Kỷ yếu Hội nghị Khoa học Quốc gia lần thứ IX "Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR'9)"; Cần Thơ, ngày 4-5/8/2016
DOI: 10.15625/vap.2016.0004

AN IMPROVEMENT IN MEASURING THE SEMANTIC SIMILARITY BETWEEN RDF ONTOLOGIES

Pham Thi Thu Thuy[(1)], Nguyen Dang Tien [(2)]

[(1)]Nha Trang University, [(2)]People's Police University and Logistics
thuthuy@ntu.edu.vn, dangtient36@gmail.com

Abstract— RDF (Resource Description Framework) ontologies has been playing an important role for many knowledge applications because they support a source of precisely defined terms. However, the wide-spread of RDF ontologies creates a demand for automatic way of assessing their similarity. In this paper, we present a novel method to measure the semantic similarity between elements in different RDF ontologies. This measure is designed so as to enable extraction of information encoded in RDF element descriptions and to take into account the element relationships with its ancestors and children. We evaluate the proposed measures in the context of matching two RDF ontologies to determine the number of matches between them and then compare with human estimation and the related methods. The experimental results show that our similarity values are better than other approaches with regard to the accuracy of semantics and structure similarities.

Keywords— Similarity, RDF Ontologies, Measure.

I. INTRODUCTION

RDF (Resource Description Language) and its supporting vocabulary language, RDF Schema, have become widely-used languages for representing data in the Semantic Web [1]. However, the increasing number of RDF ontologies leads to the heterogeneity problem. The same entities may be modeled differently by using different terms or placed in different positions in the entity hierarchy. This heterogeneous problem causes a great challenge to integrate the RDF ontologies. Measuring the entity similarity between two RDF ontologies is the core for the success of the information integration.

Several approaches have been proposed to measure the entity similarity between different ontologies [2-4], or measure the similarity between a given text with the text in a RDF document [26]. However, most of these methods only consider the information which describes the entities such as name, definition, and property. Further, the similarity values of some factors such as data type and definition are given by the users' judgment [11, 12].

This paper presents a novel method that measures the semantic similarity between entities from different RDF ontologies. The semantics of the entities are implied in name, their descriptions and their relationships with other entities in the schema tree. This paper's contributions are several:

- It proposes novel measures to compute the definition similarity in RDF.
- It discusses and introduces novel measure to calculate the name and data type similarities of RDF elements.
- It describes a set of experiments conducted to evaluate our computations and compares them with human judgment and with related work.

The remainder of the paper is organized as follows. The related methods are presented in Section 2. Section 3 describes the motivating example. Section 4 discusses our approach to measuring RDF similarity. The experiment evaluation is given in Section 5. Finally, Section 6 concludes the paper.

II. RELATED WORK

In this section, we present two research directions that are related to our paper: (1) Matching between two RDF documents; (2) Measuring the similarity between entities in different documents.

First, there are several approaches related to RDF Schema matching. Leme et al. [14, 24] introduce some RDF property matching heuristics based on similarity functions. However, the matching only works well if two elements exactly have the same name, data type, and other relations whereas our approach considers not only the linguistics but also the semantics of the element names. Oldakowski et al. [7] and Zhang et al. [13] propose a matching between two RDF graphs. However, both the methods find the matches by relying on the distance similarity of objects in the RDF graphs and they did not concern the definition and data type similarities among entities. Samur Araujo et al. [25] propose an instance matching between a source and a target datasets.

Second, some approaches are proposed to measure the element similarity between documents. Yan et al. [8] and Kling et al. [9] extend the distance-based method to find the similarity between XML elements for querying purpose. Do et al. [18] compute the name similarity between elements of two XML Schemas. Yang et al. [19] use linguistic taxonomy based on entity definitions in WordNet [10] to gain the most accurate semantics for the element names. Some researchers [11, 12, 17, 27], employ supplemental functions to calculate the similarity of a particular feature of a given

schema, such as structural similarity, the similarity of leaf nodes or root nodes, data types and constraints. All the partial results are then combined into the final similarity value using a weighted sum function.

In general, approaches in the first direction try to find matches between a source RDF element and a target RDF element. The main technique of these matches are based on the exact name and data type similarities between the source and destination. Our method is most similar to the second approaches, although our computation focuses on the similarity between elements in different RDF Schemas. However, the important difference between these approaches and our approach is that the description, the name, and the data type similarity values are derived with our proposed measures without any user intervention. This paper is the extended version of our previous paper [23]. In this version, we update the description similarity with a datatype compatibility table and add the metrics for calculating super similarity and children similarity. Then we also update the experimental result.

III. R2SIM FRAMEWORK AND MOTIVATING EXAMPLE

The framework of R2Sim includes the input, the R2Sim computation, and the output. The input is two RDF Schemas. The main component of this framework is the R2Sim computation, which is composed of the description and neighborhood similarity measures. The outputs are the similarity values of elements between RDF Schemas. The R2Sim framework is depicted in Fig. 1.

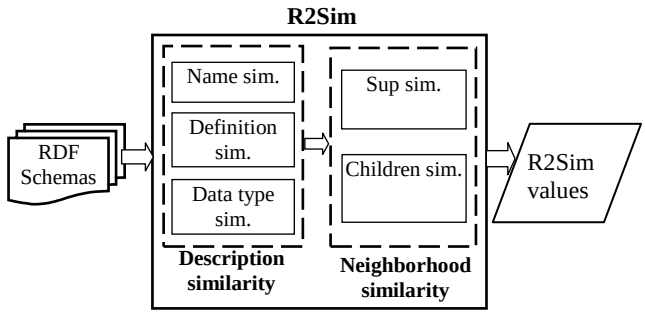

Fig. 1. *The framework of the R2Sim method.*

The description similarity in Fig. 1 comprises the similarity of the element name (*Name sim.*), the definition similarity, and the data type similarity. The neighborhood similarity encompasses two individual measures: the super element similarity (*Sup sim.*) and the children similarity (*Children sim.*). The final R2Sim similarity is the combination of all the partial results using a weighted sum function.

To illustrate the R2Sim method, we first restrict ourselves to the hierarchical schemas. The RDF Schemas are encoded as graphs, where the nodes represent the schema elements and the vectors indicate the relationship between elements. We motivate R2Sim with the real RDF data set *MotorVehicle.rdfs* [1] and the *Vehicle.rdfs* which is extracted from the book [5]. The representing trees of two RDFS files are displayed in Fig. 2 and Fig. 3, respectively.

s = rdfs:subClassOf ; t = rdf:type

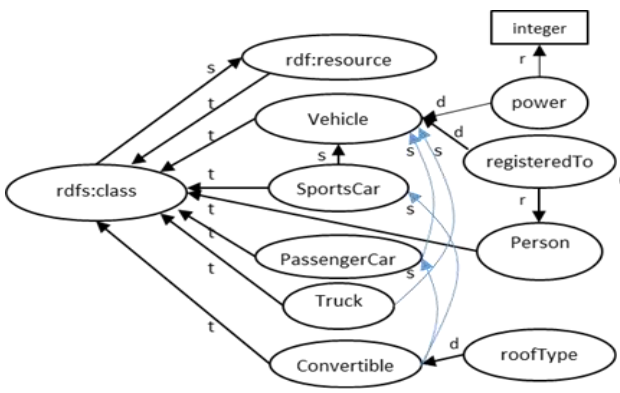

Fig. 2. Tree representation of *MotorVehicle.rdfs* [1]

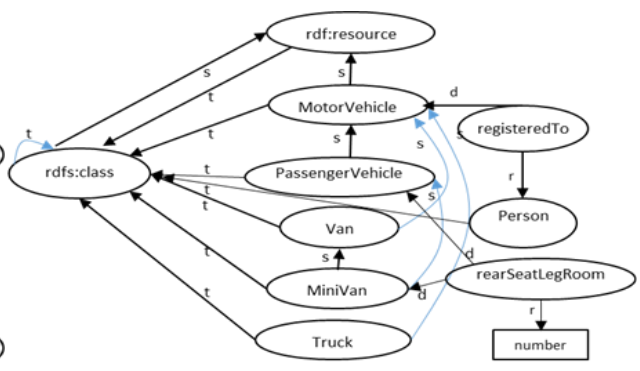

Fig. 3. Tree representation of *Vehicle.rdfs* [5]

In Fig. 2 and Fig. 3, the characters *t*, *s*, *d*, and *r* are short forms of *rdf:type*, *rdfs:subClassOf*, *rdfs:range*, and *rdfs:domain*, respectively. Although it is obvious that there are common characteristics between some elements in Fig. 2 and Fig. 3, there is also much variation between element descriptions and their neighborhood relationships that

challenge the measuring algorithm. Our motivation is to find the most suitable matching from each entity in Fig. 2 to one entity in Fig. 3. Details of each similarity measurement are presented in the next sections.

IV. RDFSIM METHOD

The semantic similarity between entity C_1 and C_2 is defined as the weighted sum of the description similarity (*DcSim*) and the neighborhood similarity (*NbSim*):

$$R2Sim(C_1,C_2)=\frac{\alpha_1 * DcSim(C_1,C_2)+\alpha_2 * NbSim(C_1,C_2)}{\alpha_1+\alpha_2} \quad (1)$$

Where α_1 and α_2 are the weight parameters between 0 and 1. In this paper, we assume that *DcSim* and *NbSim* have an equivalent role, so *0.5* is assigned to both α_1 and α_2. These weight factors are used to scale the R2Sim results to range between 0 and 1. Higher R2Sim values represent a greater similarity between elements of two RDF Schemas.

4.1. Description Similarity

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [<!ENTITY xsd
  "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xml:base="http://www.uibk.ac.at/iwi/schemas/vehicles">
<rdfs:Class rdf:ID="Vehicle"/>
<rdfs:Class rdf:ID="SportsCar">
    <rdfs:subClassOf rdf:resource="Vehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="PassengerCar">
    <rdfs:subClassOf rdf:resource="Vehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Truck">
    <rdfs:subClassOf rdf:resource="Vehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Convertible">
  <rdfs:subClassOf rdf:resource="Passengercar"/>
  <rdfs:subClassOf rdf:resource="Sportscar"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Person"/>
<rdf:Property rdf:ID="registeredTo">
  <rdfs:domain rdf:resource="#Vehicle"/>
  <rdfs:range rdf:resource="#Person"/>
</rdf:Property>
<rdfs:Datatype rdf:about="&xsd;integer"/>
<rdf:Property rdf:ID="power">
    <rdfs:domain rdf:resource="#Vehicle"/>
    <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
<rdf:Property rdf:ID="roofType">
    <rdfs:domain rdf:resource="#Convertible"/>
</rdf:Property>
</rdf:RDF>
```

Fig. 4. Expressions for RDF Schema *Vehicle.rdfs.*

The RDFS comprises of the vocabulary, the data model, and the data type. The vocabulary allows us to determine the name similarity between nodes of two RDF Schemas. The data model, which represents the relationship of the entities, is used to compute the neighborhood similarity. The data type helps us to improve the similarity quality between properties. For instance, consider a RDF Schema for *Vehicle.rdfs* in Fig. 4.

In Fig. 4, the vocabulary includes *Vehicle*, *SportCar*, *registeredTo*, and so on, which are defined by *rdfs:Class* or *rdf:Property*. The data model represented by *rdfs:subClassOf*, *rdfs:domain*, *rdfs:range*, and so on, expresses the relationship of an entity with its super and children entities. The data types are defined externally to RDF Schema, and referenced by their URIrefs [5]. In this paper, the vocabulary, the data type, and some factors in the data model are combined to form the description similarity measure.

The description similarity between two entities C_1 in $RDFS_1$ and C_2 in $RDFS_2$ is defined as the weighted sum of the name similarity (*NSim*), definition similarity (*DfSim*), and data type similarity (*DtSim*) as follows:

$$DcSim(C_1,C_2)=\frac{\beta_1 * NSim(C_1,C_2)+\beta_2 * DfSim(C_1,C_2)+\beta_3 * DtSim(C_1,C_2)}{\beta_1+\beta_2+\beta_3} \quad (2)$$

Where β_1, β_2 and β_3 are the weight parameters between 0 and 1. Each similarity measure is presented in the following subsections. In the case that either entity C_1 or C_2 does not contain a data type description, then $DtSim(C_1,C_2)=0$.

4.1.1. Name Similarity

The name similarity computes the linguistic and semantic similarity between elements in two RDF Schemas. Element names in the RDF Schema are often declared as a word or a set of words. Moreover, since RDF tags are created freely, similar semantic notions can be represented by different words (e.g., *car* and *automobile*), or different elements can have linguistic similarity (e.g., *van* and *minivan*).

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). After normalizing the element name, the first step separates the composed element into single words. For example, *PassengerCar* becomes *Passenger* and *Car*.

The second step finds the synonyms for each compared element name by looking them up in the WordNet thesaurus [10] 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's metric between two entities C_1 and C_2 is:

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

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.

The proposed linguistic similarity measurement (3) works effectively when two entities are not completely identical in their names. Specifically, when two element names are not found in WordNet [10], 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:

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

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 the semantic computation improves the quality of the name similarity measurement when entities in each synonym set are not completely 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 elements that are tokenized in the first step. Since each combined element is split into token lists, the similarity of two elements C_1 and C_2 is equal to the similarity of two token lists T_1 and T_2. The metric for computing the name similarity between T_1 and T_2 is:

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

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

4.1.2. Definition Similarity

Since each entity is usually defined by several RDF Schema terms, the definition similarity of pair of entities must compute the resemblance of all of their terms. According to the class hierarchy and the constraint descriptions in the RDF Schema [1], we measure the similarity of four common RDFS terms, such as *rdf:type* (*rt*), *rdfs:subClassOf* (*rs*), *rdfs:range* (*rr*), and *rdfs:domain* (*rd*).

The definition similarity (*DfSim*) of two entities C_1 and C_2 in different RDF Schemas is determined by the following equation:

$$DfSim(C_1, C_2) = \chi * \frac{\min(C_1.rt, C_2.rt)}{\max(C_1.rt, C_2.rt)} + \delta * \frac{\min(C_1.rs, C_2.rs)}{\max(C_1.rs, C_2.rs)} + \varepsilon * \frac{\min(C_1.rr, C_2.rr)}{\max(C_1.rr, C_2.rr)} + (1 - \chi - \delta - \varepsilon) * \frac{\min(C_1.rd, C_2.rd)}{\max(C_1.rd, C_2.rd)} \quad (6)$$

Where χ, δ, and ε are weight parameters. Since the roles of four computed terms are assumed to be equivalent, we assign *0.25* to each of parameters; *min* and *max* are short forms of the minimum and maximum, respectively.

For instance, consider the definition similarity between *PassengerVehicle* (*PV*) and *PassengerCar* (*PC*) in Fig.2 and Fig. 3, respectively.

$$DfSim(PV, PC) = 0.25 * \frac{1}{1} + 0.25 * \frac{2}{2} + 0.25 * 0 + 0.25 * \frac{0}{1} = 0.5$$

4.1.3. Data Type Similarity

We found that other approaches related to measuring the similarity between data types, such as [11, 12], often assign the similarity value for each data type pair. In this paper, we propose a novel technique to calculate these values.

Since most of RDF Schema's data types are similar to those of XML Schema, we explore the constraining facets of XML Schema data type in [6], and then define the metric for measuring the similarity among the data types based on their constraining similarity:

$$DSim1(C_1, C_2) = \frac{\sum_i \left| \{ cf_i \mid C_1[cf_i] = C_2[cf_i], 1 \le i \le n_{cf} \} \right|}{max(n_{C_1.cf}, n_{C_2.cf})} \quad (7)$$

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 element C_1 and C_2.

The results of equation (7) 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 similarities 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:

$$DSim2(C_1, C_2) = \frac{max(n_{C_1.cf}, n_{C_2.cf})}{n_{cf}} \quad (8)$$

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; n_{cf} is the number of constraining facets, in this case $n_{cf} = 12$.

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:

$$DtSim(C_1, C_2) = \frac{\phi_1 * DSim1(C_1, C_2) + \phi_2 * DSim2(C_1, C_2)}{\phi_1 + \phi_2} \quad (9)$$

Where ϕ_1 and ϕ_2 are weight parameters between 0 and 1. In this paper, we assign *0.5* to ϕ_1 and ϕ_2 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 RDF data types are presented in Table 1.

Table 1. *RDF data type compatibility by equation (9)*

	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 (9).

4.2. Neighborhood Similarity

The neighborhood similarity (*NbSim*) between two elements C_1 in $RDFS_1$ and C_2 in $RDFS_2$ is computed based on the assumption that two elements are similar if their super elements and their children are similar. Therefore, we compute the neighborhood similarity by including these two factors. The neighborhood similarity (*NbSim*) of two elements C_1 and C_2 determined by the following equation (10):

$$NbSim(C_1, C_2) = \frac{\varphi_1 * SpSim(C_1, C_2) + \varphi_2 * ChSim(C_1, C_2)}{\varphi_1 + \varphi_2} \quad (10)$$

Where *SpSim* is the super similarity; *ChSim* is the children similarity; φ_1, and φ_2 are weight parameters. Since the roles of *SpSim* and *ChSim* are assumed to be equivalent, we assign *0.5* to φ_1 and φ_2.

4.2.1. Super Similarity

Super entities are the set of super classes defined by *rdfs:subClassOf* and the properties of those classes. For instance, the super entities of element *SportCar* in Fig. 3 are *Vehicle*, *power*, and *registeredTo*. Usually, the super entity of each element within a RDF 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 $SC_1 = [C_{11}, C_{12},\ldots, C_{1k}]$, and the super element of and element C_2 is is $SC_2 = [C_{21}, C_{22},\ldots, C_{2t}]$, where k and t are the numbers of super elements of the element C_1 and C_2, respectively. If $k \geq t$, we take each element in SC_1 to compare with each element in SC_2. Otherwise, if $k < t$, we compare each element in SC_2 with each element in SC_1. The highest value of the measurement is chosen. The super similarity (*SpSim*) of two elements C_1 and C_2 is presented as following matrices (11) and (12):

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

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

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 (11) and (12) is determined by the following equations (13) and (14), respectively.

$$SpSim(C_1, C_2) = \frac{\sum_{i=1}^{k} \max_{j=1}^{t}(DcSim(C_{1i}, C_{2j}))}{k} \tag{13}$$

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

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 two compared elements is a root element, then $SpSim(C_1,C_2) = 0$.

4.2.2. Children Similarity

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, in order to calculate the children similarity of two elements C_1 in $RDFS_1$ and C_2 in $RDFS_2$, we collect all children of elements 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 elements C_1 and C_2 can be presented as following matrices (15) and (16):

$$ChSim(C_1, C_2) = \begin{bmatrix} DcSim(C_{11}, C_{21}) \cdots DcSim(C_{11}, C_{2n}) \\ \vdots \quad \ddots \quad \vdots \\ DcSim(C_{1m}, C_{21}) \cdots DcSim(C_{1m}, C_{2n}) \end{bmatrix}, \text{m}\geq\text{n} \tag{15}$$

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

Where *DcSim* is the description similarity of each child element of element C_1 and each child element of element C_1. The children similarity of two elements C_1 and C_2 in the matrices (15) and (16) are determined by the following equations (17) and (18), respectively:

$$ChSim(C_1, C_2) = \frac{\sum_{i=1}^{m} \max_{j=1}^{n}(DcSim(C_{1i}, C_{2j}))}{m} \tag{17}$$

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

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*.

Depending on the expected similarity value (threshold value), the semantic similarity between two element C_1 and C_2 (*R2Sim*) can be divided into two groups, high similarity and low similarity, and then the matching and integrating strategies for those elements will be applied. In this paper, we assign *0.7* to the threshold value. Therefore, if value of *R2Sim* is greater than or equal *0.7*, then two elements are highly similar.

V. EXPERIMENTAL EVALUATION

We perform experiments to answer two questions.

1. How much advantage does the R2Sim provide, compared to other approaches?
2. How effective is each similarity factor in measuring the semantic similarity between elements in different RDF Schemas?

To answer these questions, we select the data set and set up the implementation as follows.

5.1. Data Set and Setup

The semantic similarity between elements in different RDF Schemas (R2Sim) 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 Simpson et al. [21] to our implementation. We evaluate the proposed measures in the context of matching two RDF Schemas to determine the number of matches between them and then compare with other approaches. The criteria for evaluating the quality of matching system are *precision* and *recall*, which originate from information retrieval [22] and were adapted to ontology matching [18].

To examine the performance of R2Sim, we download about 20 RDF Schemas from [20] as source schemas and then modify each source schema to generate a corresponding destination schema. This paper presents the test results with five RDFS sources from [20] and their modified schemas. The characteristics of five RDF Schemas are presented in Table 2.

Table 2. The characteristics of the tested schemas

#	Schema name	File size (KB)	# classes (source/ destination)	# properties (source/ destination)
1	GeneOntology	4642	7853/7000	40107/40200
2	RealEstateData	2311	2925/3050	19687/15600
3	ACM-Computing	231	312/312	1146/1200
4	MovieDatabase	86	96/80	379/300
5	Educational	8	17/30	26/40

In Table 2, the destination schema of *GeneOntology* is modified by increasing the number of properties and decreasing the number of classes of the source schema. In contrast, we decrease the number of properties and increase the number of classes of *RealEstateData*. For *ACM-Computing*, we keep the same number of classes and increase the number of properties. For *MovieDatabase*, we increase two numbers whereas we decrease those in *Education* schema. The results of simulation are presented in next section.

5.2. Experiment Results

Since our approach focuses on the similarity between RDF Schema elements, we compare our method to similar works such as Leme et al. [17], Do et al. [18], and Algergawy et al. [12]. The precision, recall and F-measure values among R2Sim and related work are presented in Fig. 5, Fig. 6, and Fig. 7. Note that in this paper, the threshold values are chosen between 0.3 and 1, since those similarity values lower than 0.3 are mostly different and easy to determine by human observing.

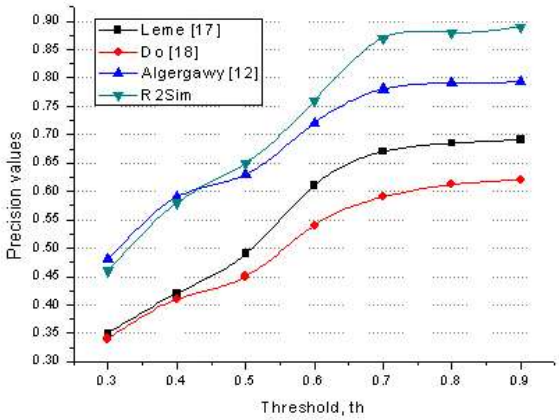

Fig. 5. Precision among R2Sim and related approaches

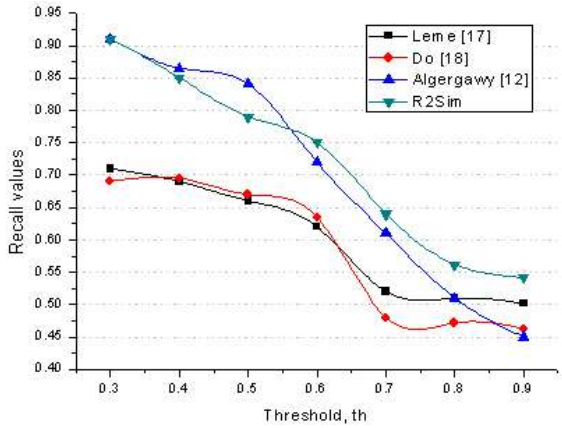

Fig. 6. Recall among R2Sim and related approaches

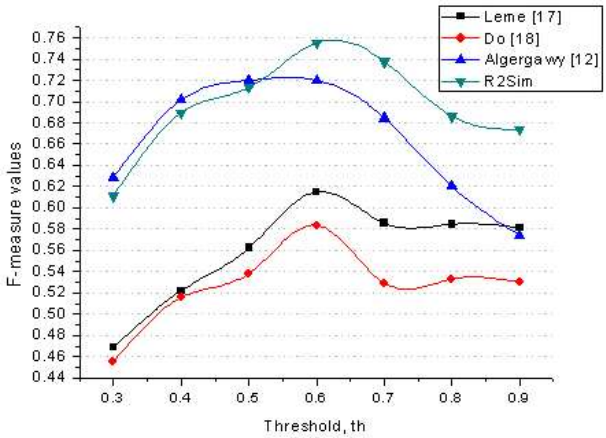

Fig.7. F-measure among R2Sim and related work

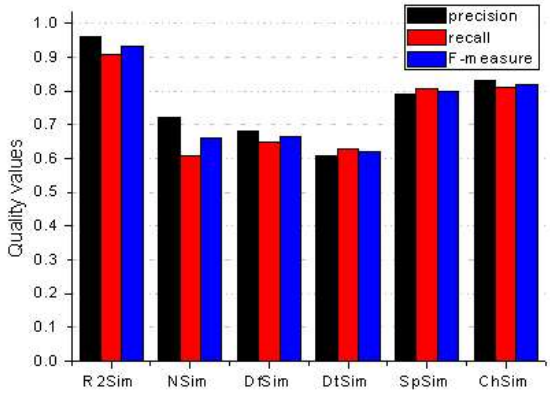

Fig.8. Quality of R2Sim, NSim, DfSim, DtSim, SpSim, and ChSim

The comparison results in Fig. 5, 6, and 7 show that our R2Sim significantly outperforms the other methods at all thresholds, followed by the methods of Algergawy, Leme, and Do. The Algergawy's method outperforms the R2Sim when the thresholds are equal or less than *0.5*. The main reason for this is that the data type similarity values of Algergawy'method are very high and based on user's judgment. However, for high threshold values, Algergawy's method has less accurate similarity values. The measures of Do and Leme have poor results since they are simply based on the string similarity of element names, but Leme's method is better than Do's method since Leme approach still considers the data type similarity.

Further, in order to determine the most important factor that affects the similarity values, we separate five similarity factors (*NSim, DfSim, DtSim, SpSim,* and *ChSim*) and compare with the whole combination of them (*R2Sim*). The result is presented in Fig. 8.

The columns in Fig. 8 show that *DtSim* has the lowest measure quality. Its F-measure values is only 62% in comparing with 66% of *NSim* and *DfSim*, about 80% of *SpSim* and *ChSim*. The reason is that the data type similarity measure is mostly applied for property elements whereas the number of class elements in RDFS is very high, so only *DtSim* cannot differentiate the semantic similarity of RDF elements. Among five measuring factors, *ChSim* gives the highest similarity value. However, regarding the best quality achieved, we observe that the combination of all similarity factors outperforms *ChSim*. Therefore, it is better to use multiple similarity measures instead of using a single measure.

VI. CONCLUSIONS

This paper proposes a novel similarity measuring technique for RDF elements. We present a semantic similarity measurement method that computes both description and neighborhood resemblances. The experimental evaluation demonstrates that our method outperforms the human judgment and related approaches, especially our approach gets best result when processing complex RDF documents. Further, the combination of all measuring factors provides important information for deriving the correct similarity values.

We hope that the research has established a foundation to help the integration of different RDF Schemas. If this method is popularized, a large amount of RDF Schema data on the current Web will be integrated into the useful

ontology for the Semantic Web and its applications. Our future research will focus on computing the similarity of RDF individuals based on the RDF Schema's relatedness.

REFERENCES

[1] Frank Manola, Eric Miller, *W3C*, 2004, http://www.w3.org/TR/rdf-primer/#rdfschema

[2] Doan A. H., Madhavan J., Domingos P., Ontologies Matching: A Machine Learning Approach, *Handbook on Ontologies in Inf. Systems*, Springer-Velag, 2003.

[3] Ehrig M., Sure Y., Ontology Mapping – an integrated approach, *1st European Semantic web Symposium*, 2004.

[4] Oundhankar S., K. Verma, Sivashanugam K., Discovery of web services in a Multi-Ontologies and Federated Registry Environment, *International Journal of Web Services Research*, 1, 3, 2005.

[5] Ronald M., Thomas H., Rene P., Enterprise Knowledge Infrastructures, 2nd edition, Springer, 2009.

[6] D Vint Productions, XML Schema - Data Types Quick Reference, http://www.xml.dvint.com, 2003.

[7] E. Pyshkin, A. Kuznetsov, Approaches for Web Search User Interfaces: How to improve the search quality for various types of information, *JoC*, Vol.1, No.1, pp.1-8.

[8] Roman Y Shtykh, Qun Jin, A human-centric integrated approach to web information search and sharing, *HCIS 2011*, 1:2 (22 November 2011)

[9] Vitaly Klyuev, Ai Yokoyama, Web Query Expansion: A Strategy Utilising Japanese WordNet, *JoC*, Vol.1, No.1, pp.23-28.

[10] Princeton University, WordNet_ A lexical database for English, http://wordnet.princeton.edu/wordnet

[11] R. Nayak, T. Tran, A progressive clustering algorithm to group the XML data by structural and semantic similarity", *Pattern Recognition & Artificial Intelligence* 21(4) (2007) 723-743.

[12] Alsayed Algergawy, Richi Nayak, Gunter Saake, Element similarity measures in XML schema matching, *Journal of Information Sciences*, pp. 4975-4998, 2010.

[13] Sergey Melnik, Bridging the gap between RDF and XML, 1999, http://www-db.stanford.edu/melnik/rdf/syntax.html

[14] M. Ferdinand, C. Zirpins, and D. Trastour, Lifting XML Schema to OWL, *Web Engineering – 4th International Conference, ICWE*, pp. 354-358, 2004.

[15] Tim Berners-Lee, A strawman unstriped syntax for RDF in XML, *W3C*, March 2007.

[16] Jonathan Boden, Simplified XML syntax for RDF, http://www.openhealth.org/RDF/RDFSurfaceSyntax.html

[17] Leme, L. A. P. P.; Casanova, M. A.; Breitman, K. K. & Furtado, A. L. Evaluation of similarity measures and heuristics for simple RDF schema matching. *Technical Report 44/08*, Dept. Informatics, PUC-Rio 14, 2008.

[18] Hong-Hai Do, and Erhard Rahm, COMA - A System for Flexible Combination of Schema Matching Approaches, *Proceedings of the Very Large Data Bases conference (VLDB)*, pp 610–621, 2002.

[19] Dongqiang D. Yang and David M.W. Powers, Measuring Semantic Similarity in the Taxonomy of WordNet, *The 28th Australasian Computer Science Conference (ACSC2005)*, Australia, pp. 315-322, 2005.

[20] A. Maganaraki and L. Sidirourgos, RDF Schema Registry, http://139.91.183.30:9090/RDF/Examples.html, 2004

[21] Troy Simpson, Crowe M., WordNet.Net http://opensource.ebswift.com/WordNet.Net, 2005

[22] Wikipedia, Precision and recall, http://en.wikipedia.org/wiki/Precision_and_recall

[23] Pham Thi Thu Thuy, Young-Koo Lee, and Sungyoung Lee, "R2Sim: A Novel Semantic Similarity Measure for Matching between RDF Schemas", The 2012 FTRA International Conference on Advanced IT, engineering and Management (FTRA AIM 2012), Seoul, Korea, February 6-8, 2012.

[24] Roberto De Virgilio, Antonio Maccioni, Riccardo Torlone, "A similarity measure for approximate querying over RDF data", EDBT'13 Proceedings of the Joint EDBT/ICDT 2013 Workshops, pp. 205-213, 2013.

[25] Samur Araujo, Jan Hidders, Daniel Schwabe, Arjen P. de Vries, "SERIMI – Resource Description Similarity, RDF Instance Matching and Interlinking", OM, volume 814 of CEUR Workshop Proceedings, CEUR-WS.org, 2011.

[26] Marcelo Schiessl, Rita Berardi, and Marisa Brascher, "Similarity between text and RDF", Information Services and Use, 34 (2014), 325-330.

[27] Mehwish Alam and Amedeo Napoli, "An Approach Towards Classifying and Navigating RDF data based on Pattern Structures", Proceedings of the International Workshop on Formal Concept Analysis and Applications 2015 co-located with 13th International Conference on Formal Concept Analysis., Jun 2015, Nerja, Spain. 1434, pp.33-48, 2015.

MỘT CẢI TIẾN TRONG VIỆC ĐO LƯỜNG ĐỘ TƯƠNG ĐỒNG NGỮ NGHĨA GIỮA CÁC TÀI LIỆU RDF

Phạm Thị Thu Thúy, Nguyễn Đăng Tiến

TÓM TẮT— *RDF hiện đang đóng vài trò quan trọng trong các ứng dụng tri thức bởi RDF cung cấp một lượng thuật ngữ cho phép mô tả chính xác dữ liệu. Tuy nhiên, sự lớn mạnh của RDF dẫn đến nhu cầu đánh giá sự giống nhau giữa các tài liệu có tính tương đồng. Bài báo này trình bày một cải tiến trong việc so sánh sự tương quan về ngữ nghĩa giữa các phần tử trong tài liệu RDF. Các công thức đo lường chú trọng đến thông tin được mô tả trong các phần tử RDF và mối quan hệ giữa các phần tử cha và con. Các công thức đề xuất được thực nghiệm bằng cách ánh xạ các tài liệu RDF với nhau để xác định số lượng tương quan và so sánh kết quả với nhận định khách quan của người dùng. Các kết quả thực nghiệm chỉ ra rằng phương pháp của chúng tôi cho kết quả độ tương tự chính xác hơn các phương pháp liên quan.*

Keywords— *Độ tương tự, tài liệu RDF, đo lường.*