Available online at http://scik.org
Engineering Mathematics Letters, 2 (2013), No. 1, 1-19
ISSN 2049-9337

COMPUTATION OF ONTOLOGY RESEMBLANCE COEFFICIENTS FOR IMPROVING SEMANTIC INTEROPERABILITY

LAMBRINI SEREMETI[1], IOANNIS KOUGIAS[2,*]

[1]Faculty of Science and Technology, Hellenic Open University, Patras, Greece

[2]Department of Telecommunication Systems and Networks, Technological Educational Institute of Messolonghi, Greece

Abstract: In open and dynamic environments, where various heterogeneous agents need to communicate, a shared ontology that explicitly and formally describes the whole domain of interest, or an alignment that provides semantically related entities among distinct ontologies, can be employed. The former case is infeasible, because a unique conceptual view of a domain is not widely accepted. Hence, the case, usually adopted, is the latter, where independently developed heterogeneous ontologies exist. A challenging issue is how an agent, charged with the task of carrying out the alignment, should select a suitable execution of matchers, to establish correspondences between ontology entities, in the fastest and most efficient way. A solution to this challenge is to use metrics for estimating the resemblance of a given pair of ontologies. To this end, we propose two metrics, as similarity coefficients, to estimate the lexical or structural resemblance of a given pair of ontologies.

Keywords: ontologies; heterogeneity; ontology alignment; similarity metrics; semantic interoperability.

2000 AMS Subject Classification: 68Q55; 68M14; 68T30

---

[*]Corresponding author
Received December 10, 2012

1. Introduction

The ability to communicate is one of the key capabilities of an agent (whether human, or machine) within a multi-agent environment. The entities in these smart environments can maintain their own semantic descriptions by using ontologies, which are formal knowledge representation models [6]. Unfortunately, the variety of ways that a domain can be conceptualized, results in the creation of heterogeneous ontologies with contradicting, or overlapping parts. The heterogeneity, at the ontology level, can mainly occur because of two reasons: (1) different ontologies could use different terminologies to describe the same domain of interest, and (2) even if two ontologies use the same names/labels for their entities, their corresponding structure can be different. Consequently, in order to achieve successful communication within such environments, where ontologies are used, it is necessary to bring them into a mutual agreement, that is, to align them, by establishing semantically related entities between the two ontologies [3]. Various methods and tools for assisting the process of ontology alignment have been developed. Based on our experience [9] with small and medium size ontologies which are characterized by limited hierarchy and not well-defined terminology, we have observed that the proposed algorithms (lexical-based, structure-based, constraint-based, instance-based) and strategies (execution of a single alignment algorithm, a parallel, or a sequential execution of ontology alignment algorithms) do not perform well in the case of limited hierarchy structure of the involved ontologies, or absence of constraints, or entity labels being tightly oriented towards the creator's view of the domain and not towards a general, common and agreed-upon vocabulary of the domain.

The usual practice in an alignment problem is to select a suitable alignment tool, import the ontologies in question and finally accept, or reject some of the suggested correspondences that the tool has produced. This is significantly far from getting quickly the best results, because it depends on the selection of the tool, its collection of matchers (alignment algorithms), their composition in execution (parallel, or sequential), and especially on the characteristics of the imported ontologies, that is, the kind of heterogeneity they introduce, etc. So, one of the challenging issues in ontology alignment in multi-agent environments, when good results in real-time are needed, is to estimate whether the heterogeneity of the ontologies to be aligned, is mainly lexical, or structural.

The solution we propose to this challenging issue is to assess the lexical and the structural similarity of the pair of ontologies to be aligned and depending on these measures, decide whether to apply a string-based, or a structure-based alignment algorithm.

The rest of this paper is organized as follows: Sections 2 and 3 briefly introduce related work in ontology alignment and various similarity metrics that are used by different matchers, in order to compare entities of a given pair of ontologies from different perspectives. Section 4 presents the proposed similarity measures as coefficients able to estimate two ontologies' resemblance and Section 5 concludes with future work.

2. Related Work on Ontology Alignment

Many researchers have investigated the problem of ontology alignment, mostly by proposing several ontology alignment tools and matchers (or matching algorithms) [4], [5], [7], which exploit various types of information in ontologies, that is, entity labels, taxonomy structures, constraints and entities' instances. These tools can be classified into two large categories: those that make use of a single matcher in order to calculate similarities between ontology entities and those which use a family of parallel or sequential matchers in composition. In the latter category, the similarity between two ontology entities is finally computed by a composite method, such as a weighted aggregation of the similarities obtained by each matcher separately.

A challenging issue while applying these methods consists of deciding whether a single matcher, or a combination of different matchers, performs better and in what cases, that is, for which kind of ontologies in question. Hence, given a specific pair of ontologies to be aligned, one should define a criterion to determine when a special matcher should be used. Based on this consideration, we propose the calculation, during a pre-alignment step, of two similarity coefficients, which estimate whether the resemblance of the ontologies in question is mainly lexical, or structural. Then, depending on their values, an agent who is charged with the task of the alignment process, can select the execution of suitable matchers, in order to establish correspondences between ontology entities, in a more effective and efficient way.

3. Related Work on Similarity Metrics

Considerable work has been made on metrics for measuring the degree of similarity between two entities of the ontologies to be aligned [2], [5], [8]. These metrics are functions that map a pair of entities of a given pair of ontologies to a value between 0 and 1, and they can be mainly classified into string-based and structure-based metrics. The purpose of these measures is to have a means to calculate lexical or structural similarity, respectively, between the entities of the given ontologies.

Our goal in presenting the new measures is to study the resemblance between ontologies in question, instead of studying the detailed relationship between entities of the given pair of ontologies, as do the metrics proposed in the literature. Although these metrics can provide good results regarding the similarity between entities, that is, at the entities' level, they are inappropriate at the ontology level. Ontologies used in multi-agent environments require processing in real-time, so the complexity of the classical metrics used at the ontology level, should be very low, leading to a fast estimation of ontologies' resemblance. As far as we know, such a kind of measures is used by the RiMOM ontology alignment multi-strategy [10], in order to enhance the alignment process. In comparison with RiMOM's metrics, our proposed measures appear to be more accurate, as we demonstrate later in section 4.

4. Similarity Coefficients

An agent charged with the task of the alignment process must be aware of the particularities of the source ontologies. To this end, we propose two similarity coefficients for ontology resemblance (structural, or lexical). These coefficients are used during a pre-alignment process, in order to select the suitable family of matchers, as well as the way of composing them. Their values fall into the range of the closed interval $[0,1]$. The first of the similarity coefficients examines the relative structure of the two ontologies, based on the comparison of the lengths of all paths leading from the root of each ontology to each of its leaves. The second one, after discovering concepts with identical labels in both ontologies, considers the relative proximity of these common concepts, inside each one of the ontologies to be aligned.

4.1 Definition of Similarity Coefficients

We define the Structural Similarity Coefficient, denoted by $\sigma(O_1, O_2)$, which is a similarity metrics at an ontology level (as opposed to an entity level), with values that range from 0 to 1. The Structural Similarity Coefficient describes the similarity between two ontologies globally (as opposed to local structural similarities between ontology entities), based on their structural resemblance. In order to compute it, one has to follow the constructive procedure described below:

4.1.1 **Definition of the Structural Similarity Coefficient**

Given two ontologies O_1 and O_2, calculate the vectors $\overline{l_1}$, $\overline{l_2}$ having as elements the lengths of all the paths from the root of each ontology, to all its leaves, i.e.,

$\overline{l_1} = [l_{11}, l_{12}, ..., l_{1i}, ...]$, with $l_{1i} =$ length of the path from the root of ontology O_1 to its i^{th} leaf, $i = 1, 2, ..., \#$ leaves of ontology O_1

$\overline{l_2} = [l_{21}, l_{22}, ..., l_{2j}, ...]$ with $l_{2j} =$ length of the path from root of ontology O_2 to its j^{th} leaf, $j = 1, 2, ..., \#$ leaves of ontology O_2

Let $L = \max\{|\overline{l_1}|, |\overline{l_2}|\}$, with $|\overline{l_i}|$ the dimension of vector $\overline{l_i}$, $i = 1, 2$. Create two new vectors $\overline{a}$, $\overline{t}$, by choosing between the vectors $\overline{l_i}$ $i = 1, 2$ the one that has the greatest dimension and by completing the other vector with leading zeros. Both vectors $\bar{a}$, $\bar{t}$, have dimension L.

If $|\bar{l}_i|>|\bar{l}_j|$, $i, j \in \{1,2\}$ and $i \neq j$, then $\bar{a}=\bar{l}_i$, $\bar{t}=\left[\bar{0}, \bar{l}_j\right]$, with the dimension of $\bar{0}$ being equal to $L-\min\{|\bar{l}_1|,|\bar{l}_2|\}$.

Now compute a square LxL matrix C, with elements $c_{ij}=|a_i-t_j|$, $i, j=1,2,...,L$. Then, create two new vectors $\bar{r}$ and $\bar{s}$, by appropriately reordering the vectors $\bar{a}$ and $\bar{t}$, as explained hereafter.

Let us consider two sets B and T with cardinalities equal to L and let β_i, τ_i, $i=1,2,...,L$, denote their respective elements. Consider the bipartite graph having as nodes the elements of the sets B and T and containing all possible edges between respective elements of the two sets. The edge linking β_i, to τ_j $i, j=1,2,...,L$, has a weight equal to $c_{ij}=|a_i-t_j|$. One can then always find a square matrix X with dimensions LxL having elements x_{ij}, $i, j=1,2,...,L$, such that the following relations hold:

1. $\forall i=1,2,...,L$, $\sum_{j=1}^{L} x_{ij}=1$

2. $\forall j=1,2,...,L$, $\sum_{i=1}^{L} x_{ij}=1$

3. $\forall i, j=1,2,...,L$, $x_{ij} \geq 0$

4. $\sum_{i=1}^{L}\sum_{j=1}^{L} c_{ij}x_{ij}$ is minimized

It can be proven that such elements x_{ij}, $i, j=1,2,...,L$, exist and take either the value 0, or the value 1. If $x_{ij}=1$, then the i^{th} element of the reordering $\bar{r}$ is $r_i=a_i$, while the j^{th} element of the reordering $\bar{s}$ is $s_j = t_j$. The structural similarity between the two ontologies is finally calculated as the cosine of the angle between the vectors $\bar{r}$ and $\bar{s}$:

$$\sigma(O_1, O_2) = \frac{\bar{r}.\bar{s}}{\|\bar{r}\|.\|\bar{s}\|} = \frac{\sum_{i=1}^{L} r_i s_i}{\sum_{i=1}^{L} r_i^2 \sum_{i=1}^{L} s_i^2}. \tag{1}$$

We define the Lexical Similarity Coefficient, at an ontology level, with values ranging from 0 to 1. In order to calculate the Lexical Similarity Coefficient, we consider two factors. The first factor is based on the number of concepts/classes having the same label in both ontologies (inter-ontology factor), while the second one takes into account the relative proximity that these common concepts have among them, inside each one of the ontologies (intra-ontology factor).

4.1.2 **Definition of the Lexical Similarity Coefficient**

Given two ontologies O_i and O_j, $i, j = 1, 2$, $i \neq j$, with a number of cc pairs of concepts with the same label, that is, $(\varepsilon_1^{O_i}, \varepsilon_1^{O_j})$, $(\varepsilon_2^{O_i}, \varepsilon_2^{O_j}), \ldots, (\varepsilon_k^{O_i}, \varepsilon_k^{O_j})$, $i, j = 1, 2$, $i \neq j$, $k = 1, 2, \ldots, cc$, respectively, the Lexical Similarity Coefficient is calculated as:

$$\lambda(O_i, O_j) = \frac{\sum_{k=1}^{cc} [1 - \frac{\left|\delta\left(\varepsilon_k^{O_i}\right) - \delta\left(\varepsilon_k^{O_j}\right)\right|}{\max(\delta\left(\varepsilon_k^{O_i}\right), \delta\left(\varepsilon_k^{O_j}\right))}]}{\max(\#conceptsofO_i, \#conceptsofO_j)}, \tag{2}$$

$i, j = 1, 2$, $i \neq j$, where the term $\delta\left(\varepsilon_k^{O_i}\right)$ ranks concept $\varepsilon_k^{O_i}$ of ontology O_i, by taking into account how far, in terms of number of edges, the remaining common concepts $\varepsilon_p^{O_i}, p \neq k$ are from concept $\varepsilon_k^{O_i}$ in ontology O_i and is given by

$$\begin{aligned} \delta\left(\varepsilon_k^{O_i}\right) &= \varphi_k^{O_i} + \frac{d\left(\varepsilon_k^{O_i}, n_1\right)}{cc-1}\left(1-\varphi_k^{O_i}\right)\rho + \frac{d\left(\varepsilon_k^{O_i}, n_2\right)}{cc-1}\left(1-\varphi_k^{O_i}\right)\rho\left(1-\rho\right) + \ldots \\ \ldots &+ \frac{d\left(\varepsilon_k^{O_i}, n_m\right)}{cc-1}\left(1-\varphi_k^{O_i}\right)\rho\left(1-\rho\right)^{m-1} + \frac{d\left(\varepsilon_k^{O_i}, \mathbf{O}\left(n_{m+1}\right)\right)}{cc-1}\left(1-\varphi_k^{O_i}\right)\left(1-\rho\right)^{m}, \end{aligned} \tag{3}$$

where

$$\varphi_k^{O_i} = 1 + (\alpha - 1)\,\mathrm{sgn}[(cc-1) - d\left(\varepsilon_k^{O_i}, n_1\right)]^{1}, \qquad (4)$$

with α, ($0 \prec \alpha \prec 1$) a constant added to the rank of common concept $\varepsilon_k^{O_i}$, due to its lexical similarity to concept $\varepsilon_k^{O_j}, i, j = 1, 2, i \neq j$ and where we define:

n_1 to be the 1-neighborhood of concept $\varepsilon_k^{O_i}$, containing all common concepts $\varepsilon_p^{O_i}, p \neq k$, that are within a distance of exactly one edge from $\varepsilon_k^{O_i}$ in O_i,

n_2 to be the 2-neighborhood of concept $\varepsilon_k^{O_i}$, containing all common concepts $\varepsilon_p^{O_i}, p \neq k$, that are within a distance of exactly two edges from $\varepsilon_k^{O_i}$ in O_i,

…

…

n_m to be the m-neighborhood of concept $\varepsilon_k^{O_i}$, containing all common concepts $\varepsilon_p^{O_i}, p \neq k$, that are within a distance of exactly m edges from $\varepsilon_k^{O_i}$ in O_i,

$\mathbf{O}\left(n_{m+1}\right)$ to be the remote-neighborhood of concept $\varepsilon_k^{O_i}$, containing all common concepts $\varepsilon_p^{O_i}, p \neq k$, that are within a distance of more than m edges from $\varepsilon_k^{O_i}$ in O_i.

Then, $d\left(\varepsilon_k^{O_i}, n_q\right), q = 1, 2, ..., m$, denotes the number of common concepts $\varepsilon_p^{O_i}, p \neq k$ that are within a distance of exactly q edges from $\varepsilon_k^{O_i}$ in O_i and $d\left(\varepsilon_k^{O_i}, \mathbf{O}(n_{m+1})\right)$ denotes the number of common concepts $\varepsilon_p^{O_i}, p \neq k$, within a distance of more than m edges from $\varepsilon_k^{O_i}$ in O_i.

---

[1]The signum function is defined as: $\mathrm{sgn}(x) = \begin{cases} -1 & if \quad x < 0 \\ 0 & if \quad x = 0 \\ 1 & if \quad x > 0 \end{cases}$

$\frac{1}{2} \prec \rho \prec 1$ is a forgetting factor, penalizing more severely the common concepts $\varepsilon_p^{O_i}, p \neq k$ that are more distant from $\varepsilon_k^{O_i}$ in O_i (in more distant neighborhoods).

4.2 Implementation of Similarity Coefficients

The idea behind the Structural Similarity Coefficient, is to compare the structure of the two ontologies, based on the minimization of the sum of the absolute values of the differences between the lengths of all the respective pairs of paths belonging to the two ontologies; these paths lead from the root of each ontology to each of its leaves.

In order to count the lengths of the paths, we can use a graph traversal algorithm like DFS (Depth First Search) together with a counter, initialized at zero, augmented by one each time an edge is found, decreased by one each time that backtracking is considered and memorized in a stack each time a leaf (no descendants) is encountered. DFS is effective enough, of complexity $\Theta(V^2)$ when a representation with adjacency matrices is used and $\Theta(V+E)$ when a representation with adjacency lists is used, where V is the number of the graph vertices and E is the number of the graph edges.

The vectors $\overline{l_1}$, $\overline{l_2}$ having as elements the lengths of all the paths of the ontologies thus obtained, may have different dimensions. That is why we add leading zeros to the vector with the lower dimension, in order to compensate this difference in dimensions (these zeros can be considered to correspond to the missing paths of one of the ontologies with respect to the other). The vectors $\overline{a}$ and $\overline{t}$ are thus obtained. We have now established a correspondence between the paths of one of the ontologies and the respective paths of the second one. In the aim to minimize the sum of the absolute values of the differences of the lengths of the corresponding pairs of paths, we need to reorder the vectors $\overline{a}$ and $\overline{t}$ into new vectors $\overline{r}$ and $\overline{s}$, respectively.

In order to achieve this, we reformulate the problem, as a linear assignment problem. We consider a bipartite graph with all possible nodes connecting the elements of two sets B and T of cardinality L, as seen in Figure 1. We consider that the edge linking β_i to τ_j, $i, j = 1, 2, ..., L$, has a weight equal to $c_{ij} = | a_i - t_j |$ (i.e. the absolute value of the difference of the lengths of the respective paths).

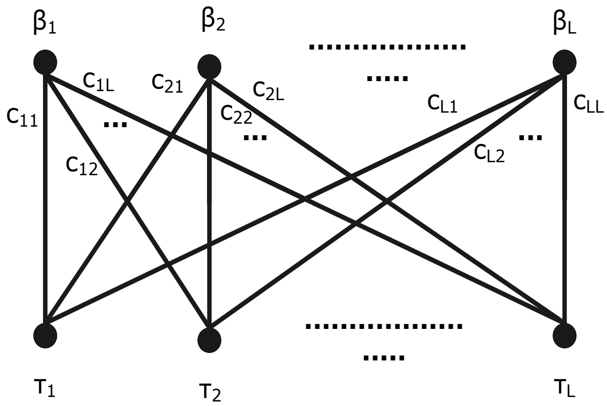

Figure 1. The bipartite graph between the elements of the sets B and T

The matrix C corresponds to a weight function $C : BxT \rightarrow R$. In order to maximize the resemblance between the structures of the two ontologies, we need to minimize the sum of the absolute values of the differences of lengths between respective paths, that is, referring to Figure 1, we need to find a bijection $f : B \rightarrow T$, such that the cost function $\sum_{i=1}^{L} c_{ij}$ is minimized, with $f(\beta_i) = \tau_j$ being the image of β_i under the bijection f. But, this is the formal definition of the linear assignment problem. The assignment problem is a special case of the transportation problem, which is a special case of the minimum cost flow problem, which in turn is a special case of the linear problem. It is thus possible to solve the minimization problem that we have, by using the simplex algorithm (very effective in practice, generally taking 2 to 3 times the number of equality constraints iterations at most and converging in expected polynomial time for certain distributions of random inputs), or more specialized algorithms, like the Bellman-Ford algorithm ($O(V^2 E)$), or the Hungarian algorithm ($O(V^2 \log(V) + VE)$). Hereafter, we re-express our minimization problem, as a standard linear problem. Find a matrix X with dimensions LxL, having elements x_{ij}, $i, j = 1, 2, ..., L$, that minimizes the objective function $\sum_{i=1}^{L}\sum_{j=1}^{L} c_{ij} x_{ij}$, subject to the following constraints:

1. $\forall i = 1, 2, ..., L$, $\sum_{j=1}^{L} x_{ij} = 1$, that is, each element of the set B is assigned to exactly one element of the set T

2. $\forall j = 1, 2, ..., L$, $\sum_{i=1}^{L} x_{ij} = 1$, that is, each element of the set T is assigned to exactly one element of the set B

Both the above mentioned constraints are due to the bijection f that we are searching.

3. $\forall i, j = 0, 1, ..., L,\ x_{ij} \geq 0$

The variables x_{ij}, $i, j = 1, 2, ..., L$ represent the assignment (or not) of β_i to τ_j, $i, j = 1, 2, ..., L$, taking the value 1 if the assignment is done and taking the value 0 otherwise. The vectors $\bar{r}$ and $\bar{s}$ are obtained by appropriately reordering the vectors $\bar{a}$ and $\bar{t}$ with the help of the matrix X, which is obtained as the solution of the simplex algorithm. The matrix X has only one non zero element in each of its rows and in each of its columns and this non zero element has a value of 1. If for some $x_{ij} = 1$, then the i^{th} element of the reordering $\bar{r}$ is $r_i = a_i$, while the j^{th} element of the reordering $\bar{s}$ is $s_j = t_j$.

Finally, the structural similarity between the two ontologies is calculated as the cosine of the angle between the vectors $\bar{r}$ and $\bar{s}$:

$$\sigma(O_1, O_2) = \frac{\bar{r}.\bar{s}}{\|\bar{r}\|.\|\bar{s}\|} = \frac{\sum_{i=1}^{L} r_i s_i}{\sum_{i=1}^{L} r_i^2 \sum_{i=1}^{L} s_i^2}. \quad (5)$$

As a more time efficient alternative, the reordered vectors $\bar{r}$ and $\bar{s}$ can be obtained by simply sorting the vectors $\bar{a}$ and $\bar{t}$ with a $V\log(V)$ algorithm like quicksort and then taking pairs of values which are at the same positions in the two sorted vectors.

The idea behind the Lexical Similarity Coefficient is to initially rank each common concept in both ontologies, based on the distance, in terms of the number of edges, between this common concept and all the remaining common concepts, in each ontology. Then, if a common concept is ranked equally in both ontologies, we assign the value 1 for this pair of common concepts in the calculation of the Lexical Similarity Coefficient, else, i.e., if a common concept is ranked differently in both ontologies, we substract from the value of 1, an amount which depends on the difference of rankings.

In order to compute the Lexical Similarity Coefficient, firstly, the concepts/classes of the two ontologies O_1 and O_2 are examined for the presence of same labels. After case normalization, diacritics suppression, blank normalization, link stripping, punctuation elimination applied to both ontologies, a total string is formed from the labels of all classes/concepts of ontology O_1. Then, each label of classes/concepts of ontology O_2 , is compared to this total string, by using a string matching algorithm, such as the Boyer-Moore algorithm ($\mathrm{O}(w)$, with w the length of the total string in O_1).

In this way, corresponding pairs of same labels $(\varepsilon_1^{O_1},\varepsilon_1^{O_2}),(\varepsilon_2^{O_1},\varepsilon_2^{O_2}),\ldots,(\varepsilon_k^{O_1},\varepsilon_k^{O_2}), k=1,2,...,cc$ are established and memorized, with $\varepsilon_k^{O_1}$ the label of a concept in O_1 and $\varepsilon_k^{O_2}$ the same label of the corresponding concept in O_2.

For each label $\varepsilon_k^{O_i}$, $i=1,2$, $k=1,2,...,cc$, we then count the number of labels $\varepsilon_p^{O_i}, p\neq k$ that are in the $1-,2-,...,m-$ neighborhoods of $\varepsilon_k^{O_i}$ in $O_i, i=1,2$, respectively. The remaining labels belong to the remote- neighborhoods $\mathbf{O}\left(n_{m+1}\right)$ in $O_i, i=1,2$. The quantities $d\left(\varepsilon_k^{O_i}, n_q\right), k=1,2,...,cc, i=1,2, q=1,2,...,m$, as well as $d\left(\varepsilon_k^{O_i}, \mathbf{O}(n_{m+1})\right), k=1,2,...,cc, i=1,2$, can thus be computed.

Practically, we search for the labels $\varepsilon_p^{O_i}, p \neq k$ in the 1-neighborhood (parent_of and children_of $\varepsilon_k^{O_i}$) and in the 2-neighborhood (parent_of (parent_of), children_of (parent_of) and children_of (children_of) $\varepsilon_k^{O_i}$). In this special case, the remaining labels belong to $\mathbf{O}(n_3)$.

$\frac{d(\varepsilon_k^{O_i}, n_q)}{cc-1}$ denotes the percentage of common labels $\varepsilon_p^{O_i}, p \neq k$ in a distance of exactly q edges from $\varepsilon_k^{O_i}$, $q=1,2,...,m$, i.e. in its q -neighborhood in $O_i, i=1,2$.

In the ideal case of an infinite number of $q-$neighborhoods, we would like to weight the percentage of common labels $\varepsilon_p^{O_i}, p \neq k$ in the n_q- neighborhood of $\varepsilon_k^{O_i}$, according to Table 1.

Table 1. Assignment of weights in the case of an infinite number of $q-$neighborhoods

n_1	n_2	n_3	$\cdots$	n_q	$\cdots$
ρ	$\rho(1-\rho)$	$\rho(1-\rho)^2$	$\cdots$	$\rho(1-\rho)^{q-1}$	$\cdots$

In practice, we restrain ourselves up to an m -neighborhood. In this case we have the following assignment of weights of Table 2.

Table 2. Assignment of weights in the case restrained to an m -neighborhood

n_1	n_2	n_3	$\cdots$	n_{m-1}	n_m	$O(n_{m+1})$
ρ	$\rho(1-\rho)$	$\rho(1-\rho)^2$	$\cdots$	$\rho(1-\rho)^{m-2}$	$\rho(1-\rho)^{m-1}$	ξ

We calculate ξ in such a way that the sum of weights equals 1:

$$\rho+\rho\left(1-\rho\right)+\rho\left(1-\rho\right)^2+...+\rho(1-\rho)^{m-2}+\rho(1-\rho)^{m-1}+\xi=1$$

from which, by using $1+\rho+\rho^2+...+\rho^{m-1}=\dfrac{1-\rho^m}{1-\rho}$, we deduct that $\xi=(1-\rho)^m$.

In order to have a decreasing series of weights, we impose $\xi\prec\rho(1-\rho)^{m-1}$, which leads to the choice $\rho\succ\dfrac{1}{2}$.

In the case where all $\varepsilon_p^{O_i}, p\neq k$ are in the 1-neighborhood of $\varepsilon_k^{O_i}$ in O_i, it is $\varphi_k^{O_i}=1$ and thus $\delta(\varepsilon_k^{O_i})=1$. In all other cases, it is $0<\varphi_k^{O_i}=\alpha<1$.

Concerning the complexity of the proposed Similarity Coefficients, the determining factor, in the case of the Structural Similarity Coefficient, is the complexity of the algorithm for resolving the assignment problem, while, in the case of the Lexical Similarity Coefficient, the determining factor is the string matching problem. Since the existing algorithms for solving these problems are efficient, exhibiting polynomial running time, they confer polynomial computational complexity to the herein proposed algorithms.

4.3 Examples of Similarity Coefficients

For the ontologies of Figure 2, we compute the Structural Similarity Coefficient as

$$\sigma(O_1,O_2)=\frac{2\cdot2+1\cdot1+1\cdot0}{\sqrt{2^2+1^2+1^2}\sqrt{2^2+1^2+0^2}}=\sqrt{\frac{5}{6}}=0.9129,$$

which means that they have very similar structure. The corresponding structure similarity factor used in [10], in order to measure the structural similarity between two ontologies, has a value of 0.5 in the case of our example. Our Structural Similarity

Coefficient depicts more accurately the similarity of structure between the two ontologies, which becomes apparent when flipping O_1 horizontally.

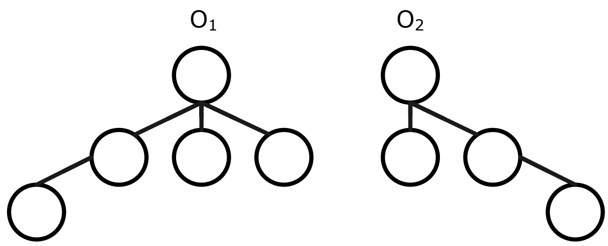

Figure 2. The ontologies of example 1

The Structural Similarity Coefficient for the ontologies of Figure 3 is calculated as

$\sigma(O_1, O_2) = \frac{n}{\sqrt{n}\sqrt{n^2}} = \frac{1}{\sqrt{n}}$, that is, $\sigma(O_1, O_2) \to 0$ as $n \to \infty$.

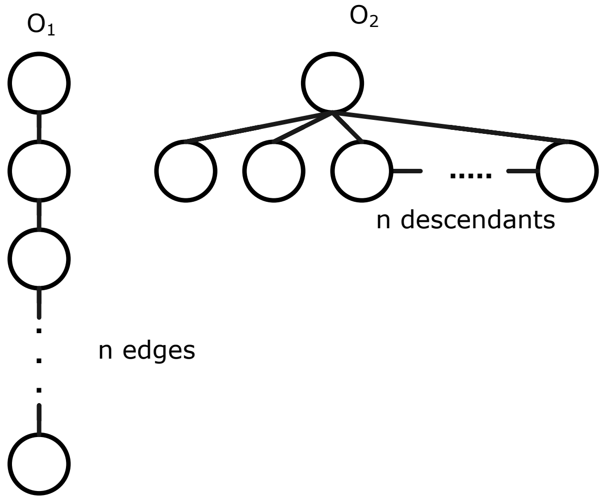

Figure 3. The ontologies of example 2

We consider now the ontologies of Figure 4 and compute the Lexical Similarity Coefficient of pairs O_1 and O_2 and O_1 and O_3, respectively, by choosing $a = 0.8$ and $\rho = 0.6$.

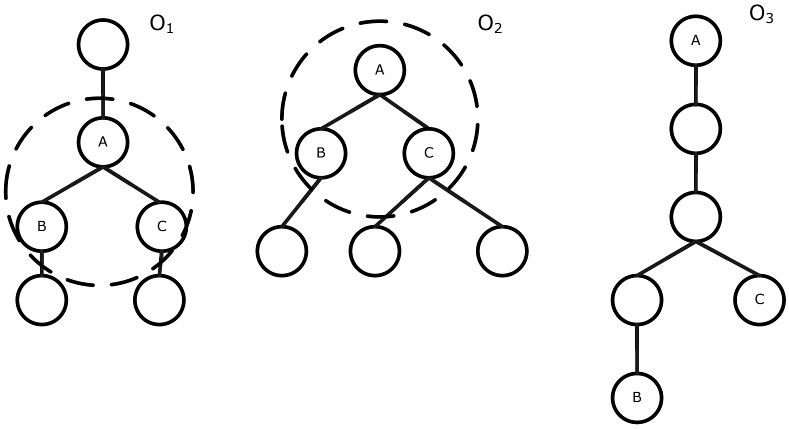

Figure 4. The ontologies of example 3

The ontologies O_1, O_2 and O_3 have common labels A, B and C. Thus, $cc = 3$ and we choose $m = 2$, limiting ourselves to $1-,2-$neighborhoods n_1, n_2 and $\mathbf{O}(n_3)$.

When computing the Lexical Similarity Coefficient between O_1 and O_2, since each common concept distributes in the same way the remaining common concepts in its neighborhoods, in both ontologies, it results that $\lambda(O_1, O_2) = \frac{1+1+1}{6} = 0.5$.

When comparing lexically O_1 to O_3, it is

$\delta\left(A^{O_1}\right)$=1,

$$\delta\left(B^{O_1}\right) = \delta\left(C^{O_1}\right) = 0.8 + \frac{1}{2}\cdot 0.2 \cdot 0.6 + \frac{1}{2}\cdot 0.2\cdot 0.6\cdot 0.4 = 0.884$$

$$\delta\left(A^{O_2}\right) = \delta\left(B^{O_2}\right) = \delta\left(C^{O_2}\right) = 0.8 + 1\cdot 0.2\cdot 0.4^2 = 0.832$$

resulting in

$$\lambda(O_1, O_3) = \frac{(1-0.168)+(1-0.0588)+(1-0.0588)}{6} = 0.4524.$$

The lexical similarity factor proposed in [10], is computed to be equal to 0.5, for both pairs of ontologies of the above presented example, taking into account that the three concepts A, B and C are common in all ontologies (in a total of 6 concepts in each ontology), but ignoring the fact that interrelations among them are not preserved the same in O_3. In comparison, our Lexical Similarity Coefficient is more accurate. This is justified by the results obtained, where we calculate the lexical similarity between O_1 and O_2 to be equal to 0.5 (the interrelations among the common concepts A, B and C are preserved in ontologies O_1 and O_2), while in the case of the lexical similarity between O_1 and O_3, our coefficient is calculated to be less than 0.5, depicting the differences in the interrelations among common concepts in these ontologies.

The detection of such differences in interrelations among common concepts is essential, since it restricts the problem of polysemy (words that have multiple senses), occurring when comparing ontology entities on the basis of their labels. Indeed, intuitively, groups of common labels in both ontologies, are more probably referring to the same concepts, while distant distinct common labels, may reflect homonyms and thus name different concepts.

Another example is depicted in Figure 5, where the ontologies O_1 and O_2 have four common concepts.

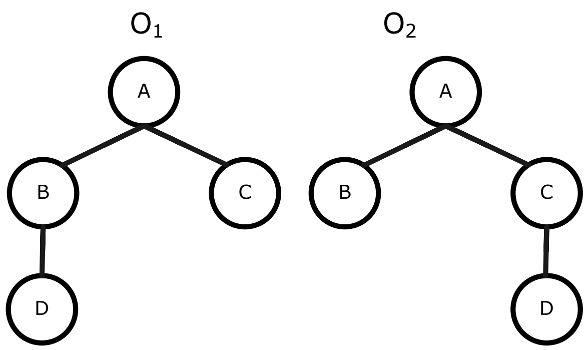

Figure 5. The ontologies of example 4

Here, the common concepts B and C distribute differently the remaining common concepts (A, C and D for B and A, B and D for C), while A and D distribute their respective remaining common concepts in the same way, in both ontologies. The result obtained is $\lambda(O_1, O_2) = 0.9832$. In opposition, the Lexical Similarity Factor proposed in [10] is calculated to have a value of 1 for this pair of ontologies, thus considering them as identical. The Lexical Similarity that we propose is still more accurate, having a value of less than 1, due to the differences in interrelations between the common concepts in the two ontologies. The exact amount of the difference obtained, can be adjusted by a proper choice for the values of the weighting coefficients a and ρ.

5. Conclusion and future work

Ontology alignment tools have benefited a lot from the use of lexical and structural similarity measures, in order to discover semantic correspondences between entities of different ontologies. Though powerful metrics exist in literature, they have been developed and purposed for a entities' level comparison, instead of the herein proposed metrics, which are suitable for a comparison at the ontology level. The ascertainment that short size ontologies, as well as the particularities of other ontologies influence adversely the performance of alignment tools that comprise a family of matchers and that use metrics which are suitable for large-scale ontologies, motivated the suggestion of two coefficients, which guide the selection of the right composition of available matchers, in order for the alignment to be correct and fast.

Future work includes the implementation of these coefficients by using the Alignment API 4.0 [1]. Then, experiments will be carried out with real-world ontologies and finally standard metrics, such as precision (the percentage of correctly discovered alignment in all discovered alignments) and recall (the percentage of correctly discovered alignments in all correct alignments) will be used, to evaluate the alignment results obtained.

REFERENCES

[1] J. David, J. Euzenat, F. Scharffe, C. T. dos Santos, The Alignment API 4.0, Semantic Web Journal 2 (2010) 3-10.

[2] D. Dhyani, M. W. Keong, S. Bhowmick, A survey of web metrics, aCM Computing Surveys 34 (2002) 469-503.

[3] M. Ehrig, Ontology Alignment: Bridging the semantic Gap, Springer, 2007.

[4] M. Ehrig, Y. Sure, FOAM – framework for ontology alignment and mapping; Results of the ontology alignment initiative, in: B. Ashpole, M. Ehrig, J. Euzenat, H. Stuckenschmidt (Eds.), CEUR Workshop on Integrating Ontologies Proceedings, 2005, pp. 72-76.

[5] J. Euzenat, P. Shvaiko, Ontology Matching, Springer-Verlag, Heidelberg (DE), 2007.

[6] T. Gruber, Towards Principles for the Design of Ontologies used for Knowledge Sharing, International Journal of Human-Computer Studies 43 (1995) 907-928.

[7] W. Hu, Y. Qu, G. Cheng, Matching large ontologies: A divide-and-conquer approach, Data & Knowledge Engineering 67 (2008) 140-160.

[8] R. Ichise, An analysis of multiple measures for ontology mapping problem, International Journal of Semantic Computing 4 (2010) 103-122.

[9] A. Kameas, L. Seremeti, Ontology-based knowledge management in NGAIEs, in: T. Heinroth, W. Minker (Eds.), Next Generation Intelligent Environments: Ambient Adaptive Systems, Springer, 2011, pp. 85-126.

[10] J. Li, J. Tang, Y. Li, Q. Luo, RiMOM: A dynamic multistrategy ontology alignment framework, IEEE Transaction on Knowledge Data Engineering 21 (2009) 1218-1232.