Evolution-based Refinement of Cross-language Ontology Alignments

Juliana Medeiros Destro, Julio Cesar dos Reis, Ricardo da S. Torres[1], Ivan Ricarte[2]

[1]Institute of Computing – University of Campinas (UNICAMP) – Brazil

[2]School of Technology – University of Campinas (UNICAMP) – Brazil

{juliana.destro,jreis,rtorres}@ic.unicamp.br, ricarte@unicamp.br

Abstract. *Ontology alignment plays a key role for information interconnectivity between computational systems relying on ontologies described in different natural languages. Existing approaches for ontology matching usually provide equivalent type of relation in the generated mappings. In this article, we propose a refinement technique to enable the update of the semantic type of the mapping such as "is-a", "part-of", etc. Our approach relies on information from the ontology evolution to apply refinement actions. We formalize the refinement actions and procedures, as well as apply the proposal in application scenarios.*

1. Introduction

Ontologies are usually created by different authors, using different vocabularies and, possibly, in different natural languages. The number of ontologies created in different languages grows as their use increases [Trojahn et al. 2014]. The process of creating correspondences, or mappings between concepts, is called ontology matching. Ontology mappings are crucial for enabling interconnectivity in multiple systems.

Ontology mapping refinement expands the semantic relations identified during the matching process. We differentiate *relation* from *relationship*, where the former represents a mapping, and the latter represents concept connections in an ontology. Refinement can modify or enrich semantic relations. For instance, during the refinement process, an equivalence ($\equiv$) relation (*i.e.*, a relation defining that two interrelated concepts are equivalent) can be modified to an *is-a* ($\sqsubseteq$) (*i.e.*, representing a relation in which one concept is a specialization of the other) [Arnold and Rahm 2014]. Mapping refinement poses a challenge due to the difficulties in establishing semantic relations between concepts, beyond the relatedness identified by the matching procedures.

Ontology mappings with enriched semantic correspondences might boost ontology merging [Raunich and Rahm 2011]. However, most of the matching approaches are approximative and infer mappings based on thresholds of relatedness between concepts. In this sense, finding only exact matches or at most *is-a* relations between pairs of concepts affects not only monolingual ontology matching, but also cross-lingual matching, when the source and target ontologies are labeled in different natural languages.

When a knowledge domain expands, ontologies representing the domain need to be updated to reflect domain changes. Consequently, ontologies are constantly evolving, by adding and removing concepts and relationships over time. These changes indicate how concepts and their relationships with each other evolved. In this investigation, we argue that ontology changes can be a valuable source of information to enhance the correspondences found between concepts beyond equivalences based on the type of semantic relation.

Our goal is to investigate the use of ontology evolution information, such as change operations (*e.g.*, changing the value of an attribute, and removal of a concept), to help in the process of mapping refinement. We believe that the use of this information might provide an understanding of how the concepts were updated over time to support the decision and application of actions required to modify the type of semantic relation in mappings.

Our conducted investigation assesses change operations and their correlation with semantic relations in mappings. In particular, we analyze how the evolution of concepts impacts on their relationships with neighbor concepts on the same ontology via illustrative cases. In addition, we analyze how that fact can be useful to enrich the semantics of correspondences. Overall, our study reveals a promising approach on the use of ontology evolution changes to enhance semantic relations in mappings.

The remainder of this article is organized as follows: Section 2 presents the foundations and related work. Afterwards, in Section 3 we present a set of formal definitions including the research problem. Section 4 presents our proposal for refinement of cross-lingual mappings. Section 5 introduces an application scenario to exemplify our proposal and discusses lessons learned. Finally, Section 6 wraps up the article and points out future research.

2. Related Work

The effects of ontology evolution in mappings have been already demonstrated in the literature [Dos Reis et al. 2014]. Ontology changes may impact established mappings and, for instance, cause modifications in semantic relations between interrelated concepts. Several examples of ontology mapping changes where presented by [Gross et al. 2012]. A detailed descriptive analysis of the impact of ontology changes on mappings were presented in the work [Dos Reis et al. 2014]. They showed the correlation between ontology changes and mapping evolution. The method proposed by [Dinh et al. 2014] aims at identifing the most relevant concept's attributes for supporting mapping adaptation when ontologies evolve, using differences identified among current and past versions of the ontologies.

Several approaches were developed to tackle the matching problem. For instance, [Trojahn et al. 2014] presented an extensive survey on matching systems and techniques for accomplishing multilingual and cross-lingual ontology matching. Ontology matching techniques have considered the use of similarity methods relying on background knowledge. Similarity measures aim to calculate the degree of relatedness between concepts exploiting different knowledge sources (*e.g.*, ontologies, thesauri, and domain corpora).

Align ontologies by doing matching with external knowledge sources was proposed in the work [Aleksovski et al. 2006]. They explored paths between the anchored matched concepts to find mapping between concepts. Differently, [Sabou et al. 2008] proposed to align ontology concepts by selecting the most appropriate ontology over multiple and heterogeneous knowledge sources. The *TaxoMap* approach has used the *WordNet* lexical database as background knowledge [Hamdi et al. 2010].

Domain-specific resources have only been superficially studied in the biomedical domain for the purpose of ontology alignment. For example, [Zhang and Bodenreider 2007] proposed exploring the *Unified Medical Language System* (UMLS)[1] structure to accomplish matching between anatomical ontologies. Their results indicated that domain knowledge is a key factor in the identification of additional mappings compared with the generic matching approach.

Ontology mapping refinement helps to expand the types of semantic relations identified during the matching process. Some techniques use external resources aiming to improve and increase the number and precision of established mappings. The work of [Arnold and Rahm 2014] defined a mapping refinement technique by using a set of equivalent mappings as input. They explored generic external resources and proposed a two-step enrichment technique to improve existing imprecise mappings. They used linguistic techniques and resources like *WordNet* to refine semantic relations between aligned concepts. Their work aimed to transform equivalence between concepts into an *is-a* or *part-of* relation, which may further reflect the real semantics of mapped concepts. The use of external resources influences the results and needs further research to determine their impact. The work of [Stoutenburg 2008] argues that the use of upper ontologies (an ontology which consists of very general terms that are common across all domains) and linguistic resources might enhance the alignment process.

The *TaxoMap* matching tool [Hamdi et al. 2010] explored pattern-based refinement techniques, applying manually created patterns to other mappings in the same domain. In contrast, [Spiliopoulos et al. 2008] presented the *Classification-Based Learning of Subsumption Relations* method for ontology alignment. This automated method relies on the exploration of patterns that describe the relation between concepts (*e.g.*, siblings at the same hierarchy level or attributes with same content). These patterns are identified by applying a classification task using machine learning methods.

Our proposed approach in this investigation is a novel technique for mapping refinement. We combine the information obtained from ontology change operations with semantic similarity measures based on defined refinement actions applicable to mappings.

3. Definitions and Problem Characterization

3.1. Definitions

We use the following notations and definitions throughout this paper.

Ontology. An ontology O specifies a conceptualization of a domain in terms of concepts, attributes and relationships [Gruber 1995]. Formally, an ontology $O = (C_O, R_O, A_O)$ consists of a set of concepts C_O or $Concepts_O$ interrelated by directed relationships R_O. For each concept $c_k \in C_O$, $L(c_k)$ defines the value of the preferred label for c_k expressing its name denoted by a natural language string. For example, *"cardio vascular diseases"* describes the label of a concept. The labels can be defined by properties in RDF schema like *rdfs:label*, and SKOS (*Simple Knowledge Organization System*) like *skos:prefLabel*. Concept $c_k \in C_0$ is associated with a set of attributes $A_O(c) = \{a_1, a_2, ..., a_p\}$. Each relationship $relation(c_1, c_2) \in R_O$ is typically a triple (c_1, c_2, r), where r is the relationship (*e.g.*, "is_a", "part_of", and "advised_by")

[1] www.nlm.nih.gov/research/umls (As of May 2018).

inter-relating c_1 and c_2. Neighborhood of a concept consist of the set of concepts with a relation to c_s, defined as $neighborhood(c_s) = c_s, sup(C_s), sub(C_s)$, where $sup(C_s)$ is the set of super concepts of c_s and $sub(C_s)$ is the set of sub concepts of c_s.

Similarity between concepts. Given two particular concepts c_1 and c_2, the similarity between them can be defined as the maximum similarity between each couple of attributes from c_1 and c_2. Formally:

$$sim(c_1, c_2) = \arg\max\ sim(a_{1x}, a_{2y}) \tag{1}$$

where $sim(a_{1x}, a_{2y})$ is the similarity between two attributes a_{1x} and a_{2y} denoting concepts c_1 and c_2 respectively. We can compute this similarity at different linguistic levels: *character*, *string*, and *semantic* level [Dinh et al. 2014].

Similarity measures. Similarity function used to calculate similarity between concepts. Formally:

$$f(c_1, c_2) = sim(c_1, c_2) \tag{2}$$

where $f(c_1, c_2)$ is the similarity function and $sim(c_1, c_2)$ denotes the calculated similarity between concepts c_1 and c_2. We use similarity measures at *character* and *semantic* linguistic levels.

Mapping. Given two concepts c_1 and c_2 from two different ontologies, a mapping m_{12} can be defined as:

$$m_{12} = (c_1, c_2, semType, conf) \tag{3}$$

where $semType$ is the semantic relation connecting c_1 and c_2.

The following types of semantic relation are considered: *unmappable* [$\perp$], *equivalent* [$\equiv$], *narrow-to-broad* [$\leq$], *broad-to-narrow* [$\geq$] and *overlapped* [$\approx$]. For example, concepts can be equivalent (*e.g.*, "cabeça"$\equiv$"head"), ("cabeça" in Portuguese language) one concept can be less or more general than the other (*e.g.*, "thumb"$\leq$"dedo")("dedo" in Portuguese language) or concepts can be somehow semantically related ($\approx$). The $conf$ is the similarity between c_1 and c_2 indicating the confidence of their relation [Euzenat and Shvaiko 2013]. $\mathcal{L}_{XY} = \{(m_{12})_k | k \in \mathbb{N}\}$ consists of the set of mappings between two ontologies O_X and O_Y as the result of an alignment process. Cross-lingual mapping is established between O_X and O_Y with concepts denoted by different natural languages, where $L(c_1)$ is expressed in language α, and $L(c_2)$ is expressed in language β such that $\alpha \neq \beta$. In a cross-lingual mapping, O_X and O_Y have concepts denoted by different natural language.

Ontology change operations. An ontology change operation (OCO) is defined to represent a change in an attribute, in a set of one or more concepts or in a relationship between concepts. OCO is classified into two main categories: *atomic* and *complex* changes (*cf.* Table 1). Each OCO in the atomic category cannot be split into smaller operations, whereas each one in the complex category is composed of more than one

Table 1. Ontology change operations (OCOs) [Hartung et al. 2013].

Change operation		Description
A	$addC(c)$	Addition of a new concept $c \in O_X^j$
t	$delC(c)$	Deletion of an existing concept $c \in O_X^{j-1}$
o	$addA(a,c)$	Addition of a new attribute a to a concept $c \in O_X^{j-1}$
m	$delA(a,c)$	Deletion of an attribute a from a concept $c \in O_X^{j-1}$
i	$addR(r,c_1,c_2)$	Addition of a new relationship r between two concepts c_1 and c_2 which belongs to O_X^{j-1}
c	$delR(r,c_1,c_2)$	Deletion of an existing relationship r between two concepts c_1 and c_2 which belongs to O_X^{j-1}
	$chgA(c,a,v)$	Change of attribute a in concept c with the new value v
	$moveC(c,p_1,p_2)$	Moving of concept c (and its subtree) from concept p_1 to concept p_2
C	$substitute(c_i,c_j)$	Replacement of concept $c_i \in O_X^{j-1}$ by concept $c_j \in O_X^j$
o	$merge(C_k,c_j)$	Fusion of a set of multiple concepts $C_k \subset O_X^{j-1}$ into concept $c_j \in O_X^j$
m	$split(c_i,C_r)$	Split of concept $c_i \in O_X^{j-1}$ into a set of resulting concepts $C_r \subset O_X^j$
p	$toObsolete(c)$	Sets status of concept c to *obsolete* (c is no longer available)
l	$delInnerC(c_i,p_j)$	Deletion of concept c_i where $p_j \in sup(c_i)$ and $sub(c_i) \neq \emptyset$ from ontology O_X^{j-1}
e	$delLeafC(c_i,p_j)$	Deletion of leaf concept c_i where $p_j \in sup(c_i)$ and $sub(c_i) = \emptyset$ from ontology O_X^{j-1}
x	$addInnerC(c_i,p_j)$	Addition of a sub concept c_i under the concept $p_j \in sup(c_i)$ to the ontology O_X^j
	$addLeafC(c_i,p_j)$	Addition of leaf concept c_i where $p_j \in sup(c_i)$ and $sub(c_i) = \emptyset$ to the ontology O_X^j
	$revokeObsolete(c)$	Revokes obsolete status of concept c (*i.e.*, c becomes active)

atomic operation. For instance, the operation $chgA(c,a,v)$ is composed of two atomic operations, $delA(a,c)$ and $addA(a,v)$. We denote successive ontology versions derived from evolution by O^{j-1} and O^j to identify ontologies created in time $j-1$ and j. Changes may occur from one version to another, and we consider existing tools to automatically detect change operations [Noy et al. 2002].

3.2. Problem Statement

Consider two versions of the same source ontology O_X^{j-1} at time $j-1$ and O_X^j at time j, a target ontology O_Y^j, and a set of mappings $\mathcal{L}_{XY}^j$ between O_X^j, and O_Y^j at time j. Suppose that the frequency of new releases of O_X and O_Y is different and at time j only O_X has evolved. We assume that the evolution is likely to provide useful information for mapping refinement of $\mathcal{L}_{XY}^j$, to enrich semantic relations and obtain the refined mappings $\mathcal{L'}_{XY}^j$. All mappings in $\mathcal{L}_{XY}^j$ have initially the type of semantic relation *equivalent* [$\equiv$] or *overlapped* [$\approx$] and we assume them as a mapping candidate set.

Given a mapping $m_{12} \in \mathcal{L}_{XY}^j$ associated with a concept c_1 affected by changes in the ontology, the challenging issue is to determine an exact and suited action of refinement to apply to m_{12}. To address this challenge, we define and formalize a set of *mapping refinement actions* (*cf.* Section 4.1).

The mapping refinement actions are part of refinement procedures, playing a key role to improve the quality of mappings. The objective is to enrich the mapping set by considering different semantic relations between concepts, for instance, equivalence relations can be refined to *is-a* or *part-of*.

In this investigation, we study how $\mathcal{L}_{XY}^j$ can be refined (*e.g.*, new mapping relations derived) based on ontology changes related to ontology evolution. The refined output consists of the $\mathcal{L'}_{XY}^j$. In particular, we address the following research questions:

- How to exploit ontology change operations for mapping refinement?
- Is it possible to reach mapping refinement without applying a new matching operation in the whole target ontology?

4. Refinement of Ontology Mappings

We propose and formalize a set of refinement actions aiming at refining mapping sets (Subsection 4.1) and how these actions are applicable in a refinement procedure (Subsection 4.2).

4.1. Refinement Actions

We present an approach to refine ontology mappings based on different types of ontology changes (Table 1). The proposal explores OCOs for refining mappings individually. For this purpose, we define actions as pre-defined behaviours of mapping refinement into algorithms designed to enrich ontology mappings according to ontology evolution (*cf.* Section 4.2).

The distinct actions representing different possibilities for refining mappings include: *mapping movement*, *mapping derivation*, *semantic relation modification* and *no action*. In the following, we formally describe each action. To this end, let $m_{12} \in \mathcal{L}^{j}_{XY}$ be the mapping between two particular concepts $c_1 \in O^j_X$ and $c_2 \in O^j_Y$.

Mapping derivation source. This is an action for which an existing mapping from $\mathcal{L}^{j}_{XY}$ derives a new mapping with the same target concept and different source concept. This action results in addition of a new mapping m_{k2} to $\mathcal{L}'^{j}_{XY}$.

$$\begin{array}{l} deriveS(m_{12}, c_k) \longrightarrow m_{12} \in \mathcal{L}^{j}_{XY} \wedge m_{k2} \notin \mathcal{L}^{j}_{XY} \wedge \\ (\exists c_k \in O^j_X, m_{k2} \in \mathcal{L}'^{j}_{XY} \wedge sim(c_1, c_k) \geq \sigma) \wedge \\ m_{12} \notin \mathcal{L}'^{j}_{XY} \end{array} \tag{4}$$

where $sim(c_1, c_k)$ denotes the similarity between c_1 and $c_k \in neighborhood(c_1)$, and σ denotes the threshold used to compare the derived mapping.

Mapping derivation target. This is an action for which an existing mapping m_{12} in $\mathcal{L}^{j}_{XY}$ derives a new mapping with the same source and a different target. This action results in addition of a new mapping m_{1v} to $\mathcal{L}'^{j}_{XY}$.

$$\begin{array}{l} deriveT(m_{12}, c_v) \longrightarrow m_{12} \in \mathcal{L}^{j}_{XY} \wedge m_{1v} \notin \mathcal{L}^{j}_{XY} \wedge \\ (\exists c_v \in O^j_Y, m_{1v} \in \mathcal{L}'^{j}_{XY} \wedge sim(c_1, c_v) \geq \sigma) \wedge \\ m_{12} \in \mathcal{L}'^{j}_{XY} \end{array} \tag{5}$$

Semantic relation modification. This is an action in which the type of the semantic relation of a given mapping is modified. This action is designed for supporting the refinement of mappings with different types of semantic relations rather than only considering the type of equivalence relation ($\equiv$).

$$\begin{array}{l} modSemType(m_{12}, new\_semType_{12}) \longrightarrow m_{12} \in \mathcal{L}'^{j}_{XY} \wedge \\ new\_semType_{12} \in \{\perp, \equiv, \leq, \geq, \approx \wedge semType_{12} \neq new\_semType_{12}\} \end{array} \tag{6}$$

The action for the modification of semantic relation can be applied in conjunction with the actions of move of mapping and derivation of mapping. That is when moving a mapping, it is also possible to modify the type of the semantic relation of such mapping. The same applies for derivation of mapping.

4.2. Refinement Procedure

The mapping refinement phase takes into account concepts from one version of the source ontology to another (O_X^{j-1} and O_X^j) to refine a candidate mapping set (suggests modifications on the mappings). The necessary instances of OCOs are identified from one ontology version at time $j-1$ to another at time j with a *diff* computation [Hartung et al. 2013]. It generates a *diff*, which is basically a set of changes identified between two versions of the same ontology. This article considers only the changes affecting O_X^j, *i.e.*, *diff*(O_X^{j-1}, O_X^j).

The candidate mapping set $\mathcal{L}_{XY}^j$ undergoes the mapping refinement procedure. We describe the procedure in two phases:

1. The output of executed ontology change detection tools is used to identify mappings with potential of refinement. The identification is based on the type of ontology evolution operations that affected the concepts. For instance, the addition of a concept to an ontology may indicate a specialization of another concept (*e.g.*, in O_X^j, the concept "*Eagle*" was added as child of the concept "*Bird*", being the former a specialization of the latter). Therefore, any candidate mapping involving the concepts "*Eagle*" and "*Bird*" are identified with possibility of refinement.
2. After the selection of mappings for refinement, for each selected mapping from $\mathcal{L}_{XY}^j$, an action is executed based on the type of ontology change. The action may include local rematch between concepts, a direct decision to perform modification in the semantic relation of the candidate mapping (*e.g.*, a $\equiv$ relationship may be replaced with a $\sqsubseteq$), or other appropriate action. The final output refers to the mapping set $\mathcal{L}_{XY}^{\prime j}$.

Algorithm 1 presents the main procedure to refine $\mathcal{L}_{XY}^j$. The input is the candidate mappings $\mathcal{L}_{XY}^j$ and the $diff_{(O_X^{j-1}, O_X^j)}$. For each mapping $m_{12} \in \mathcal{L}_{XY}^j$, the algorithm verifies if the concept $c_1 \in O_X^j$ was affected by change operations with the use of the $diff_{(O_X^{j-1}, O_X^j)}$. The algorithm then invokes the appropriate procedure for each case by considering addition change operations and revision change operations. If the concept was not affected by change operations from the $diff_{(O_X^{j-1}, O_X^j)}$, then no action is applied to (m_{12}). The output is the refined mapping $\mathcal{L}_{XY}^{\prime j}$.

We grouped the OCOs into two categories: (i) **AdditionOCO** adds concepts or information to concepts into the ontology. It consists of OCOs by including: $addC(c)$, $addInnerC(c_s, p_s)$, $addLeafC(c_s, p_s)$, $revokeObsolete(c)$, $addA(a, c_s)$ and $addR(r, c_{s1}, c_{s2})$; (ii) The **RevisionOCO** group of ontology changes revise existing concepts. It consists of OCOs such as: $merge(C_k, c_s)$ and $split(c_i, C_s)$. In the following, we explain the procedures involved by Algorithm 1.

AdditionProcedure. This procedure is invoked when c_1 was affected by some OCO in the *AdditionOCO* group. Algorithm 2 presents the proposed strategy for refining mappings associated to addition changes. For each mapping m_{12}, the neighborhood of the both c_1 and c_2 is retrieved to perform a local rematch. The *rematch* function receives a set of source concepts C_1 and a set of target concepts C_2 and returns a similarity matrix (*simMatrix*). The objective in applying a local rematch is to compare the similarities between the neighborhood of the source and target concepts. The similarities values found then drive modifications to the semantic relation established in m_{12}.For ex-

Algorithm 1 Mapping refinement procedure.

Require: $\mathcal{L}^{j}_{XY}; diff_{(O_X^{j-1}, O_X^{j})}$

for all $m_{12} \in \mathcal{L}^{j}_{XY}$ **do**
    for $c_1 \in m_{12}$ **do**
        if $AdditionOCO(c_1) \in diff_{(O_X^{j-1}, O_X^{j})}$ **then**
            $AdditionProcedure(m_{12})$
        else if $RevisionOCO(c_1) \in diff_{(O_X^{j-1}, O_X^{j})}$ **then**
            $RevisionProcedure(m_{12}; diff_{(O_X^{j-1}, O_X^{j})})$
        end if
    end for
end for
return $\mathcal{L}'^{j}_{XY}$

Algorithm 2 Mapping refinement for addition changes.

Require: m_{12}

for $c_1 \in m_{12}$ **do**
    $neighC_1 \leftarrow neighborhood(c_1);$
    $neighC_2 \leftarrow neighborhood(c_2);$
    $simMatrix(C_1, C_2) \leftarrow rematch(neighC_1, neighC_2);$
    for all $(c_{1i}, c_{2i}) \in simMatrix(C_1, C_2)$ **do**
        if $c_{1i} = sup(c_1)$ **and** $(sim(c_{1i}, c_2) > sim(c_1, c_2))$ **then**
            $semType \leftarrow relation(c_{1i}, c_1);$
            $modSemTypeM(m_{12}, semType);$
            $deriveS(m_{12}, c_{1i});$
        end if
        if $(c_{2i} = sup(c_2)$ **or** $c_{2i} = sub(c_2))$ **and**
        $sim(c_1, c_{2i}) => sim(c_1, c_2)$ **then**
            $deriveT(m_{12}, c_{2i});$
        end if
    end for
end for

ample, if $sim(sup(c_1), c_2) > sim(c_1, c_2)$, the algorithm modifies the semantic relation in m_{12} to the same semantic relation of $sup(c_1)$ and c_1 and add a new mapping between $sup(c_1)$ and c_2. The local rematch also helps establishing a derivation of mapping when the $sim(c_1, sub(c_2)) \geq sim(c_1, c_2)$ or $sim(c_1, sup(c_2)) \geq sim(c_1, c_2)$.

RevisionProcedure. This procedure is used to refine mappings when c_1 was affected by some OCO in the *RevisionOCO* group. Algorithm 3 describes the proposed strategy for the refinement. For each input mapping m_{12}, the algorithm retrieves the concepts from O_X^{j-1} involved in merge or split ontology change operations. In the *merge* operation, an initial set of concepts $C_k \subset O_X^{j-1}$ gives place to a concept $c_1 \in O_X^{j}$. On the other hand, in a *split* operation, an initial concept $c_1 \in O_X^{j-1}$ is split in a set of concepts $C_s \subset O_X^{j}$.

The algorithm extracts the before evolution concepts c_1 (in the split) and the set of

Algorithm 3 Mapping refinement for revision changes.

Require: $m_{12}; diff_{(O_X^{j-1}, O_X^j)}$

for $c_1 \in m_{12}$ **do**
  if $split(c_i, C_s) \in diff_{(O_X^{j-1}, O_X^j)}$ **and** $(c_1 \in C_s)$ **then**
    if $sim(c_i, c_2) > sim(c_1, c_2)$ **and** $semType(m_{12}) = \equiv$ **then**
      $modSemTypeM(m_{12}, \leq);$
    end if
  end if
  if $merge(C_k, c_1) \in diff_{(O_X^{j-1}, O_X^j)}$ **then**
    $neighC_2 \leftarrow neighborhood(c_2);$
    $simMatrix(C_k, C_2) \leftarrow \ rematch(C_k, neighC_2);$
    for all $(c_{ki}, c_{2i}) \in simMatrix(C_k, C_2)$ **do**
      if $(c_{2i} = sup(c_2)$ **or** $c_{2i} = sub(c_2))$ **and**
      $sim(c_{ki}, c_{2i}) \geq sim(c_1, c_2)$ **then**
        $deriveT(m_{12}, c_{2i});$
      end if
    end for
  end if
end for

concepts C_k (in the merge) and compares them with $c_2 \in m_{12}$. Our aim is to explore the similarities between c_2 and these concepts and set of concepts before the revision to O_X^j, to extract information and refine m_{12}.

For example, useful information for refinement would be the similarity of the concept $c_i \in O_X^{j-1}$ involved in the split of $c_1 \in m_{12} \wedge c_1 \in C_s$, and c_2. If $sim(c_i, c_2) > sim(c_1, c_2)$, we can infer that c_1 and c_2 do not hold an $\equiv$ relation and, thus, refine the semantic relation of m_{12}.

5. Application and Discussion

We illustrate our approach via two scenarios where the refinement procedures are applied to a pair of ontologies in the biomedical domain, O_X^j and O_Y^j, at time j. We explore concepts described in different natural languages (English and Portuguese, respectively).

5.1. Scenario 1

In this first scenario, we explore the refinement procedure applied to addition change OCO. Ontology O_X evolved over time, generating two different versions from $j-1$ to j. Concept c_1 "*Angina*" in ontology O_X^j is added as a sub concept of "*Heart*".

A set of mappings $\mathcal{L}_{XY}^j$ between O_X^j and O_Y^j, on time j, is given as input for the refinement procedure. Figure 1 illustrates the mapping $m_12 \in \mathcal{L}_{XY}^j$ between concepts c_1 "*Angina*" and c_2 "*Cardiopatia*". The refinement procedure requires as input the list of change operations (OCOs) detected from one version of the ontology to another.

Our proposal leverages the evolution information to refine the proposed mapping set by computing the similarity values between concept c_1 "*Angina*" and the concepts

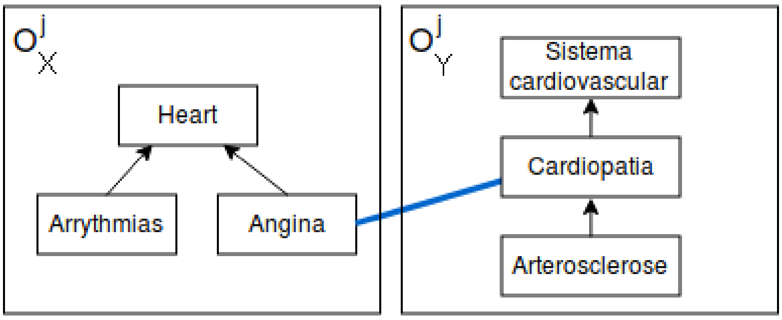

Figure 1. Illustration of the mapping $m_st \in \mathcal{L}^j_{XY}$ candidate for refinement.

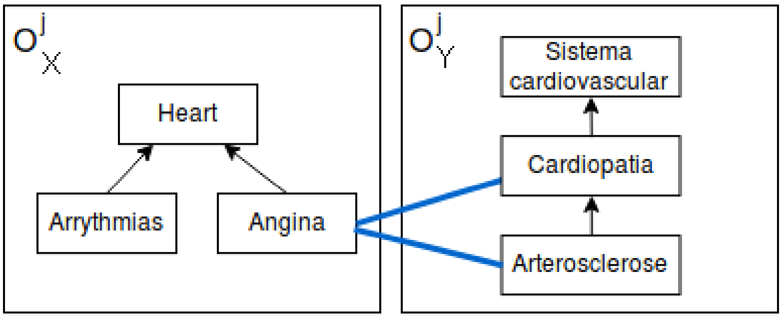

Figure 2. Resulting $\mathcal{L}'^j_{XY}$ after our refinement procedure (application of the derivation action).

of the neighborhood of target concept in j "*Cardiopatia*". To this end, we perform a cross-lingual local rematch. This local rematch is represented in step 2 of Algorithm 2.

If the similarity value between the concepts c_1 "*Angina*" and some neighbor c_{2i} of c_2 is higher than the original similarity value given by $sim(c_1, c_2)$, *i.e.* $sim(c_1, c_{2i}) \geq sim(c_1, c_2)$, the algorithm derives a mapping between c_1 and c_{2i} to reflect this finding, as illustrated in Figure 2. The input for the refinement procedure is the mapping sets and the list of change operations (OCO) affecting the source ontology.

5.2. Scenario 2

The second scenario, concept "*Cardiopathy*" in ontology O_X is split into a set of concepts $C_s \subset O^j_X$: "*Arrhythmias*" and "*Angina*".

Split operations $split(c_i, C_s)$ are used in ontology engineering to represent specializations of a given concept [Noy et al. 2002], creating a [$\leq$] relation between the original concept c_i and the concepts in the $C_s \subset O^j_X$. In order to leverage this information to modify the semantic relation of the candidate mapping between concepts "*Angina*" in O^j_X and "*Cardiopatia*" in O^j_Y, our proposed algorithm calculates similarity values between the original concept in concept from the ontology at time $j-1$ ("*Cardiopathy*") and the target concept in j "*Cardiopatia*". To this end, we perform a cross-lingual local rematch, represented in step 9 of Algorithm 3.

If the similarity $sim(c_i, c_2)$ between concepts c_i "*Cardiopathy*" and c_2 "*Cardiopatia*" is higher than the the original similarity value given by $sim(c_1, c_2)$ involving the concepts c_1 "*Angina*" and the concept c_2 , *i.e.* $sim(c_i, c_2) \geq sim(c_1, c_2)$, the algorithm modifies the semantic relation between c_1 and c_2 to reflect this finding (*cf.* Figure 3).

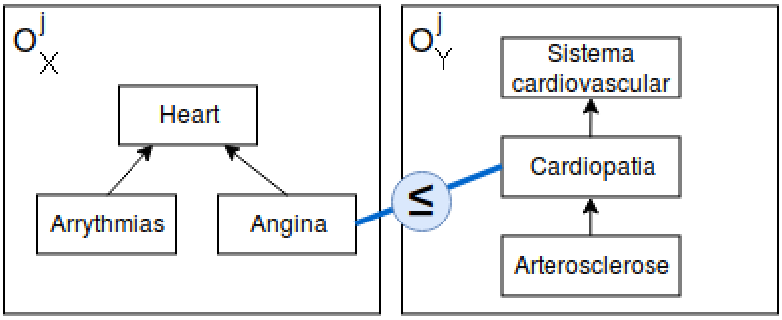

Figure 3. Cross-lingual mapping $\mathcal{L}'^{j}_{XY}$ after refinement procedure.

6. Discussion and Conclusion

Ontology mapping refinement remains an open research problem. The result of mapping refinement increases the usefulness of mapping sets, benefiting the semantic data integration of systems. This article proposed an original approach with the use of ontology change operations detected during ontology evolution to leverage mapping refinement.

We assumed that ontology evolution information is useful to decide the more adequate approach for the refinement and improve the mapping quality outcome. To the best of our knowledge, the use of ontology change operations for mapping refinement has never been proposed in literature. This aspects refers to the key originality of this paper.

Answering our research questions (i) *how to exploit ontology change operations for mapping refinement* and (ii) *if it is possible to reach mapping refinement without applying a new matching operation in the whole target ontology*: the actions defined based on the change operations and performed during the refinement procedure enrich the proposed mapping set with semantic context, which is beneficial for ontology merging and system integration. Our proposal defined algorithms that reach mapping refinement without applying a new matching operation with the whole target ontology.

Future work involves the investigation and systematic experimentation of our approach using real-world ontologies.

Acknowledgments

We thank CAPES (grant #88881.145912/2017-01), CNPq (grant #307560/2016-3), the São Paulo Research Foundation (FAPESP) (Grants #2017/02325-5 and #2013/08293-7)[2].

References

Aleksovski, Z., Klein, M., ten Kate, W., and van Harmelen, F. (2006). Matching unstructured vocabularies using a background ontology. In *Proceedings of the 15th International Conference on Managing Knowledge in a World of Networks*, pages 182–197.

Arnold, P. and Rahm, E. (2014). Enriching ontology mappings with semantic relations. *Data & Knowledge Engineering*, 93:1–18.

Dinh, D., Dos Reis, J. C., Pruski, C., Da Silveira, M., and Reynaud-Delaître, C. (2014). Identifying relevant concept attributes to support mapping maintenance under ontology evolution. *Journal of Web semantics*, 29:53–66.

[2]The opinions expressed in this work do not necessarily reflect those of the funding agencies.

Dos Reis, J. C., Pruski, C., Da Silveira, M., and Reynaud-Delaître, C. (2014). Understanding semantic mapping evolution by observing changes in biomedical ontologies. *Journal of biomedical informatics*, 47:71–82.

Euzenat, J. and Shvaiko, P. (2013). *Ontology matching*, pages 42,71–84,306. Springer, Heidelberg, 2nd edition.

Gross, A., Hartung, M., Thor, A., and Rahm, E. (2012). How do computed ontology mappings evolve?-a case study for life science ontologies. In *Proceedings of the 2nd Joint Workshop on Knowledge Evolution and Ontology Dynamics*, volume 890.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43:907 – 928.

Hamdi, F., Safar, B., Niraula, N. B., and Reynaud, C. (2010). Taxomap alignment and refinement modules: Results for oaei 2010. In *Proceedings of the 5th International Workshop on Ontology Matching (OM-2010)*, volume 689, pages 212–219.

Hartung, M., Gross, A., and Rahm, E. (2013). COnto-Diff: Generation of Complex Evolution Mappings for Life Science Ontologies. *Journal of Biomedical Informatics*, 46:15–32.

Noy, N. F., Musen, M. A., et al. (2002). Promptdiff: A fixed-point algorithm for comparing ontology versions. *AAAI/IAAI*, 2002:744–750.

Raunich, S. and Rahm, E. (2011). Atom: Automatic target-driven ontology merging. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 1276–1279. IEEE.

Sabou, M., d'Aquin, M., and Motta, E. (2008). Exploring the semantic web as background knowledge for ontology matching. *Journal on data semantics XI*, pages 156–190.

Spiliopoulos, V., Valarakos, A., and Vouros, G. (2008). Csr: discovering subsumption relations for the alignment of ontologies. *The Semantic Web: Research and Applications*, pages 418–431.

Stoutenburg, S. K. (2008). Acquiring advanced properties in ontology mapping. In *Proceedings of the 2nd PhD Workshop on Information and Knowledge Management (PIKM 2008)*, pages 9–16. ACM.

Trojahn, C., Fu, B., Zamazal, O., and Ritze, D. (2014). State-of-the-art in multilingual and cross-lingual ontology matching. In *Towards the Multilingual Semantic Web*, pages 119–135. Springer.

Zhang, S. and Bodenreider, O. (2007). Experience in aligning anatomical ontologies. *International journal on Semantic Web and information systems*, 3(2):1.