{1k}) \\ \vdots \quad\quad \ddots \quad\quad \vdots \\ DcSim(C_{2t},C_{11}) \cdots DcSim(C_{2t},C_{1k}) \end{bmatrix}, \text{ k<t} \quad (13)$$

where DcSim is the description similarity between each super element of element C_1 and each super element of element C_2. It is determined by the equation (2). The super similarity of two elements C_1 and C_2 presented in matrices (12) and (13) is determined by the following equations (14) and (15), respectively.

$$SpSim(C_1,C_2) = \frac{\sum_{i=1}^{k} \max_{j=1}^{t}(DcSim(C_{1i},C_{2j}))}{k} \quad (14)$$

$$SpSim(C_2,C_1)=\frac{\sum_{i=1}^{t}\max_{j=1}^{k}(DcSim(C_{2i},C_{1j}))}{t} \tag{15}$$

where max is the maximum similarity value of each row in the matrix.

If two elements C_1 and C_2 do not have any super element (it means they are root elements), then $SpSim(C_1,C_2)=1$. In the case that one of the two compared elements is a root element, then $SpSim(C_1,C_2)=0$.

2.2.2. Children Similarity (ChSim)

Children of an element C are the collection of properties of element C and all subclasses of element C and the corresponding properties of those subclasses. Similar to the super computation, to calculate the children similarity of two concepts C_1 in OWLS1 and C_2 in OWLS2, we collect all children of concepts C_1 and C_2 and then compare the description similarity of each children pair. Assume that m and n are the numbers of children of the element C_1 and C_2, respectively, the children similarity (ChSim) between two concepts C_1 and C_2 can be presented as following matrices (16) and (17):

$$ChSim(C_1,C_2)=\begin{bmatrix} DcSim(C_{11},C_{21})\cdots DcSim(C_{11},C_{2n}) \\ \vdots \quad \ddots \quad \vdots \\ DcSim(C_{1m},C_{21})\cdots DcSim(C_{1m},C_{2n}) \end{bmatrix}, \text{ m}\geq\text{n} \tag{16}$$

$$ChSim(C_2,C_1)=\begin{bmatrix} DcSim(C_{21},C_{11})\cdots DcSim(C_{21},C_{1m}) \\ \vdots \quad \ddots \quad \vdots \\ DcSim(C_{2n},C_{11})\cdots DcSim(C_{2n},C_{1m}) \end{bmatrix}, \text{ m<n} \tag{17}$$

where DcSim is the semantic similarity (SeSim) of each child element of C_1 and each child element of C_2. The children similarity of two elements C_1 and C_2 in the matrices (16) and (17) are determined by the following equations (18) and (19), respectively:

$$ChSim(C_1,C_2)=\frac{\sum_{i=1}^{m}\max_{j=1}^{n}(DcSim(C_{1i},C_{2j}))}{m} \tag{18}$$

$$ChSim(C_2,C_1)=\frac{\sum_{i=1}^{n}\max_{j=1}^{m}(DcSim(C_{2i},C_{1j}))}{n} \tag{19}$$

In the case that one of the elements C_1 and C_2 is the leaf node (that means it contains no child node), their children similarity is 0.

3. Experimental results

The semantic similarity between concepts in different OWL ontologies (O2Sim) is implemented with C# language. To compare the name similarity (NSim) in the description measurement, we integrate WordNet and its .NET API, which is provided by Troy and Crowe (2005) into our implementation.

We evaluate the proposed measures in the context of matching two OWL ontologies to determine the number of matches between them and then compare them with other approaches. The criteria for evaluating the quality of matching system are precision and recall[4], which originate from information retrieval and are adapted to ontology matching (Do & Erhard, 2002). Precision reflects the share of real correspondences among all found correspondences.

To examine the performance of O2Sim, we use ten specific OWL ontologies from Benchmark dataset as source ontologies. The characteristics of ten OWL ontologies are presented in Table 2.

Table 2. *The characteristics of the tested ontologies*

#	A couple of ontologies	Description
1	101-104	The hierarchical structure is the same. Same or completely different entity names.
2	201-210	The hierarchical structure is the same. Different semantics are used at several levels.
3	221-247	Different hierarchical structure. The label is semantically the same.
4	248-266	Different hierarchical structure and semantics.
5	301-304	Real-world ontologies, provided by various organizations.

To obtain the average result from five pairs of test schemas, we use the weighted average, which is the number of correct matches of each test case, as the weighted factor. The precision and recall values are calculated by the following equations:

$$precision_{avg} = \frac{\sum_{i=1}^{n}\left(W_i * precision_i\right)}{\sum_{i=1}^{n} W_i} \qquad (20)$$

$$recall_{avg} = \frac{\sum_{i=1}^{n}\left(W_i * recall_i\right)}{\sum_{i=1}^{n} W_i} \qquad (21)$$

where n is the number of test cases (in this experiment, n = 5); W_i is the number of correct matches of the test case number i; $precision_i$ and $recall_i$ are the precision score and recall score of the test case number i. The results of the simulation are presented in the next section.

Since our approach uses the hybrid method to compute the similarity of concepts between OWL ontologies, we compare our method to similar works such as Xu et al. (2020), Sun et al. (2021), and Han et al. (2017). The precision, recall, and F-measure values among O2Sim and related work are presented in Figures 3, 4, and 5, respectively. In this paper, the threshold values are chosen between 0.3 and 1 since those similarity values lower than 0.3 are primarily different and easy to determine by human observation.

[4] http://en.wikipedia.org/wiki/Precision_and_recall

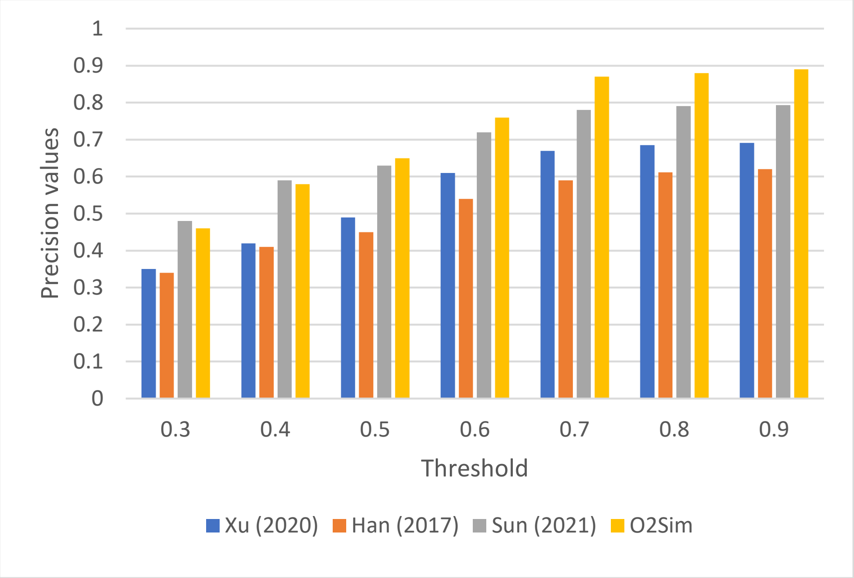


Figure 3. *Precision among O2Sim and related approaches*

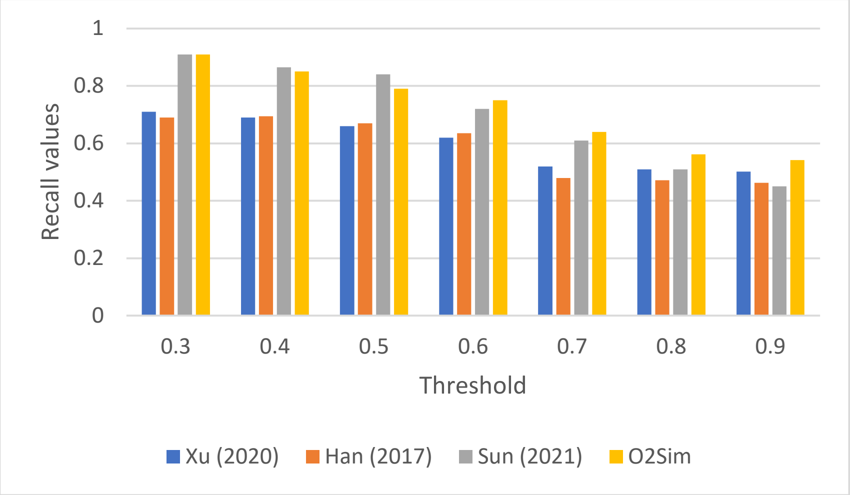


Figure 4. *Recall among O2Sim and related approaches*

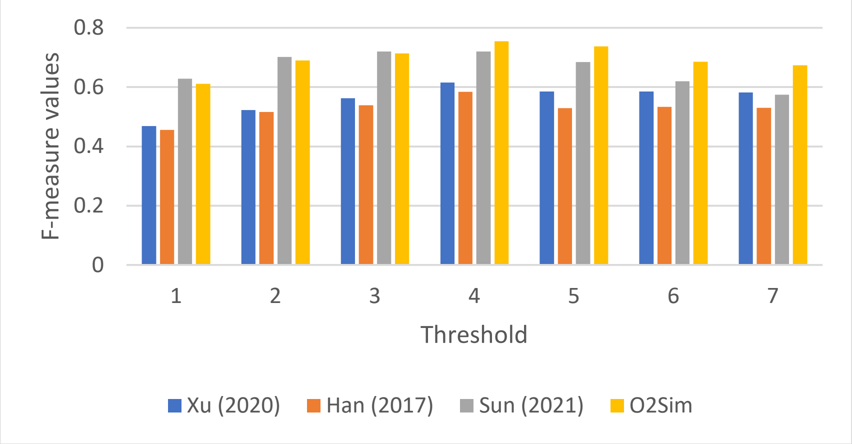


Figure 5. *F-measure among O2Sim and related work*

The comparison results in Figures 3, 4, and 5 show that our O2Sim significantly outperforms the other methods at all thresholds, followed by the methods of Sun et al. (2021), Xu et al. (2020), and Han et al. (2017). The Sun (2020)'s method outperforms the O2Sim when the thresholds are equal to or less than 0.5. The main reason for this is that the data type similarity values of Sun'method are very high and based on the user's judgment. However, for high threshold values, Sun's method has less accurate similarity values. The measures of Xu and Han have poor results since they are based on the linguistic similarity of concept names. However, Xu's method is better than Han's method since Xu's approach still considers the data type similarity.

4. Conclusions

This paper proposes a novel similarity measuring technique for OWL concepts. We present a semantic similarity measurement method that computes description and structure resemblances. The experimental evaluation demonstrates that our method outperforms human judgment and related approaches. Further, combining all measuring factors provides important information for deriving the correct similarity values.

We hope the research has established a foundation to help the integration of different OWL ontologies. If this method is popularized, a large amount of OWL data on the current Web will be integrated into the useful ontology for the Semantic Web and its applications.

❖ ***Conflict of Interest:*** *Author have no conflict of interest to declare.*

REFERENCES

Akbari, I., & Fathian, M. (2010). A Novel Algorithm for Ontology Matching. *Information Science*, *36*(3), 324-334.

Cheng, L., Jiang, Y., Ju, H., Sun, J., Peng, J. Zhou, M., & Hu, Y. (2018). InfAcrOnt: calculating cross-ontology term similarities using information flow by a random walk. *BMC Genomics* 19 (Suppl 1), 919. https://doi.org/10.1186/s12864-017-4338-6

Do, H. H., & Erhard, R. (2002). *COMA - A System for Flexible Combination of Schema Matching Approaches*. Proceedings of the Very Large Data Bases conference (VLDB), (pp 610–621).

Fayez, A., & Althobaiti, S. (2017). Comparison of Ontology-Based Semantic-Similarity Measures in the Biomedical Text. *Journal of Computer and Communications*, *5*(2), 17-27.

Han, X., Wang, Q, Guo, Y., & Cui, X. (2017). Geographic Ontology Concept Semantic Similarity Measure Model Based on BP Neural Network Optimized by PSO. *Comput. Eng. Appl.*, 53, 32-37.

Jean-Mary, Y. R., & Shironoshita, E. P., & Kabuka, M. R. (2009). Ontology Matching with Semantic Verification. *Journal of Web Semantics, 7*(3), 235-251.

Jiang, J. J., & Conrath, D. W. (1997). *Semantic similar- ity based on corpus statistics and lexical taxonomy*. Proc. Int. Conf. on Research in Computational Linguistics, (pp.19-33).

Lin, D. (1998). An information-theoretic definition of similarity. *Proceedings of the 15th International Conference on Machine Learning,* (pp 296-304).

Mingxin, G.**,** Xue, D., & Rui, J. (2013). From Ontology to Semantic Similarity: Calculation of Ontology-Based Semantic Similarity. *Journal of Computational Systems Biology, 2013,* https://doi.org/10.1155/2013/793091

Nguyen, T. T. A., & Conrad, S. (2015). Ontology Matching using multiple similarity measures. *7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K),* (pp. 603-611).

Preeti, R., & Sanjay, K. M. (2020). IWD towards Semantic similarity measure in ontology. *Journal of Information and Optimization Sciences*, *41*(7). Applied Machine Learning for IoT and Smart Data Analysis (Part-II).

Resnik, P. (1999). Semantic similarity in taxonomy: An information- based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, *11*, 95-130.

Sánchez, D., Batet, M., Isern, D., & Valls, A. (2010). Ontology-based Semantic Similarity: A New Feature-based Approach. *Expert Systems with Applications*, *39*(9), 7718-7728.

Stoilos, G., Stamou, G., & Kollias, S. (2005). *A String Metric for Ontology Alignment*. In proc. of the 4th International Semantic Web Conference, Springer LNCS, 3729, 624-637.

Sun, L., Wei, Y., & Wang, B. (2021). Similarity Calculation Method of Multisource Ontology Based on Graph Convolution Network. *Chin. J. Netw. Inf. Secur.*, 7, 149-155. Retrieved from http://kns.cnki.net/kcms/detail/10.1366.TP.20210610.1503.004.html

Troy, S., & Crowe, M. (2005). *WordNet.Net.* Retrieved from http://opensource.ebswift.com/WordNet.Net

Xu, F., Ye, X., Li, L., Cao, J., & Wang, X. (2020). Comprehensive Calculation of Semantic Similarity of Ontology Concept Based on SA-BP. *Comput. Sci.*, 47, 199-204.

Zhao, C., & Wang, Z. (2018). GOGO: An improved algorithm to measure the semantic similarity between gene ontology terms. *Sci Rep* 8, 15107, https://doi.org/10.1038/s41598-018-33219-y

NÂNG CAO SỰ SO KHỚP GIỮA CÁC TÀI LIỆU ONTOLOGIES DỰA VÀO ĐÁNH GIÁ ĐỘ TƯƠNG ĐỒNG VỀ NGỮ NGHĨA

Phạm Thị Thu Thúy

Trường Đại học Nha Trang, Việt Nam

Tác giả liên hệ: Phạm Thị Thu Thúy – Email: thuythuy@ntu.edu.vn

Ngày nhận bài: 18-10-2022; ngày nhận bài sửa: 26-10-2022; ngày duyệt đăng: 28-10-2022

TÓM TẮT

Gần đây, ngôn ngữ bản thể học (OWL ontology) đã trở thành một ngôn ngữ được sử dụng rộng rãi để cung cấp một nguồn các khái niệm được định nghĩa chính xác. Số lượng tài liệu OWL tăng tỉ lệ thuận với sự phát triển của web ngữ nghĩa dẫn đến vấn đề không đồng nhất dữ liệu. Các khái niệm giống nhau có thể được định nghĩa khác nhau bởi các thuật ngữ khác nhau và nằm ở các vị trí khác nhau trong cấu trúc tài liệu. Do đó, việc xác định sự giống nhau của phần tử trong các tài liệu ontology khác nhau trở nên quan trọng đối với sự thành công của các hệ thống tích hợp thông tin và khai thác web. Trong bài báo này, chúng tôi đề xuất một biện pháp đánh giá độ tương đồng ngữ nghĩa để so sánh các phần tử trong các tài liệu OWL khác nhau. Phương pháp này đề cập đến việc tính toán độ tương đồng về các mô tả của các phần tử và các mối quan hệ của phần tử đó với các lớp trên và con cái của nó. Chúng tôi đánh giá các công thức được đề xuất bằng cách tính toán và so khớp hai tài liệu OWL để xác định số lượng khớp giữa chúng. Kết quả thử nghiệm cho thấy sự cải thiện của chúng tôi về độ chính xác so với các phương pháp tiếp cận khác.

Từ khóa: matching; measure; ontology; OWL; semantic similarity