Semantic Web Datatype Similarity: Towards Better RDF Document Matching

Irvin Dongo, Firas Al-Khalil, Richard Chbeir, Yudith Cardinale

Semantic Web Datatype Similarity: Towards Better RDF Document Matching

Irvin Dongo[1(✉)], Firas Al Khalil[2], Richard Chbeir[1], and Yudith Cardinale[1,3]

[1] University Pau & Pays Adour, LIUPPA, EA3000, 64600 Anglet, France
{irvin.dongo,richard.chbeir}@univ-pau.fr
[2] University College Cork, CRCTC, 13 South Mall, Cork, Ireland
firas.alkhalil@ucc.ie
[3] Departamento de Computación, Universidad Simón Bolívar, Caracas, Venezuela
ycardinale@usb.ve

Abstract. With the advance of the Semantic Web, the need to integrate and combine data from different sources has increased considerably. Many efforts have focused on RDF document matching. However, they present limited approaches in the context of datatype similarity. This paper addresses the issue of datatype similarity for the Semantic Web as a first step towards a better RDF document matching. We propose a datatype hierarchy, based on W3C's XSD datatype hierarchy, that better captures the *subsumption* relationship among primitive and derived datatypes. We also propose a new datatype similarity measure, that takes into consideration several aspects related to the new hierarchical relations between compared datatypes. Our experiments show that the new similarity measure, along with the new hierarchy, produces better results (closer to what a human expert would think about the similarity of compared datatypes) than the ones described in the literature.

Keywords: Datatype hierarchy · Datatype similarity · XML · XML Schema · Ontology · RDF · Semantic Web

1 Introduction

One of the benefits offered by the Semantic Web initiative is the increased support for data sharing and the description of real resources on the web, by defining standard data representation models such as RDF, the Resource Description Framework. The adoption of RDF increases the need to identify similar information (resources) in order to integrate and combine data from different sources (e.g., Linked Open Data integration, ontology matching). Many recent works have focused on describing the similarity between concepts, properties, and relations in the context of RDF document integration and combination [22].

Indeed, RDF describes resources as triples: $\langle \texttt{subject}, \texttt{predicate}, \texttt{object} \rangle$, where subjects, predicates, and objects are all resources identified by their IRIs[1].

[1] Internationalized Resource Identifier. An extension of URIs that allows characters from the Unicode character set.

Objects can also be literals (e.g., a number, a string), which can be annotated with optional type information, called a datatype; RDF adopts the datatypes from XML Schema. The W3C Recommendation proposed in [1] points out the importance of the existence of datatype annotations to detect entailments between objects that have the same datatype but a different value representation. For example, if we consider two distinct triples containing the objects `"20.000"` and `"20.0"`, then these objects are considered as different, because of the missing datatype. However, if they were annotated as follows: `"20.000"^^xml:decimal` and `"20.0"^^xml:decimal` then we can conclude that both objects are identical. Moreover, works on XML Schema matching proved that the presence of datatype information, constraints, and annotations on an object improves the similarity between two documents (up to 14%) [4].

Another W3C Recommendation [2] proposes a simple method to determine the similarity of two distinct datatypes: the similarity between two primitive datatypes is `0` (disjoint), while the similarity between two datatypes derived from the same primitive datatype is `1` (compatible). Obviously, this method is straightforward and does not capture the degree of similarity of datatypes; for instance, `float` is more similar to `int` than to `date`. This observation lead to the development of *compatibility tables*, that encodes the similarity ($\in [0, 1]$) of two datatypes. They were used in several studies [9,23] for XML Schema matching. These compatibility tables were either populated manually by a designated person, as in [9,23] or generated automatically using a similarity measure that relies on a hierarchical classification of datatypes, as in [15,28].

Hence, in the context of RDF document matching, these works present the following limitations:

1. The Disjoint/Compatible similarity method as proposed by the W3C is too restrictive, especially when similar objects can have different, yet related, datatypes (e.g., `float` and `int` *vs* `float` and `double`).
2. The use of a true similarity measure, expressed in a *compatibility table*, is very reasonable; however, we cannot rely on an arbitrary judgment of similarity as done in [9,23]; moreover, for 44 datatypes (primitive and derived ones, according to W3C hierarchy), there are 946 similarity values ($n \times (n-1)/2$, n $=$ 44), which makes the *compatibility table* incomplete as in [9]; a similarity measure that relies on a hierarchical relation of datatypes is needed.
3. The W3C datatype hierarchy, used in other works, does not properly capture any semantically meaningful relationship between datatypes (see, for instance, how datatypes related to `dateTime` and `time` are flattened in Fig. 2).

From these limitations, there is a need to provide a better solution for any RDF document matching approach, where simple datatype similarity is considered. To achieve this, we propose:

1. An extended version of the W3C datatype hierarchy, where a parent-child relationship expresses subsumption (parent subsumes child), which makes it a taxonomy of datatypes.

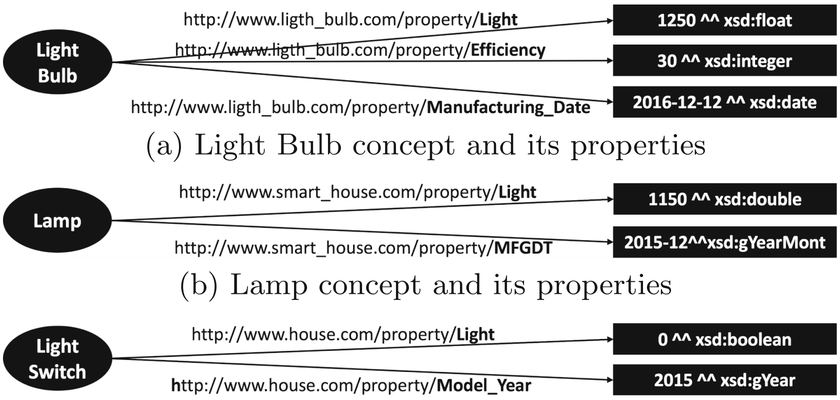
2. A new similarity measure: extending the one presented in [15], to take into account several aspects related to the new hierarchical relations between compared datatypes (e.g., children, depth of datatypes).

We experimentally compare the effectiveness of our proposal (datatype hierarchy and similarity measure) against existing related works. Our approach produces better results (closer to what a human expert would think about the similarity of compared datatypes) than the ones described in the literature.

The paper is organized as follows. In Sect. 2, we present a motivating scenario. In Sect. 3, we survey the literature on datatype similarity and compare them using our motivating scenario. In Sect. 4, we describe the new datatype hierarchy and the new similarity measure. In Sect. 5, we present the experiments we performed. And finally, we conclude in Sect. 6.

2 Motivating Scenario

In order to illustrate the limitations of existing approaches for datatype similarity, we consider a scenario in which we need to integrate three RDF documents with similar concepts (resources) but based on different vocabularies. Fig. 1 shows three concepts from three different RDF documents to be integrate. Figure 1a describes the concept of a `Light Bulb` with properties (predicates) `Light`, `Efficiency`, and `Manufacturing_Date`, Fig. 1b describes the concept of `Lamp` with properties `Light` and `MFGDT` (manufacturing date), and Fig. 1c shows the concept of `Light Switch` with properties `Light` and `Model_Year`.


Light Bulb
http://www.ligth_bulb.com/property/Light
http://www.ligth_bulb.com/property/Efficiency
http://www.ligth_bulb.com/property/Manufacturing_Date
1250 ^^ xsd:float
30 ^^ xsd:integer
2016-12-12 ^^ xsd:date
Lamp
http://www.smart_house.com/property/Light
http://www.smart_house.com/property/MFGDT
1150 ^^ xsd:double
2015-12^^xsd:gYearMont
Light Switch
http://www.house.com/property/Light
http://www.house.com/property/Model_Year
0 ^^ xsd:boolean
2015 ^^ xsd:gYear

(a) Light Bulb concept and its properties

(b) Lamp concept and its properties

(c) Light Switch concept and its properties

Fig. 1. Three concepts from three different RDF documents

To integrate these RDF documents, it is necessary to determine the similarity of the concepts expressed in them, based on the similarity of their properties. More precisely, we can determine the similarity of two properties by inspecting the datatypes of their *ranges*[2] (i.e., of their objects).

[2] A range (*rdfs:range*) defines the object type that is associated to a property.

Intuitively, considering the datatype information, we can say that:

1. `Light Bulb` and `Lamp` are similar, since their properties are similar: the `Light` property is of type `float` for `Light Bulb` and `double` for `Lamp`, we know that both `float` and `double` express floating points, and they differ only by their precisions; the same thing can be said about the properties `Manufacturing_Date` and `MFGDT`.
2. `Light Switch` is different from the other concepts; indeed, the `Light` property is expressed in `binary`, and can hold one of two values, namely 0 and 1, expressing the state of the light switch (i.e., on and off, respectively).

Hence, to support automatic matching of RDF documents based on their concepts similarity, it is necessary to have a datatype hierarchy establishing semantically meaningful relationship among datatypes and a measure able to extract these relations from the hierarchy. In the following section, we survey the literature on datatype similarity and compare them using this motivating scenario.

3 Related Work

To the best of our knowledge, there is no existing work tackling datatype similarity specifically targeting RDF documents. Hence, we review works on datatype similarity described for XML and XSD, since RDF uses the same XML datatypes proposed by the W3C (the datatype hierarchy is shown in Fig. 2), and we also consider works in the context of ontology matching. We evaluate these works in an RDF document matching/integration scenario in the discussion.

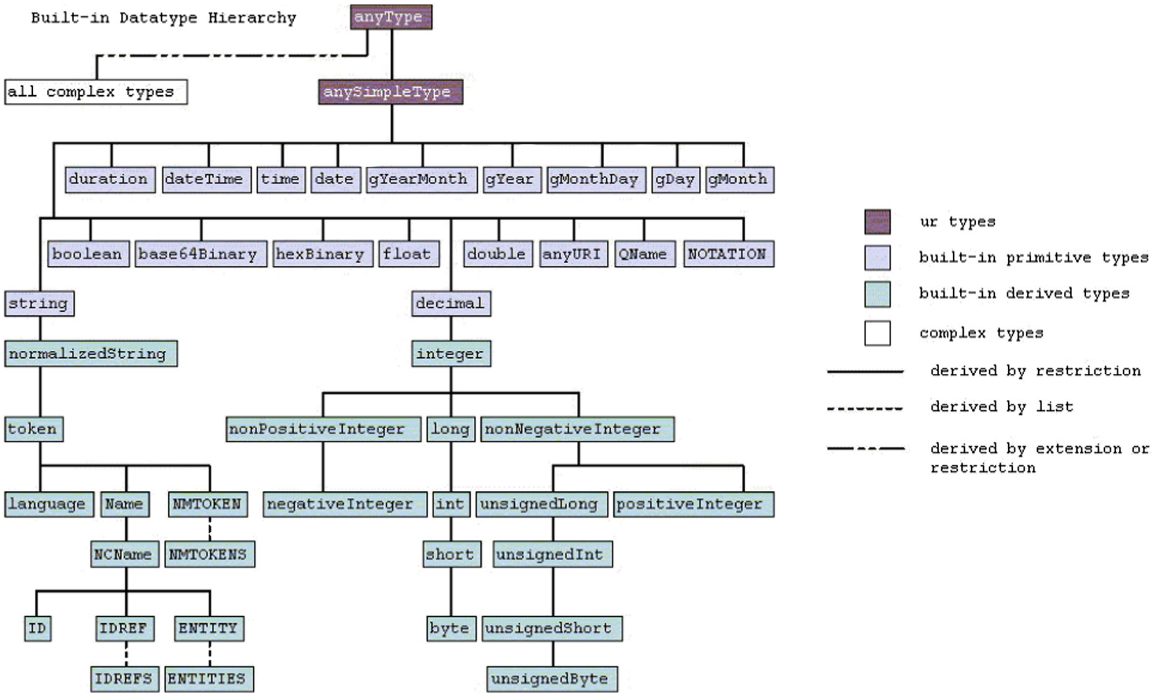


Fig. 2. W3C datatype hierarchy

Most of the existing works in the XML and XSD area are focused on schema matching in contexts of, for example, XML message mapping, web data sources integration, and data warehouse loading. The main approaches taken to establish the datatype similarity are either: 1. based on user-defined compatibility tables [5–7,9,11,23,24,27], or 2. constraining facets[3] [28], or 3. extended W3C hierarchy and measure [3,8,15].

User-defined compatibility tables, as the one presented in Table 1 (taken from [9]), express the judgment and perception of users regarding the similarity between each pair of datatypes. Hence, these tables present similarity values that are not objective, complete, or reliable.

Table 1. Datatype compatibility table of work [9]

Type (s)	Type (t)	Compatibility coefficient (s, t)
string	string	1.0
string	date	0.2
decimal	float	0.8
float	float	1.0
float	integer	0.9
integer	short	0.8

When constraining facets are considered as in [28], the similarity value between two different datatypes is calculated by the number of common facets divided by the union of them. For example, datatypes `date` and `gYearMonth` have the same facets (i.e., pattern, enumeration, whiteSpace, maxInclusive, maxExclusive, minExclusive, and minInclusive), thus, their similarity is equal to 1. This method allows to create an objective, complete, and reliable *compatibility table*; however, suitability is still missing: besides facets, which are only syntactic restrictions, other information should be considered for the Semantic Web (e.g., common datatypes attributes[4] – datatype subsumption).

Other works have proposed a new datatype hierarchy by extending the one proposed by the W3C. In [15], the author proposes five new datatype groups: `Text, Calendar, Logic, Numeric`, and `Other`. They also propose a new datatype similarity function that relies on that hierarchy and takes into account the proximity of nodes to the root and the level of the Least Common Subsumer[5] (LCS) of the two compared datatypes. The works presented in [3,8], combine semantic similarity, structural similarity, and datatype compatibility

[3] Constraining facets are sets of aspects that can be used to constrain the values of simple types (https://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#rf-facets).

[4] An attribute is the minimum classification of data, which does not subsume another one. For example, datatype `date` has the attributes year, month, and day.

[5] It is the most specific common ancestor of two concepts/nodes, found in a given taxonomy/hierarchy.

of XML schemas in a function, by using the hierarchy and similarity function proposed by [15]. Even though these works improve the similarity values, we will see their limitations in the context of our motivational scenario, concerning to misdefined datatype relations in the datatype hierarchy.

In the context of ontology matching, most of the works classify datatypes as either Disjoint or Compatible (similarity $\in \{0, 1\}$). Some of them are based on the W3C hierarchy, such as [13,17], while others take into account properties of the datatypes (domain, range, etc.) [10,14,16,19–22,25,26]. When domain and range properties are considered, if two datatypes have the same properties, the similarity value is 1, otherwise it is 0. In the context of RDF matching, in which similar objects can have different but related datatypes, this binary similarity is too restrictive. The authors in [12] generate a vector space for each ontology by extracting all distinct concepts, properties and the ranges of datatype properties. To calculate the similarity between the two vectors, they use the cosine similarity measure. However, as the measure proposed in [15], the problem remains in the datatype hierarchy that does not represent more semantically meaningful relationships between datatypes.

Table 2. Related work classification

Group	Work	Datatype similarity	Datatype requirements			
			Simple datatype	Common attributes	SW context	
					XML/XSD	RDF/OWL
1	W3C [2,10,12–14,16,17,19–22,25,26]	Disjoint/compatible (binary values)	✓	X	✓	✓
2	[4–7,9,11,23,24,27]	User-defined compatibility table	✓	X	✓	X
3	[28]	Constraining facets	✓	X	✓	X
4	[3,8,15]	Formula on extended W3C hierarchy	✓	X	✓	X

We classify the existing works into four groups (see Table 2) and we evaluate them in our motivating scenario in the upcoming section.

Resolving Motivating Scenario and Discussion: Now, we evaluate our scenario using the defined groups in Table 2. We have the datatypes `float` and `date` from the concept `Light Bulb` (Fig. 1a), datatypes `double` and `gYearMonth` from the concept `Lamp` (Fig. 1b), and `boolean` and `gYear` from concept `Light Switch` (Fig. 1c).

According to the Disjoint/Compatible similarity, either defined by the W3C or not (Group 1 in Table 2), the similarity between the three pairs of datatypes related to `Light` property (`float`–`double`, `float`–`boolean`, and `double`–`boolean`) is 0, because the three datatypes are primitives. We have the same similarity result regarding `Manufacturing_Date`, `MFGDT`, `Model_Year` properties, since their datatypes are also primitives. It means that there is no possible integration for these concepts using this similarity method. However, the concepts `Light Bulb` and `Lamp` are strongly related according to our scenario.

Based on the user-defined compatibility table shown in Table 1 (as works in Group 2 do), the similarity between `float`–`double` is a given constant > 0 (as `decimal`–`float` has in the compatibility table), however the similarity values of `double`–`boolean`, `date`–`gYearMonth`, `date`–`gYear`, and `gYearMonth`–`gYear` are not present in the compatibility table, therefore leading to a similarity value of 0 as in [15] do. In this case, concepts `Light Bulb` and `Lamp` have their respective properties `Light` considered similar, while `Manufacturing_Date` and `MFGDT` are considered disjoint, even though they are clearly related.

According to the methods of Group 3 (based on constraining facets), similarity values for `float`–`double`, `date`–`gYearMonth`, `date`–`gYear`, and `gYearMont`–`gYear` are all equal to 1 (because they have the same facets), and for `float`–`boolean` and `double`–`boolean`, the similarities are equal to 0.29 (2 common facets divided by the union of them, which is 7). Thus, the three concepts can be integrated as similar, which is incorrect. Additionally, datatypes `date`, `gYearMonth`, and `gYear` are related but not equal: besides their facets, other information (such as datatype attributes - year, month, day) should count to decide about their similarities.

Finally, according to the works in Group 4, which are based on similarity measures applied on a datatype hierarchy extended from the W3C hierarchy [15], similarity between `float`–`double` is 0.30, similarity between `float`–`boolean` and `double -boolean` is 0.09, for `date`–`gYearMonth`, `date`–`gYear`, and `gYearMonth`–`gYear` the similarity value is 0.296[6]. Even though these works manage in a better way the datatype similarity than all other Groups, there is still the issue of considering common datatypes attributes (as for work in Group 3). We can note that `date`–`gYearMonth` share year and month as common attributes, while `date`–`gYear` only have year as common attribute; thus, similarity between `date`–`gYearMonth` should be bigger than the other.

Table 3. Integration results for our motivating scenario

Concept integration	G. 1 (Sim)	G. 2 (Sim)	G. 3 (Sim)	G. 4 (Sim)	Appropriate
`Light Bulb` and `Lamp`	NI (0.00)	NI (0.40)	I (1.00)	NI (0.30)	**I**
`Lamp` and `Light Switch`	NI (0.00)	NI (0.00)	I (0.65)	NI (0.19)	**NI**
`Light Bulb` and `Light Switch`	NI (0.00)	NI (0.00)	I (0.65)	NI (0.19)	**NI**

Results were obtained by applying a threshold 0.50 for average of properties; NI = Not Integrable, I = Integrable

Table 3 summarizes the integration results of the motivating scenario. Column *Appropriate* shows the correct integration according to our intuition. One can note that existing works cannot properly determine a correct integration.

[6] We show the results according the measure proposed on [15], all other works in Group 4 propose similar measures.

With this analysis, we can observe the importance of datatypes for data integration and the limitations of the existing works, from which, the following requirements were identified:

1. The measure should consider at least all simple datatypes (primitive and derived datatypes); complex datatypes are out of the scope in this work.
2. The datatype hierarchy and similarity measure should consider common datatype attributes (subsumption relation) in order to establish a more appropriate similarity.
3. The whole approach should be objective, complete, reliable, and suitable for the Semantic Web.

We can note from Table 2, that all works consider primitive and derived datatypes and are suitable in XML and XSD contexts. Only the works in the context of ontology matching (Group 1) consider RDF data. None of these works consider common datatype attributes. The following section describes our approach, based on a new hierarchy and a new similarity measure, that overcomes the limitations of existing works and addresses these requirements.

4 Our Proposal

In this section, we describe our datatype similarity approach that mainly relies on an extended W3C datatype hierarchy and a new similarity measure.

4.1 New Datatype Hierarchy

As we mentioned before, the W3C datatype hierarchy does not properly capture any semantically meaningful relationship between datatypes and their common attributes. This issue is clearly identified in all datatypes related to date and time (e.g., `dateTime`, `date`, `time`, `gYearMonth`), which are treated as isolated datatypes in the hierarchy (see Fig. 2).

Our proposed datatype hierarchy extends the W3C hierarchy as it is shown in Fig. 3. White squares represent our new datatypes, black squares represent original W3C datatypes, and gray squares represent W3C datatypes that have changed their location in the hierarchy. We propose four new primitive datatypes: `period`, `numeric`, `logic`, and `binary`. Thus, we organize datatypes into eight more coherent groups of primitive datatypes (`string`, `period`, `numeric`, `logic`, `binary`, `anyURI`, `QName`, and `NOTATION`). All other datatypes are considered as derived datatypes (e.g., `duration`, `dateTime`, `time`) because their attributes are part of one particular primitive datatype defined into the eight groups.

We also add two new derived datatypes (`yearMonthDuration` and `dayTimeDuration`), which are recommended by W3C to increase the precision of `duration`, useful for `XPath` and `XQuery`. We classify each derived datatype under one of the eight groups (e.g., `Period` subsumes `duration`, `numeric` subsumes `decimal`) and, in each group, we specify the proximity of datatypes by a sub-hierarchy (e.g., `date` is closer to `gYearMonth` than to `gYear`).

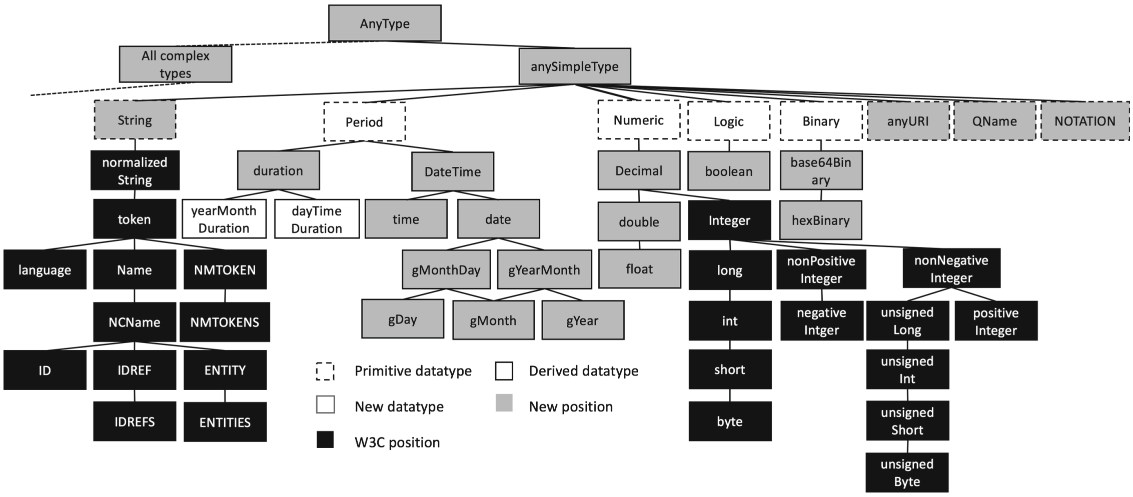


Fig. 3. New datatype hierarchy

The distribution of the hierarchy for derived datatypes is established based on the subsumption relation and stated in the following assumption:

Assumption 1. *If a datatype d_1 contains at least all the attributes of a datatype d_2 and more, d_1 is more general than d_2 (d_1 subsumes d_2).*

As a consequence of Assumption 1, the hierarchy designates datatypes more general to more specific, from the root to the bottom, which in turn defines datatypes more related than others according to their depths in the hierarchy. With regards to this scenario, we have the following assumption:

Assumption 2. *Datatypes in the top of the hierarchy are less related than datatypes in the bottom, because datatypes in the top are more general than the ones in the bottom.*

Thus, according to Assumption 2, the datatype similarity value will depend on their position (depth) in the hierarchy (e.g., `gYearMonth`–`gYear` are more similar than `period`–`dateTime`), as we show in the next section.

4.2 Similarity Measure

Our proposed similarity measure is inspired by the one presented in [15]. The authors establish the similarity function based on the following intuition:

> *"The similarity between two datatype d_1 and d_2 is related to the distance separating them and their depths in the datatype hierarchy. The bigger the distance separating them, the less similar they are. The deeper they are the more similar they are, since at deeper levels, the difference between nodes is less significant [15]."*

The authors state the similarity between two datatypes d_1 and d_2 as:

$$c(d_1, d_2) = \begin{cases} f(l) \times g(h) & \text{if } d_1 \neq d_2 \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

where:

- l is the shortest path length between d_1 and d_2;
- h is the depth of the Least Common Subsumer (LCS) datatype which subsumes datatype d_1 and d_2.
- $f(l)$ and $g(h)$ are defined based on Shepard's universal law of generalization [18] in Eqs. 2 and 3, respectively.

$$f(l) = e^{-\beta l} \tag{2}$$

$$g(h) = \frac{e^{\alpha h} - e^{-\alpha h}}{e^{\alpha h} + e^{-\alpha h}} \tag{3}$$

where α and β are user-defined parameters.

The work in [15] does not analyze the common attributes (children) of compared datatypes. For example, the datatype pair `date`–`gYearMonth` (with 2 attributes, namely `year` and `month`, in common) involves more attributes than `date`–`gYear` (with only 1 attribute, namely `year`, in common). The authors of [15] consider that the similarity values of both cases are exactly the same.

In order to consider this analysis, we assume that:

Assumption 3. *Two datatypes d_1 and d_2 are more similar if their children in the datatype hierarchy are more similar.*

Furthermore, the depth of the LCS is not enough to calculate the similarity according to Assumption 2. Notice that the difference in levels in the hierarchy is also related to similarity. For example, according to [15], we have $c($`time`, `gYearMonth`$) = c($`dateTime`, `gYear`$)$, because in both cases the distance between the datatypes is $l = 3$, and the LCS is `dateTime`, whose $h = 3$ (see Fig. 3). However, the difference between levels of `time` and `gYearMonth` is smaller than the one of `dateTime` and `gYear`, thus the similarity of `time-gYearMonth` should be bigger than the second pair (i.e., $c($`time`, `gYearMonth`$) > c($`dateTime`, `gYear`$)$). Hence, we assume:

Assumption 4. *The similarity of two datatypes d_1 and d_2 is inversely proportional to the difference between their levels.*

Based on Assumptions 3 and 4, we defined the cross-children similarity measure in the following.

Let V_{d_1p,d_2q} be the children similarity vector of a datatype d_1, with respect to datatype d_2 in levels p and q, respectively. In d_1 sub-hierarchy, d_1 has i children in level p and in d_2 sub-hierarchy, d_2 has j children in level q. Thus, V_{d_1p,d_2q} is calculated as in Eq. 4.

$$V_{d_1p,d_2q} = [c(d_1, d_{1p}^1), \ldots, c(d_1, d_{1p}^i), c(d_1, d_{2q}^1), \ldots, c(d_1, d_{2q}^j)] \tag{4}$$

where d_{1p}^{x} represents the child x of d_1 (with x from 1 to i) in level p and d_{2q}^{y} represents the child y (with y from 1 to j) of d_2 in level q.

Similarly, let V_{d_2q,d_1p} be the children similarity vector of a datatype d_2, with respect to datatype d_1 in the levels q and p respectively, defined as in Eq. 5.

$$V_{d_2q,d_1p} = [c(d_2, d_{1p}^{1}), \ldots, c(d_2, d_{1p}^{i}), c(d_2, d_{2q}^{1}), \ldots, c(d_2, d_{2q}^{j})] \tag{5}$$

For each pair of vectors V_{d_1p,d_2q} and V_{d_2q,d_1p}, we formally define the cross-children similarity for level p and q, in Definition 1.

Definition 1. *The cross-children similarity of two datatypes d_1 and d_2 for levels p and q, respectively, is the cosine similarity of their children similarity vectors V_{d_1p,d_2q} and V_{d_2q,d_1p}, calculated as:*

$$CCS_{d1p,d2q} = \frac{V_{d_1p,d_2q} \cdot V_{d_2q,d_1p}}{\|V_{d_1p,d_2q}\| \|V_{d_2q,d_1p}\|}$$

Now, considering all pairs of V (i.e., all levels of both sub-hierarchies), we define the total cross-children similarity between d_1 and d_2 in Definition 2.

Definition 2. *The total cross-children similarity of two datatypes d_1 and d_2 is calculated as:*

$$S(d_1, d_2) = \frac{1}{L_1} \times \sum\nolimits_{p=1}^{L_1} \sum\nolimits_{q=1}^{L_2} m(d1p, d2q) \times CCS_{d1p,d2q}$$

where $m(d1p, d2q)$ is a Gaussian function based on Assumption 4: L_1 and L_2 are the number of levels of sub-hierarchies of d_1 and d_2, respectively.

The Gaussian function is defined as follows:

$$m(d1p, d2q) = e^{-\pi \times (\frac{(depth(d1p) - depth(d2q))}{H-1})^2}$$

where $depth(d_{1p})$ and $depth(d_{2q})$ are the depths of the levels p and q respectively. H is the maximum depth of the hierarchy. Note that the depth of the hierarchy starts from 0. We name $S'(d_1, d_2)$ the average between $S(d_1, d_2)$ and $S(d_2, d_1)$.

$$S'(d_1, d_2) = 0.5 \times S(d_1, d_2) + 0.5 \times S(d_2, d_1) \tag{6}$$

Finally, we define similarity between datatypes d_1 and d_2 in Definition 3 as an extension of Eq. 1.

Definition 3. *Similarity between two datatypes d_1 and d_2, denoted as $sim(d_1, d_2)$, is determined as:*

$$sim(d_1, d_2) = \begin{cases} (1-\omega) \times f(l) \times g(h) + \omega \times S'(d_1, d_2) & \text{if } d_1 \neq d_2 \\ 1 & \text{otherwise} \end{cases}$$

where $\omega \in [0, 1]$ is a user-defined parameter that indicates the weight to be assigned to the cross-children similarity.

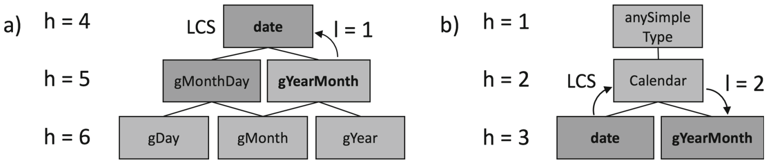


Fig. 4. (a) Sub-hierarchy from our new hierarchy; (b) sub-hierarchy from [15]

With our RDF similarity approach, we satisfy all identified requirements. This measure generates similarity values based on a hierarchy (objective, complete and reliable) for simple datatypes. The whole approach is more suitable for the Semantic Web, because common attributes among datatypes are taking into account both in the hierarchy by Assumption 1 and in the similarity measure by Definition 1.

The following section illustrates how our approach is applied to calculate similarity between the properties of the concepts `Light Bulb` and `Lamp` from our motivating scenario and, it is compared with the work in [15].

4.3 Illustrative Example

To better understand our similarity approach, we illustrate step by step the process to obtain the similarity between datatypes `date` from `Light Bulb` and `gYearMonth` from `Lamp`. We compare it with the one obtained by [15]. To do so, we fix the parameters with the following values: $\alpha = \beta = 0.3057$ (taken from [15]), and $\omega = 0.20$, which means a weigh of 20% for cross-children similarity and 80% for the distance between datatypes and their depths (i.e., $f(l)$ and $g(h)$).

According to our new datatype hierarchy, we have $l = 1$, as the distance between `date-gYearMonth`, and $h = 4$ the depth of `date`, which is the LCS. Figure 4(a) shows these values and the sub-hierarchy from the LCS, according to our new hierarchy. For [15], the distance between `date-gYearMonth` is $l = 2$ and $h = 2$ is the depth of the LCS, which is `Calendar`. Figure 4(b) shows these values and the sub-hierarchy, according to the hierarchy in [15].

Then, the similarity value for our similarity approach is (see Definition 3):

$$sim(\texttt{date}, \texttt{gYearMonth}) = 0.80 \times f(1) \times g(4) + 0.20 \times S'(\texttt{date}, \texttt{gYearMonth})$$

and for [15] is (see Eq. 1): $c(\texttt{date}, \texttt{gYearMonth}) = f(2) \times g(2)$.

According to Eqs. 2 and 3, $f(1) = 0.74$, $g(4) = 0.84$ (for our similarity approach) and $f(2) = 0.54$, $g(2) = 0.55$ (for [15]). Hence, for [15] the similarity value between `date-gYearMonth` is: $c(\texttt{date}, \texttt{gYearMonth}) = 0.297$.

For our similarity approach, the cross-children similarity is taken into account to finally calculate the similarity between `date-gYearMonth` (see Eq. 6):

$$S'(\texttt{date}, \texttt{gYearMonth}) = 0.5 \times S(\texttt{date}, \texttt{gYearMonth}) + 0.5 \times S(\texttt{gYearMonth}, \texttt{date})$$

To calculate $S'(\texttt{date}, \texttt{gYearMonth})$, we have to calculate before the total cross-children similarities, S(`date`,`gYearMonth`) and S(`gYearMonth`,`date`). From

Def. 2, we obtain:

$$S(\texttt{date}, \texttt{gYearMonth}) = \frac{1}{2} \times \sum_{p=1}^{2} \sum_{q=1}^{1} e^{-\pi \times (\frac{(depth(d1p) - depth(d2q))}{9-1})^2} \times CCS_{\texttt{date}p,\texttt{gYearMonth}q}$$

Note that `date` has two levels of children (thus, $p = 1$ to 2 in the sum), while `gYearMonth` has one level of children (thus, $q = 1$ to 1 in its sum). Replacing values, we have $S(\texttt{date}, \texttt{gYearMonth}) = 0.945$. An equivalent process is done to calculate $S(\texttt{gYearMonth}, \texttt{date}) = 0.978$. Now, we replace the obtained values in the equation: $S'(\texttt{date}, \texttt{gYearMonth}) = 0.5 \times 0.945 + 0.5 \times 0.978 = 0.961$.

The $S'(\texttt{date}, \texttt{gYearMonth})$ is replaced by the respective value in the similarity equation to finally have: $sim(\texttt{date}, \texttt{gYearMonth}) = 0.497 + 0.20 \times 0.961 = 0.688$.

Table 4. Datatypes similarity using the proposal of [15] and our approach

Datatype$_1$	*Datatype*$_2$	Similarity value [15]	Our similarity value
date	gYearMonth	0.30	0.69
date	gYear	0.30	0.46
dateTime	duration	0.30	0.37
dateTime	time	0.30	0.53
dateTime	gDay	0.30	0.29
decimal	float	0.30	0.39
double	float	0.30	0.62

Using our approach, the similarity value between `date-gYearMonth` has increased from 0.30 (according [15]) to 0.69. Table 4 compares our approach and [15], with other pairs of datatypes and their respective similarity values. Note that datatypes with attributes in common (e.g., `dateTime` and `time` have in common *time*) have greater similarity value than the ones obtained by [15]. Next section evaluates the accuracy of our approach.

5 Experiments

In order to evaluate our approach, we adopted the experimental set of datatypes proposed in [15], since there is not a benchmark available in the literature for datatype similarity. This set has 20 pairs of datatypes taken from the W3C hierarchy. These pairs were chosen according to three criteria: (i) same branch but at different depth levels (e.g., `int-long`); (ii) different branches with different depth levels (e.g., `string-int`); and (iii) identical pairs (e.g., `int-int`).

In [15], the authors used the human perception as reference values for the 20 pairs. The closer their similarity measure is to the human perception, the better the measure performs. We used the *Human Average* similarity values presented by [15] to benchmark our approach and a new *Human Average-2* dataset that we obtained by surveying 80 persons that have under- and pots-graduate degrees in computer science[3]. We also compared our work with the similarity values obtained from the compatibility table found in [9,28], and with the disjoint/compatible similarity from W3C.

Table 5. Experimental results: for the first and second experiments

Datatype 1	Datatype 2	Work [9] (Cupic)	Work [28]	W3C	Work [15]	Measure [15] + our hierarchy	Our Mea. + our hierarchy	H. Avg. from [15]	Our H. Avg-2
string	normalizedString	0.00	1.00	1.00	0.53	**0.40**	**0.47**	0.27	0.77
string	NCName	0.00	1.00	1.00	0.21	**0.16**	**0.29**	0.11	0.55
string	hexBinary	0.50	1.00	0.00	0.09	**0.09**	**0.09**	0.36	0.23
string	int	0.40	0.25	0.00	0.03	**0.05**	**0.08**	0.28	0.13
token	boolean	0.00	0.17	0.00	0.05	**0.05**	**0.05**	0.37	0.15
dateTime	time	0.90	1.00	0.00	0.30	**0.53**	**0.53**	0.70	0.71
boolean	time	0.00	0.58	0.00	0.09	**0.06**	**0.06**	0.04	0.13
int	byte	0.00	1.00	1.00	0.52	**0.52**	**0.52**	0.71	0.58
int	long	0.00	1.00	1.00	0.67	**0.67**	**0.73**	0.79	0.72
int	decimal	0.00	1.00	0.00	0.29	**0.29**	**0.38**	0.59	0.55
int	double	0.00	0.83	0.00	0.12	**0.21**	**0.23**	0.51	0.50
decimal	double	0.00	0.83	0.00	0.30	**0.53**	**0.55**	0.60	0.72
byte	positiveInteger	0.00	1.00	1.00	0.13	**0.13**	**0.13**	0.57	0.49
gYear	gYearMonth	0.00	1.00	0.00	0.30	**0.67**	**0.67**	0.65	0.65
int	int	1.00	1.00	1.00	1.00	**1.00**	**1.00**	1.00	1.00
string	byte	0.00	0.25	0.00	0.02	**0.03**	**0.03**	0.34	0.21
token	byte	0.00	0.25	0.00	0.01	**0.01**	**0.01**	0.46	0.24
float	double	0.00	1.00	0.00	0.30	**0.62**	**0.62**	0.60	0.75
float	int	0.00	0.83	0.00	0.12	**0.16**	**0.16**	0.46	0.47
gYear	negativeInteger	0.00	0.83	0.00	0.03	**0.01**	**0.01**	0.02	0.10
CC. wrt. H. Avg [15]		40.33%	38.32%	27.45%	69.45%	**80.21%**	**77.15%**	100.0%	-
CC. wrt. our H. Avg-2		29.48%	70.69%	51.09%	83.93%	**90.23%**	**92.39%**	-	100.0%

To compare how close are the similarity values to the human perception, we calculate the correlation coefficient (CC) of every work (i.e., [9,15,28], and our approach) with respect to *Human Average* and *Human Average-2*. A higher CC shows that the approach is closer to the human perception (*Human Average* and *Human Average-2*), and viceversa. The CC is calculated as follows:

$$CC = \frac{1}{n-1} \sum\nolimits_{i=1}^{n} \frac{(x_i - \bar{x})}{\sigma_x} \frac{(y_i - \bar{y})}{\sigma_y}$$

where n is the number of datatype pairs to compare ($n = 20$ in this case), x_i is the similarity value between datatype pair i, and y_i is its respective human average value, $\bar{x}$ and $\bar{y}$ are averages, and σ_x and σ_y are standard deviations with respect to all similarity values x and all human average values y. Results are shown in Table 5.

Since the similarity measures for work [15] and our work depend on the values of α and β, we evaluate the results under different assignments of α and β. To that end, we devised four experiments:

[3] Results are available: http://cloud.sigappfr.org/index.php/s/yRRbUQUeHs0NJnW.

1. In the first experiment, we fix $\alpha = \beta = 0.3057$ as chosen by [15], which they report to be the optimal value obtained by experimentation. We calculated the similarity values as in Eq. 1 to: (i) the W3C extended hierarchy [15] (column 6 in Table 5); and (ii) our proposed datatype hierarchy (column 7 in Table 5). We calculated the *CC* for both scenarios with respect to *Human Average* and *Human Average-2*. With this experiment, we evaluated the quality of our proposed datatype hierarchy.
2. In the second experiment, we fix $\alpha = \beta = 0.3057$ as chosen by [15], but instead of using their measure (Eq. 1), we used our cross-children similarity measure (see Def. 3) with our proposed datatype hierarchy (column 8 in Table 5). We fixed the $\omega = 0.20$[8]. With this experiment, we compared the quality of our approach against all other works.
3. In the third experiment, we chose values for α and β from the range $(0, 1]$, with a 0.02 step. In this case, 2010 possibilities were taken into account.
4. The fourth experiment is similar to the third one, except that a smaller step of 0.001 is considered. Therefore, there were 999181 possibilities.

As shown in Table 5, for experiments 1 and 2, we obtained a *CC* of 80.21% and 77.15% respectively, with respect to the *Human Average*. With respect to our *Human Average-2*, we obtained even better *CC* (90.23% and 92.39%).

In the third experiment, we obtained our best results for $\alpha = 0.20$ and $\beta = 0.02$, $CC = 82.60\%$ with respect to the *Human Average* (see Table 6(a), row 1). For $\alpha = 0.50$ and $\beta = 0.18$, $CC = 95.13\%$ with respect to our *Human Average-2* (see Table 6(a), row 2). In general, the similarity values generated by our work were closer to both human perception values than the other works (99.90% of the 2010 possible cases).

Similarly, for the fourth experiment, we obtained our best results for $\alpha = 0.208$ and $\beta = 0.034$ with a $CC = 82.76\%$ with respect to the *Human Average* of the work [15] (see Table 6(b), row 1). With respect to our *Human Average-2*, we obtained the best results for $\alpha = 0.476$ and $\beta = 0.165$, with a $CC = 95.26\%$ (see Table 6(b), row 2). In general, the similarity values generated by our work were closer to both human perceptions (99.97% of the 999181 possible cases).

Table 6. Results of the third and fourth experiments

	α	β	CC.
Human Average [15]	0.20	0.02	82,60%
Human Average-2	0.50	0.18	95,13%

(a) Third experiment with step = 0.02

	α	β	CC.
Human Average [15]	0.208	0.034	82.76%
Human Average-2	0.476	0.165	95.126%

(b) Forth experiment with step = 0.001

In conclusion, our approach outperforms all other works that we surveyed by considering a new hierarchy that captures a semantically more meaningful relation among datatypes, in addition to a measure based on cross-children similarity. Note that our work is not exclusive to RDF data; it can be also applied to XML data similarity and XSD/ontology matching.

[8] By experimentation, we determined this value as the optimal one.

6 Conclusions

In this paper, we investigated the issue of datatype similarity for the application of RDF matching/integration. In this context, we proposed a new simple datatype hierarchy aligned with the W3C hierarchy, containing additional types to cope with `XPath` and `XQuery` requirements in order to ensure an easy adoption by the community. Also, a new datatype similarity measure inspired by the work in [15], is proposed to take into account the cross-children similarity. We evaluated the new similarity measure experimentally. Our results show that our proposal presents a significant improvement, over the other works described in the literature.

We are currently working on extending and evaluating this work to include complex datatypes that can be defined for the Semantic Web. Also, we plan to evaluate the improvement of using our datatype similarity approach in existing matching tools [11,12].

Acknowledgments. This work has been partly supported by FINCyT/ INOVATE PERU - Convenio No. 104-FINCyT-BDE-2014.

References

1. RDF 1.1 Semantics, W3C Recommendation 25 February 2014. https://www.w3.org/TR/rdf11-mt/#literals-and-datatypes
2. XML Schema Datatypes in RDF and OWL, W3C Working Group Note 14 March 2006. https://www.w3.org/TR/swbp-xsch-datatypes/#sec-values
3. Al-Bakri, M., Fairbairn, D.: Assessing similarity matching for possible integration of feature classifications of geospatial data from official and informal sources. Int. J. Geogr. Inf. Sci. **26**(8), 1437–1456 (2012)
4. Algergawy, A., Nayak, R., Saake, G.: XML schema element similarity measures: a schema matching context. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2009. LNCS, vol. 5871, pp. 1246–1253. Springer, Heidelberg (2009). doi:10.1007/978-3-642-05151-7_36
5. Algergawy, A., Nayak, R., Saake, G.: Element similarity measures in xml schema matching. Inf. Sci. **180**(24), 4975–4998 (2010)
6. Algergawy, A., Schallehn, E., Saake, G.: A sequence-based ontology matching approach. In: Proceedings of European Conference on Artificial Intelligence, Workshop on Contexts and Ontologies, pp. 26–30 (2008)
7. Algergawy, A., Schallehn, E., Saake, G.: Improving XML schema matching performance using prufer sequences. Data Knowl. Eng. **68**(8), 728–747 (2009)
8. Amarintrarak, N., Runapongsa, S., Tongsima, S., Wiwatwattana, N.: SAXM: semi-automatic xml schema mapping. In: Proceedings of International Technical Conference on Circuits/Systems, Computers and Communications, pp. 374–377 (2009)
9. Bernstein, P.A., Madhavan, J., Rahm, E.: Generic schema matching with cupid. Technical report MSR-TR-2001-58, pp. 1–14. Microsoft Research(2001)
10. Cruz, I.F., Antonelli, F.P., Stroe, C.: Agreementmaker: efficient matching for large real-world schemas and ontologies. Proc. VLDB **2**(2), 1586–1589 (2009)
11. Do, H.-H., Rahm, E.: Coma: a system for flexible combination of schema matching approaches. In: Proceedings of VLDB, pp. 610–621 (2002)

12. Eidoon, Z., Yazdani, N., Oroumchian, F.: Ontology matching using vector space. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 472–481. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78646-7_45
13. Euzenat, J., Shvaiko, P. (eds.): Ontology Matching, vol. 18. Springer-Verlag New York Inc., New York (2007)
14. Hanif, M.S., Aono, M.: An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. J. Web Semant. **7**(4), 344–356 (2009)
15. Hong-Minh, T., Smith, D.: Hierarchical approach for datatype matching in xml schemas. In: 24th British National Conference on Databases, pp. 120–129 (2007)
16. Hu, W., Qu, Y., Cheng, G.: Matching large ontologies: a divide-and-conquer approach. Data Knowl. Eng. **67**(1), 140–160 (2008)
17. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Ontology matching with semantic verification. Web Semant. **7**(3), 235–251 (2009)
18. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of Conference on Research in Computational Linguistics, pp. 1–15 (1997)
19. Jiang, S., Lowd, D., Dou, D.: Ontology matching with knowledge rules. CoRR, abs/1507.03097 (2015)
20. Lambrix, P., Tan, H.: Sambo-a system for aligning and merging biomedical ontologies. Web Semant. **4**(3), 196–206 (2006)
21. Li, J., Tang, J., Li, Y., Luo, Q.: RiMOM: a dynamic multistrategy ontology alignment framework. Trans. Knowl. Data Eng. **21**(8), 1218–1232 (2009)
22. Mukkala, L., Arvo, J., Lehtonen, T., Knuutila, T., et al.: Current state of ontology matching. A survey of ontology and schema matching. Technical report 4, University of Turku, pp. 1–18 (2015)
23. Nayak, R., Tran, T.: A progressive clustering algorithm to group the XML data by structural and semantic similarity. Int. J. Pattern Recogn. Artif. Intell. **21**(04), 723–743 (2007)
24. Nayak, R., Xia, F.B.: Automatic integration of heterogenous XML-schemas. In: Proceedings of Information Integration and Web Based Appslications & Services, pp. 1–10 (2004)
25. Ngo, D., Bellahsene, Z.: Overview of YAM++(not) yet another matcher for ontology alignment task. Web Semant.: Sci. Serv. Agents WWW **41**, 30–49 (2016)
26. Stoilos, G., Stamou, G., Kollias, S.: A string metric for ontology alignment. In: Proceedings of International Conference on the SW, pp. 624–637 (2005)
27. Thang, H.Q., Nam, V.S.: Xml schema automatic matching solution. Comput. Electr. Autom. Control Inf. Eng. **4**(3), 456–462 (2010)
28. Thuy, P.T., Lee, Y.-K., Lee, S.: Semantic and structural similarities between XML schemas for integration of ubiquitous healthcare data. Pers. Ubiquitous Comput. **17**(7), 1331–1339 (2013)