

# Identifying change patterns of concept attributes in ontology evolution

Duy Dinh<sup>1</sup>, Julio Cesar Dos Reis<sup>1,2</sup>, Cédric Pruski<sup>1</sup>, Marcos Da Silveira<sup>1</sup>,  
Chantal Reynaud-Delaître<sup>2</sup>

<sup>1</sup> CR SANTEC, Public Research Centre Henri Tudor, Luxembourg

<sup>2</sup> LRI, University of Paris-Sud XI, France

{duy.dinh,julio.dosreis,cedric.pruski,marcos.dasilveira}@tudor.lu,  
chantal.reynaud@lri.fr

**Abstract.** Ontology versions are periodically released to ensure their usefulness and reliability over time. This potentially impacts dependent artefacts such as mappings and annotations. To deal with this issue requires to finely characterize ontology entities' changes between ontology versions. This article proposes to identify change patterns at attribute values when an ontology evolves, to track textual statements describing concepts. We empirically evaluate our approach by using biomedical ontologies, for which new ontology versions are frequently released. Our achieved results suggest the feasibility of the proposed techniques.

**Keywords:** #eswc2014Dinh, ontology evolution, ontology changes, change patterns, ontology versioning, biomedical ontologies

## 1 Introduction

The dynamic aspect of knowledge in various domains requires that knowledge engineers apply changes to different ontology entities by adding, removing and revising them. This periodically leads to new ontology versions, which ensures that software applications use the most up-to-date representation of the domain knowledge. Ontology changes potentially impact mappings, annotations and queries which rely on these ontologies [1,2].

Changes applied to generate new ontology versions are not always fully documented, which impedes the minimization and handling of their impact. To this end, we need methods to automatically identify ontology change operations (OCOs) in an explicit way, given two versions of the same ontology [3]. Our previous studies have underlined the need of precisely characterizing the evolution of attributes describing concepts for maintaining mappings valid over time [4,5].

When analyzing two consecutive versions of the same ontology, for instance, we found cases where textual statements which are values of attributes describing concepts are completely transferred from one concept to its siblings. This had affected the associated mappings since their definition relies on such textual information. For example, we observed this case with the concept “560.39” of the

ICD-9-CM<sup>3</sup> (ICD) biomedical ontology. Such concept contains three attributes and one of them has as value “Fecal impaction” (release 2009). Five mappings are defined with this concept as domain, and one of these mappings has a range called “Fecal impaction (disorder)”, from SNOMED CT<sup>4</sup> (SCT). After evolution (*i.e.*, ICD release 2010), the attribute value “Fecal impaction” is no longer associated with the ICD concept and the previously mentioned mapping has been removed. Moreover, the concept “Fecal impaction” has been newly created in ICD (release 2010) and is reconnected to “Fecal impaction (disorder)” of SCT.

Literature has highlighted challenges related to ontology changes’ management and has proposed change patterns to improve the ontology evolution process [6,7]. Although useful tools exist to identify the most traditional and frequent OCOs between two ontology versions [3,8,9], taking into account the nature of changes (*e.g.*, atomic or complex) and the type of changes (*e.g.*, addition, removal, splitting, merging of entities), these tools fail to automatically identify ontology modifications at a finer level of detail, required for supporting tasks dependent on ontology changes (*e.g.*, mapping adaptation). This remains an open issue that requires further research.

To cope with this issue, our proposal underscores a nontrivial solution to recognize the diffusion of attribute values between concepts from one version of the ontology to another. We inquire whether techniques based on linguistic characteristics of textual values, combined with similarity measure, play a role in supporting automatic change patterns identification at the level of concept attributes. In summary, we make the following contributions:

- We formally define a set of *ontology change patterns* to express different behaviours of the evolution of attributes.
- We introduce a novel linguistic-based approach implementing methods to automatically identify instances of the proposed change patterns by comparing successive ontology versions. We investigate different techniques to rank candidate attributes in the identification method. Our systematic study provides useful tools to precisely characterize ontology evolution.
- We experimentally assess our approach by using real-world biomedical ontologies as a case study. We investigate the influence of different aspects in the performance of the proposed methods and our obtained results show innovative findings.

We structure the remainder of this article as follows: Section 2 discusses related work; Section 3 reports on our approach to change patterns; Section 4 describes the techniques for identifying change patterns; Section 5 presents the evaluation while Section 6 draws conclusions and future work.

## 2 Ontology change patterns

Noy & Klein [6] have originally evoked the notion of *change patterns* (CPs) through a first simple classification of changes that may affect entities of on-

<sup>3</sup> [www.cdc.gov/nchs/icd/icd9cm.htm](http://www.cdc.gov/nchs/icd/icd9cm.htm)

<sup>4</sup> [www.ihtsdo.org/snomed-ct](http://www.ihtsdo.org/snomed-ct)

tologies at evolution time. This classification, under basic and complex changes, paved the way for new approaches addressing ontology evolution. These approaches explore CPs to characterize complex changes and evolution scenarios, simplifying the management of ontologies to control the impact of the evolution and to ensure consistency in ontology [10].

Change patterns may allow to identify complex changes between versions of the same ontology. Groner *et al.* [11] addressed the problem of refactoring recognition using reasoning to semantically compare different versions of an OWL DL ontology. They proposed a high-level categorization of ontology changes like the refactoring patterns in software engineering, and applied it to OWL ontology.

Some approaches define CPs at the level of RDF data model. Auer & Herre [12] proposed to support ontology evolution by using basic changes and aggregate them into more complex changes in RDF. Their approach consists in annotating the derived compound changes with meta-information and classifying them as ontology evolution patterns. Differently, Rieß *et al.* [13] proposed a pattern-based approach to evolving data and refactoring RDF knowledge bases. They defined basic evolution patterns that can be combined into compound ones. Their work formally specifies modular evolution patterns in a declarative manner, capturing simple evolution and refactoring operations on both data and schema levels.

Djedidi & Aaufaure defined an ontology evolution methodology driven by a pattern-oriented modelling. They proposed the *Change Management Patterns* to guide the ontology evolution process by driving and controlling change application while maintaining consistency of the evolving ontology [14]. They considered four kinds of consistency concerning the OWL DL language: structural, logical, conceptual and domain modeling consistency [14]. The solution looks for invariances in change management that repeatedly appear when ontologies evolve. They proposed three types of patterns: change patterns classifying types of changes, inconsistency patterns classifying types of logical inconsistencies, and alternative patterns classifying types of inconsistency resolution alternatives.

Javed *et al.* suggested an approach to dealing with ontology evolution through a framework of compositional operators where they represent domain changes as CPs [7]. They composed this framework with different levels of change operators, and empirically studied ontology evolution to investigate the relationships between generic and domain-specific changes to determine common CPs.

This literature review clearly highlights that existing approaches exploit CPs to deal with ontology evolution, and frequently their definition relies on ontology meta-models and languages (*e.g.*, OWL or RDF). While existing change patterns seem sufficient to identify a set of inconsistencies, they remain inefficient for dealing with the impact of ontology evolution on dependent artifacts because their design fails to consider requirements for adapting mappings. We address CPs at the level of attribute values using linguistic-based features for identifying the diffusion of textual values between concepts over time. Complementary to other approaches, we refine meta-model patterns on model level to further support the ontology evolution impact, which influences the way we design the required CPs and the recognition methods.

### 3 Change patterns in attribute values

#### 3.1 Preliminaries and problem definition

We adopt a traditional definition of ontology [15]. We define a set of concepts of an ontology  $O_x$  at time  $j$  as  $C(O_x^j) = \{c_1^j, c_2^j, \dots, c_n^j\}$ . Each concept  $c_i^j \in C(O_x^j)$ , described by a set of attributes, has a unique identifier. We consider the set of attributes characterizing a concept  $c$  as  $A(c) = \{a_1, a_2, \dots, a_n\}$  (e.g., name, definition, synonym, etc.). For instance, an attribute  $a_i$ , of type *name*, contains the *value* “*cardio\_vascular\_diseases*”. We use  $a_i.value$  to denote the value of an attribute  $a_i$ . A relationship  $r \in R$  interconnects two concepts and has a specific type, e.g., “*subsumption*”, “*part-of*”, etc.

The *context* of a concept  $c_i$  in the ontology stands for a set of *super concepts* ( $sup(c_i)$ ), *sub concepts* ( $sub(c_i)$ ) and *sibling concepts* of  $c_i$  ( $sib(c_i)$ ), as following:

$$CT(c_i) = sup(c_i) \cup sub(c_i) \cup sib(c_i) \quad (1)$$

where

$$\begin{aligned} sup(c_i) &= \{c_k | c_k \in C(O_x), c_i \sqsubset c_k \wedge c_i \neq c_k\} \\ sub(c_i) &= \{c_k | c_k \in C(O_x), c_k \sqsubset c_i \wedge c_i \neq c_k\} \\ sib(c_i) &= \{c_k | c_k \in C(O_x), sup(c_k) = sup(c_i) \wedge c_i \notin sup(c_k)\} \end{aligned} \quad (2)$$

where  $c_i \sqsubset c_k$  means that  $c_i$  is related to  $c_k$  through a subsumption relationship.

Figure 1 depicts the investigated scenario. Given an attribute  $a_i$  from a concept  $c$  at time  $t_0$ , we investigate a way to characterize how such attribute evolves by considering the context of the concept  $c^1$  at time  $t_1$  (i.e., in the new version of ontology  $O_x$ ). Evolution of ontology entities usually remains restricted in an ontology space like the context [4]. We focus on  $a_i.value$  to identify useful behaviours of evolution concerning the attributes and search for describing these behaviours as well-delineated change patterns. We face issues to determine which attribute at time  $t_1$  represents the most adequate candidate in the recognition process to identify CPs occurrences. We apply syntactic analysis techniques to recognize textual values of attributes in different versions of the same ontology.

#### 3.2 Proposed change patterns

Considering **change patterns** (CPs) as means to deal with ontology entity changes, we focus on changes related to concept’s attribute values. Therefore, our defined change patterns relate to the linguistic characteristics of the attributes’ value before and after their evolution. We denote  $O_x^0$  an ontology  $O_x$  at time  $t_0$  and  $c_k^1 \in C(O_x^1)$  a concept belonging to this ontology at time  $t_1$ . A change pattern between an attribute  $a_p^0$  of concept  $c_k^0 \in C(O_x^0)$  and another attribute  $a_q^1$  of concept  $c_{cand}^1 \in CT(c_k^1)$  occurs when changes in the value of the attribute  $a_p^0$  which shares some similarity with the attribute  $a_q^1$  are observed. In addition, we suppose that the attribute  $a_q^1$  is new or its value differs at time  $t_1$  from the one at time  $t_0$ . Therefore, any change pattern must satisfy the following constraint :

$$a_q^0 \notin A(c_{cand}^0) \vee a_q^0 \neq a_q^1 \quad (3)$$

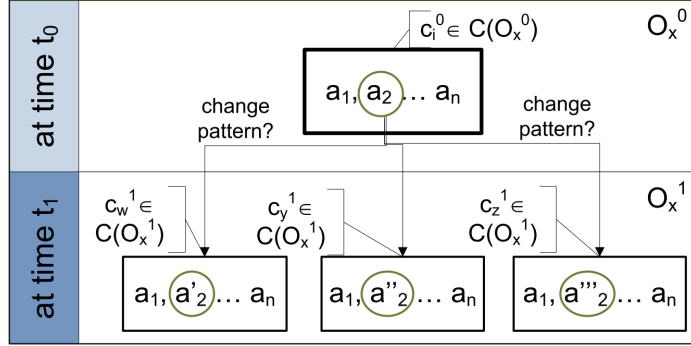


Fig. 1. Problem definition

We define the CP classes as “total copy” (TC), “total transfer” (TT), “partial copy” (PC), and “partial transfer” (PT). Table 1 illustrates the proposed change patterns and presents examples borrowed from the biomedical domain. We justify our definition of CPs through the specific needs to understand ontology changes to support mapping adaptation [5]. We assume that correctly identifying the defined CPs will support addressing the adaptation of ontology mappings [2].

Table 1. Description and examples of the proposed change patterns from attribute  $a_p^0$  to attribute  $a_q^1$ . The symbol  $\emptyset$  means that the corresponding attribute does not exist.

attribute	CP		CP type	example	
	$t_0$	$t_1$		$t_0$	$t_1$
$a_p$	ABC	ABC	total copy (TC)	'portal systemic encephalopathy'	'portal systemic encephalopathy'
$a_q$	ABC	ABC(D)		$\emptyset$	'portal systemic encephalopathy'
$a_p$	ABC	<del>ABC</del>	total transfer (TT)	'fecal impaction'	$\emptyset$
$a_q$	<del>ABC</del>	ABC(D)		$\emptyset$	'fecal impaction'
$a_p$	ABC	ABC	partial copy (PC)	'familial hyperchylomicronemia'	'familial hyperchylomicronemia'
$a_q$	ABC	AB(D)		$\emptyset$	'familial chylomicronemia'
$a_p$	ABC	<del>ABC</del>	partial transfer (PT)	'eye swelling'	$\emptyset$
$a_q$	<del>ABC</del>	AB(D)		$\emptyset$	'head swelling'

In what follows, we define  $W(a_i^j)$  as a set of words/tokens from  $a_i$ .value of an attribute  $a_i$ , and  $w_{ki}^j$  as a single word/token from an attribute value at time  $t_j$ . The  $sim(a_p^0, a_q^1)$  function refers to the similarity between the value of the attributes  $a_p^0 \in A(c_k^0)$  and  $a_q^1 \in A(c_{cand}^1)$ . The used similarity measure indicates the degree of relatedness between two given textual values. We use the  $\gamma$  param-

eter to control the overlap in terms of words between two attribute values. Since the performance of the similarity measure is not the focus of this paper, we keep it generic in our definition of CPs so that we can choose it as a parameter in our experiments. We formalize each type of CP between  $a_p$  and  $a_q$ , if any, as follows:

- **Total copy.** A total copy of content occurs between attribute  $a_p^0$  in concept  $c_k$  and  $a_q^1$  in concept  $c_{cand}$  if and only if a minimal degree  $\gamma$  of words in  $a_p$  appears in  $a_q$  and a minimal similarity value  $\tau$  exists between them. Formally:

$$TC(a_p^0, a_q^1) \Leftrightarrow \begin{cases} a_p^0 \in A(c_k^0) \\ c_k^1 \in C(O_x^1) \\ a_p^1 \in A(c_k^1) \\ sim(a_p^0, a_q^1) \geq \tau \\ \|W(a_p^0) \cap W(a_q^1)\| / \|W(a_p^0)\| \geq \gamma \end{cases} \quad (4)$$

- **Total transfer.** A total transfer of content occurs between attribute  $a_p^0$  in concept  $c_k$  and  $a_q^1$  in concept  $c_{cand}$  if and only if a minimal degree  $\gamma$  of words in  $a_p$  appears in  $a_q$  and a minimal similarity value  $\tau$  exists between them while the original attribute  $a_p^0$  is removed from  $c_k^1 \in O_x^1$ . Note that in total copy (cf. Equation 4)  $a_p^1 \in A(c_k^1)$  while in total transfer (cf. Equation 5)  $a_p^1 \notin A(c_k^1)$  which states the main difference between them. Formally:

$$TT(a_p^0, a_q^1) \Leftrightarrow \begin{cases} a_p^0 \in A(c_k^0) \\ a_p^1 \notin A(c_k^1) \\ sim(a_p^0, a_q^1) \geq \tau \\ \|W(a_p^0) \cap W(a_q^1)\| / \|W(a_p^0)\| \geq \gamma \end{cases} \quad (5)$$

- **Partial copy.** A partial copy of content occurs between attribute  $a_p^0$  in concept  $c_k$  and  $a_q^1$  in concept  $c_{cand}$  if and only if there exists a partial overlap between words constituting attributes  $a_p^0$  and  $a_q^1$ , while respecting a minimal similarity value  $\tau$  and a degree of overlap between 0 and  $\gamma$ . Formally:

$$PC(a_p^0, a_q^1) \Leftrightarrow \begin{cases} a_p^0 \in A(c_k^0) \\ c_k^1 \in C(O_x^1) \\ a_p^1 \in A(c_k^1) \\ \exists w_{ip}^0 \in W(a_p^0), w_{ip}^0 \in W(a_q^1) \\ \exists w_{jp}^0 \in W(a_p^0), w_{jp}^0 \notin W(a_q^1) \\ sim(a_p^0, a_q^1) \geq \tau \\ 0 \leq \|W(a_p^0) \cap W(a_q^1)\| / \|W(a_p^0)\| \leq \gamma \end{cases} \quad (6)$$

- **Partial transfer.** A partial transfer of content occurs between attribute  $a_p^0$  in concept  $c_k$  and  $a_q^1$  in concept  $c_{cand}$  if and only if there exists a partial overlap between words constituting attributes  $a_p^0$  and  $a_q^1$  while respecting a

minimal similarity value  $\tau$ , a degree of overlap between 0 and  $\gamma$ , and the original attribute  $a_p^0$  is removed from  $c_k^0 \in O_x$ . Formally:

$$PT(a_p^0, a_q^1) \Leftrightarrow \begin{cases} a_p^0 \in A(c_k^0) \\ a_p^1 \notin A(c_k^1) \\ \exists w_i^p \in W(a_p^0), w_i^p \in W(a_q^1) \\ \exists w_j^p \in W(a_p^0), w_j^p \notin W(a_q^1) \\ sim(a_p^0, a_q^1) \geq \tau \\ 0 \leq \|W(a_p^0) \cap W(a_q^1)\| / \|W(a_p^0)\| \leq \gamma \end{cases} \quad (7)$$

## 4 Recognizing change patterns related to attributes

In our approach to recognize change pattern, we first determine a candidate attribute  $a_q^1$  in the context of a concept  $c_k^1$  (Section 4.1). This candidate refers to a changed attribute at time  $t_1$  related to the attribute  $a_p^0$  in concept  $c_k^0$  that we used to identify occurrences of CPs (Section 4.2).

### 4.1 Candidate attribute in the context

We designed Algorithm 1 that explores textual attributes from a given concept at time  $t_0$ . In particular, given an attribute  $a_p^0 \in A(c_k^0)$  from  $O_x^0$ , the algorithm courses the whole set of changed attributes of the context of  $c_k$  at time  $t_1$  by calculating the similarity to retrieve candidate attributes. It aims to find the most adequate attribute in the context of the given one from  $A(c_k)$ , which we will use in Algorithm 2 to identify change patterns. We consider the types of comparable textual attributes as a parameter in our approach. For example, we can take only attributes of type “name” and “synonym” into consideration when comparing the attribute values (*i.e.*, strings denoting concepts). Our methods exclude all types of attributes out of the comparable set of attributes defined beforehand. The function  $sim(a_i^0, a_j^1)$  computes the similarity between two given attribute values. It returns a value ranging from 0 to 1. The higher the result is, the more similar these attributes are. We explore traditional string-based similarity metrics (the *bi-gram* measure), when calculating the similarity between attribute values in Algorithm 1. We selected this metric as the default similarity because it performs well on ontology matching [16].

Algorithm 1 generates a list of candidate attributes which is denoted as  $S_{cand}(a_p^0) = \{(a_{q_1}, sim_{pq_1}), (a_{q_2}, sim_{pq_2}), \dots, (a_{q_m}, sim_{pq_m})\}$ , where  $a_{q_i} \in A(CT(c_k^1))$  and  $sim_{pq_i} = sim(a_p^0, a_{q_i}^1)$ . In fact,  $S_{cand}(a_p^0)$  stores the candidate attributes along with their similarity with the attribute  $a_p^0 \in A(c_k^0)$ . This algorithm uses a ranking function to determine the best candidate attribute as a result.

We distinguish two ranking approaches to find the best candidate attribute: **global** and **local**. The candidate attribute may have a strong influence on the CP identification method which motivates us to investigate both rankings.

- **Global ranking (GR)**. In this ranking the best  $a_q^1$  candidate attribute (found at time  $t_1$ ) refers to the one that has the highest similarity with a

---

**Algorithm 1:** Find candidate attribute in the context
 

---

**Require:**  $a_p^0 \in A(c_k^0); CT(c_k^1) \subset C(O_x^1)$   
 $sim \leftarrow \emptyset; a_q^1 \leftarrow \emptyset; S_{cand} \leftarrow \emptyset;$   
**for all**  $c_i^1 \in CT(c_k^1)$  **do**  
   **for all**  $a_i^1 \in A(c_i^1)$  **do**  
     **if**  $a_i^0 \notin A(c_i^0) \vee a_i^0 \neq a_i^1$  **then**  
        $sim \leftarrow sim(a_p^0, a_i^1);$   
        $S_{cand} \leftarrow S_{cand} \cup \{(a_i^1, c_i^1, sim)\};$   
     **end if**  
   **end for**  
**end for**  
**return**  $S_{cand} \leftarrow rank(S_{cand}).first;$

---

given attribute  $a_p^0 \in A(c_k^0)$ . We denote this as the **global candidate** because the selection relies on the optimum similarity value considering the whole context. Formally:

$$candidate_{GR}(S_{cand}(a_p^0)) \leftarrow \arg \max_{a_{q_i}^1 \in A(CT(c_k^1))} \{sim(a_p^0, a_{q_i}^1)\} \quad (8)$$

- **Local ranking (LR).** Unlike the GR, this approach assumes that the best candidate attribute locates in a part of the evolving ontology where we observe most changes in attributes. LR executes two steps: (1) it analyzes which elements of the context of concept  $c$  (*i.e.*,  $sup(c)$ ,  $sub(c)$ ,  $sib(c)$ ) has the highest number of changed attributes; (2) based on this result, it selects the most similar attribute. We refer to this as the **local candidate** because the selection relies on the optimum similarity value considering part of the context. We compute the distribution of the different relationship types from the context in the list of changed attributes  $S_{cand}(a_p^0)$  as follows:

$$dist(S_{cand}(a_p^0), \Gamma) = \frac{\sum f(a_{q_i}^1)}{\|S_{cand}(a_p^0)\|} \quad (9)$$

- $\Gamma$  is among the three types of relationships we consider in  $CT(c)$  interconnecting super, sub and sibling concepts.
- $f(a_{q_i}^1)$  stands for the function counting the frequency of a particular relationship type, defined as follows:

$$f(a_{q_i}^1) = \begin{cases} 1 & \text{if } rel(a_{q_i}^1) = \Gamma \\ 0 & \text{otherwise} \end{cases}$$

where  $rel(a_{q_i}^1)$  refers to the type of relationship between concepts  $c_k^0$  denoted by attribute  $a_p^0$  and  $c_{cand}^1$  denoted by attribute  $a_{q_i}^1$ . We define the local candidate as follows:

$$candidate_{LR}(S_{cand}(a_p^0)) \leftarrow \begin{cases} best\_dist(\Gamma) \leftarrow \arg \max_{a_{q_i}^1 \in A(CT(c_k^1))} dist(S_{cand}(a_p^0), \Gamma) \\ \arg \max_{a_{q_i}^1 \in best\_dist(\Gamma) \wedge rel(a_{q_i}^1) = \Gamma} sim(a_p^0, a_{q_i}^1) \end{cases} \quad (10)$$



## 4.2 Identification method

Algorithm 2 describes the designed procedure to identify CPs. The best candidate  $c_{cand}^1$  refers to the concept denoted by attribute  $a_q^1$ , retrieved with algorithm 1. For each candidate  $a_q^1$ , the algorithm checks whether its similarity value with attribute  $a_p^0$  is greater or equal to a threshold  $\tau$ , and the conditions for applying each type of change pattern on the couple of attributes  $a_p^0$  and  $a_q^1$ . To this end, it calculates the number of common words between  $a_p^0$  and  $a_q^1$  by removing stop words from the original attributes. The algorithm also explores whether attributes  $a_p^0 \in A(c_k^0)$  and  $a_q^1 \in A(c_{cand}^1)$  remain at time  $t_1$  (*i.e.*, it is not removed). According to the definitions, the algorithm assigns the adequate CP. Given two versions of the same ontology, we can apply Algorithm 2 to all concepts placed in ontology regions affected by traditional change operations.

---

### Algorithm 2: Change pattern identification

---

**Require:**  $a_p^0 \in A(c_k^0); c_k^0 \in C(O_x^0); CT(c_k^1) \subset C(O_x^1)$   
 $CP \leftarrow \emptyset; sim \leftarrow 0; nbEqWords \leftarrow 0$   
 $a_q^1; sim \leftarrow \text{Algorithm 1}(a_p^0; CT(c_k^1));$   
**if**  $a_q^1 \neq \emptyset$  **then**  
  **if**  $0 < sim > \tau$  **then**  
     $nbEqWords \leftarrow \|W(a_p^0) \cap W(a_q^1)\|$   
    **if**  $nbEqWords / \|W(a_p^0)\| \leq \gamma \wedge nbEqWords > 0 \wedge nbEqWords < \|W(a_p^0)\|$   
      **then**  
        **if**  $a_p^1 \in A(c_k^1)$  **then**  
           $CP \leftarrow PC(a_p^0, a_q^1);$   
        **else**  
           $CP \leftarrow PT(a_p^0, a_q^1);$   
        **end if**  
      **else**  
        **if**  $nbEqWords / \|W(a_p^0)\| \geq \gamma$  **then**  
          **if**  $a_p^1 \in A(c_k^1)$  **then**  
             $CP \leftarrow TC(a_p^0, a_q^1);$   
          **else**  
             $CP \leftarrow TT(a_p^0, a_q^1);$   
          **end if**  
        **end if**  
      **end if**  
    **end if**  
  **end if**  
**return**  $(a_p^0, a_q^1, CP);$

---

## 5 Experimental evaluation

We present the used materials followed by the experimental procedure conducted to achieve the following objectives:

- We evaluate the effectiveness of the proposed methods for identifying change patterns based on exploiting lexical features of attributes.
- We assess the proposed ranking functions by comparing their performance.

### 5.1 Materials

In the conducted experiments we used various versions of three large biomedical ontologies: SNOMED-CT (SCT), MeSH and ICD-9-CM (ICD9). Table 2 presents statistics regarding the number of concepts, attributes and the number of direct subsumption relationships between concepts, since this study focused on exploiting the hierarchical structure of ontologies. SCT contains a much higher number of concepts than MeSH and ICD9. Table 2 also depicts the evolution of concepts and attributes for the three studied biomedical ontologies in a combined way over the last years. This dynamic evolution motivates us to use biomedical ontologies as a case study in this research.

**Table 2.** Evolution of biomedical ontologies. The numbers between parentheses represent the change rate between two releases of the same ontology.

ontology	year	#concepts	#attributes	#subsumptions
ICD-9-CM	2009	12734	34065	11619
	2011	13059 (+2.55%)	34963 (+2.64%)	11962 (+2.95 %)
SNOMED-CT	2010	386965	1531288	523958
	2012	395346 (+2.12%)	1570504 (+2.50%)	539245 (+2.83%)
MeSH	2012	50367	259565	59191
	2013	50971 (+1.18%)	264783 (+1.97%)	59844 (+1.09%)

**Reference change patterns.** To evaluate the effectiveness of our approach, we defined a set of reference change patterns as our standards. We needed to build our own set of reference since no available gold standard exists for the investigated context. To this end, we conducted the following steps:

- We combined the ontologies and we randomly selected a set of 1.000 couples of attributes. We defined the size of our sample in accordance with the involved experts taking into account their availability and scientific consistencies for our experiments. One attribute of a couple comes from a concept at time  $t_j$  and the other one in the same couple comes from a concept in the context of the former concept in the same ontology at time  $t_{j+1}$ . We chose these couples based on the similarity between attribute values, excluding attributes with very low similarity and unchanged attributes at time  $t_{j+1}$ .
- We invited three ontology engineering experts to evaluate all selected couples of attributes to assign their answer regarding CPs. For this purpose, we supported them with a software tool suited to present additional information regarding each attribute. This tool presents the couple of attributes along with concepts in the context, the attributes denoting concepts as well as the changes affecting them, *etc.* We gave instructions on the purpose of the different patterns, and recorded the answers for each evaluator separately.

- The evaluators performed one round of evaluation, and we merged the agreement answers. The domain experts collaborated and re-evaluated a second round only with the disagreement part of couples. We merged the final agreement couples with the respective correct answers according to the evaluators. We achieved an average agreement rate of 86%. Finally, we retained 675 pairs of attributes that had the consent from all evaluators for our experiments.

## 5.2 Experimental procedure

For evaluating the effectiveness of our CP identification algorithm, we computed the standard metrics of *Precision*, *Recall* and *F-measure* based on the reference CPs as input. Specifically, we computed the precision as the number of CPs correctly identified by the algorithm over the total number of identified CPs. Recall was computed as the number of correctly identified CPs over the total number of relevant/expected CPs in the set of reference. F-measure was computed as the harmonic mean of precision and recall.

We investigated the influence of the similarity threshold in the CP identification algorithm. For this purpose, we analyzed the CP identification performance by varying the similarity threshold from 0 to 1 to observe the performance of our algorithm, and we set  $\tau = \gamma$ . Additionally, we examined the quality of the outcomes by comparing both GR and LR ranking functions proposed.

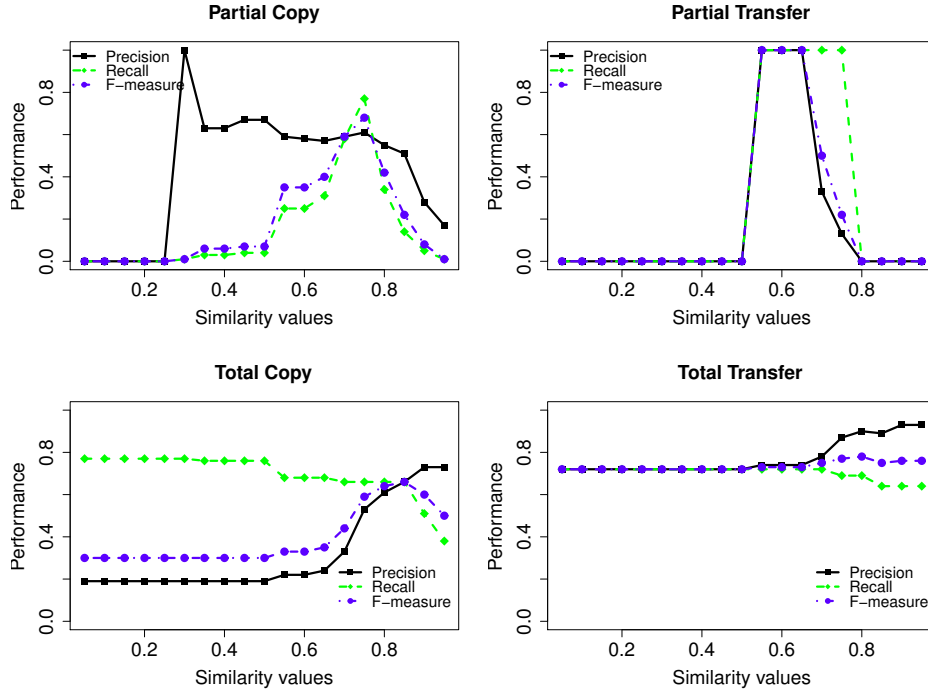
## 5.3 Results

Figure 2 presents the effectiveness of the CP identification algorithm in terms of precision, recall and F-measure by varying the similarity thresholds (denoted as  $\tau$ ). We achieved these results using the global ranking in Algorithm 1.

The performance of this algorithm varies according to the value of  $\tau$ . Overall, the F-measure is greater than 0.60 for all types of CP. We observe that the similarity threshold plays a relevant role in CP identification because its performance dramatically changes when the threshold is set very low (*e.g.*,  $\tau < 0.5$ ). Our CP identification algorithm reaches the best performance with thresholds ranging from 0.7 to 0.9, which points out the necessity of having a minimal similarity between attributes to boost the identification results.

By observing the results for each type of CP, we found that the identification of partial copy CP reaches the highest F-measure of 0.68 (precision=0.61, recall=0.77) at  $\tau = 0.75$ . This remains similar to the case of total copy CP, where the highest F-measure is 0.66 (precision=0.66, recall=0.66) at  $\tau = 0.85$ . Moreover, the recall for total copy CP tends to be higher than the precision for  $\tau < 0.85$ , but we observed the contrary phenomenon for partial copy CP. We potentially explain this by the fact that for correctly identifying total copy CPs require higher similarity between attributes, while for partial copy, the higher the similarity value, the lower the number of partial copy CPs correctly identified.

Regarding total transfer CP, Algorithm 2 reaches the best F-measure at 0.78 (precision=0.90, recall=0.69) for  $\tau = 0.80$ . The algorithm performs better on identifying total transfer than on partial transfer. Partial transfer CP seems a



**Fig. 2.** Effectiveness of the CP identification method (using GR ranking) in terms of precision, recall and f-measure for the different types of change patterns.

particular case (not frequently found) because evaluators assigned only one case in the reference change patterns.

To analyze the results comparing the proposed rankings, we retain the maximum value of the similarity threshold, *i.e.*,  $\tau = 0.85$ , that optimizes the performance of the GR (denoted as  $baseline_{MAX}$ ). Tables 3 and 4 present the achieved results for transfer and copy CPs by running our method using the local ranking function. We chose the similarity thresholds among the values in the set  $\{0.2, 0.4, 0.6, 0.75, 0.8\}$  to analyze the performance, and compare the difference in terms of precision, recall and F-measure between LR and GR.

Results in Table 3 reveal that the LR improves the performance in terms of recall for total transfer with a maximum improvement rate of +4.69%, while the performance dramatically decreases for precision and F-measure. For partial transfer CP, the LR method outperforms the baseline for values of  $\tau$  in the interval  $[0.6, 0.75]$ . The performance of the latter is zero for either precision, recall and F-measure, probably because the similarity threshold was very high ( $\tau = 0.85$ ). This suggests that CP identification for either partial or total transfer must use a flexible or approximate string matching with an appropriate similarity threshold that should not be very low (*e.g.*,  $< 0.5$ ) nor very high (*e.g.*,  $> 0.8$ ).

Regarding the performance of the local ranking for identifying copy CPs (cf. Table 4), we observe that for total copy CP, the precision is proportional to the similarity threshold while the recall is not. For partial copy CP, the local ranking shows a significant improvement rate of +29.41 % for precision, +378% for recall and +204% for F-measure by using  $\tau = 0.75$ .

**Table 3.** Performance of the identification method by using the LR ranking for transfer CP. Numbers in parentheses correspond to the difference between P, R, F obtained by LR comparatively to  $\tau$  of the  $baseline_{MAX}$  of GR.

$\tau$	CP	Transfer of attributes					
		total			partial		
		P	R	F	P	R	F
		( $\Delta P$ )	( $\Delta R$ )	( $\Delta F$ )	( $\Delta P$ )	( $\Delta R$ )	( $\Delta F$ )
	$baseline_{MAX}$	0.89	0.64	0.75	0.00	0.00	0.00
	<b>0.2</b>	0.50 (-43.82%)	0.67 (+4.69%)	0.57 (-24.00%)	0.00 (0.00%)	0.00 (0.00%)	0.00 (0.00%)
	<b>0.4</b>	0.50 (-43.82%)	0.67 (+4.69%)	0.57 (-24.00%)	0.00 (0.00%)	0.00 (0.00%)	0.00 (0.00%)
	<b>0.6</b>	0.53 (-40.45%)	0.67 (+4.69%)	0.59 (-21.33%)	0.33 (INF)	1.00 (INF)	0.50 (INF)
	<b>0.75</b>	0.79 (-11.24%)	0.67 (+4.69%)	0.72 (-4.00%)	0.05 (INF)	1.00 (INF)	0.10 (INF)
	<b>0.8</b>	0.79 (-11.24%)	0.67 (+4.69%)	0.72 (-4.00%)	0.00 (0.00%)	0.00 (0.00%)	0.00 (0.00%)

**Table 4.** Performance of the identification method by using the LR ranking for copy CP. Numbers between parentheses correspond to the difference between P, R, F obtained by LR comparatively to  $\tau$  of the  $baseline_{MAX}$  of GR.

$\tau$	CP	Copy of attributes					
		total			partial		
		P	R	F	P	R	F
		( $\Delta P$ )	( $\Delta R$ )	( $\Delta F$ )	( $\Delta P$ )	( $\Delta R$ )	( $\Delta F$ )
	$baseline_{MAX}$	0.66	0.66	0.66	0.51	0.14	0.22
	<b>0.2</b>	0.18 (-72.73%)	0.65 (-1.52%)	0.28 (-57.58%)	0.00 (-100.00%)	0.00 (-100.00%)	0.00 (-100.00%)
	<b>0.4</b>	0.18 (-72.73%)	0.63 (-4.55%)	0.28 (-57.58%)	0.63 (+23.53%)	0.03 (-78.57%)	0.06 (-72.73%)
	<b>0.6</b>	0.21 (-68.18%)	0.58 (-12.12%)	0.31 (-53.03%)	0.56 (+9.80%)	0.20 (+42.86%)	0.30 (+36.36%)
	<b>0.75</b>	0.53 (-19.70%)	0.56 (-15.15%)	0.55 (-16.67%)	0.66 (+29.41%)	0.67 (+378%)	0.67 (+204%)
	<b>0.8</b>	0.56 (-15.15%)	0.41 (-37.88%)	0.47 (-28.79%)	0.61 (+19.61%)	0.28 (+100%)	0.39 (+77.27%)

## 5.4 Discussion

We found that the suggested types of CPs at the level of attributes can be observed in real cases of ontology evolution. These CPs refine the traditional ones at a finer level of granularity to characterize ontology evolution. Overall results pointed out the effectiveness of the proposed method underlaid by similarity measure and intersection of words between attribute values to recognize CPs between ontology versions. We demonstrated that the similarity threshold plays an important role in the quality of the outcomes. We explain this by the fact that our method selects the candidate attribute based on the similarity that is proportional to the degree of relatedness between the analyzed attributes.

When comparing the overall performance of CP identification under the GR and LR methods, our findings demonstrated that considering both the types of context relationships as well as their distribution affect the identification results. The LR method performs better, in particular for partial copy with a significant improvement compared to the GR. However, for total copy and transfer the performance under LR remains low, probably because the nature of these CPs requires a relatively high similarity threshold. We conclude that CPs of partial type should base on the local ranking for selecting the candidate attribute in CP recognition, while for CPs of total type we recommend using the global ranking.

Our scholarly obtained findings have revealed evidences of the quality of the results that were yielded by the proposed method, relying on standard evaluation metrics. In addition, we conducted experiments using real biomedical ontologies which strengthens our results.

## 6 Conclusion

Ontology evolution requires further means to describe specific changes at different entities. This plays a relevant role in controlling the impact of changes on dependent artefacts. In this article, we defined change patterns of concept attributes to characterize the evolution of their textual values. We designed a novel method to recognize the change patterns between ontology versions and empirically evaluated our proposition by observing the evolution of biomedical ontologies. We studied the influence of different aspects in the change pattern identification on the quality of the outcomes. The achieved results showed evidences of the performance of the proposed method.

In addition to existing and traditional ontology change operations, our contribution in this article originally allows to characterize ontology evolution by means of change patterns at attribute level. This stands for fine-grained changes that may facilitate tasks related to the impact of ontology evolution such as mapping and annotation maintenance. As future work, we aim to study techniques to recognize the way attribute values become more or less semantically specific in ontology evolution, and to investigate to which extent the different types of change patterns may influence the way ontology mappings evolve.

## Acknowledgment

The National Research Fund (FNR) of Luxembourg entirely supports this work under the *DynaMO* research project (Grant #C10/IS/786147).

## References

1. Groß, A., Dos Reis, J., Hartung, M., Pruski, C., Rahm, E.: Semi-automatic adaptation of mappings between life science ontologies. In: Proc. of the Conference on Data Integration in the Life Sciences. (2013) 90–104
2. Dos Reis, J.C., Dinh, D., Pruski, C., Da Silveira, M., Reynaud-Delaitre, C.: Mapping adaptation actions for the automatic reconciliation of dynamic ontologies. In: Proc. of Conference on Information and Knowledge Management. (2013) 599–608
3. Hartung, M., Groß, A., Rahm, E.: Conto-diff – generation of complex evolution mappings for life science ontologies. *Biomedical Informatics* (2013) 15–32
4. Dos Reis, J.C., Pruski, C., Da Silveira, M., Reynaud-Delaitre, C.: Understanding semantic mapping evolution by observing changes in biomedical ontologies. *Journal of Biomedical Informatics* **47** (2014) 71–82
5. Dos Reis, J., Pruski, C., Da Silveira, M., Reynaud-Delaitre, C.: Characterizing semantic mappings adaptation via biomedical kos evolution: A case study investigating snomed ct and icd. In: Proc. of the AMIA Symposium. (2013) 333–342
6. Noy, N.F., Klein, M.: Ontology evolution: Not the same as schema evolution. *Knowledge and information systems* **6**(4) (2004) 428–440
7. Javed, M., Abgaz, Y., Pahl, C.: Ontology change management and identification of change patterns. *Journal on Data Semantics* **2**(2-3) (2013) 119–143
8. Noy, N.F., Musen, M.A.: PROMPTDIFF: A Fixed-Point Algorithm for Comparing Ontology Versions. In: Proc. of AAAI 2002. (2002) 744–750
9. Kremen, P., Smid, M., Kouba, Z.: Owldiff: A practical tool for comparison and merge of owl ontologies. In: 22nd International Workshop on Database and Expert Systems Applications (DEXA). (2011) 229–233
10. Shaban-Nejad, A., Haarslev, V.: Bio-medical ontologies maintenance and change management. In Sidhu, A., Dillon, T., eds.: *Biomedical Data and Applications*. Springer Berlin Heidelberg (2009) 143–168
11. Gröner, G., Parreiras, F.S., Staab, S.: Semantic Recognition of Ontology Refactoring. In: Proc. of the 9th International Semantic Web Conference. ISWC’10. Springer-Verlag, Berlin, Heidelberg (2010) 273–288
12. Auer, S., Herre, H.: A Versioning and Evolution Framework for RDF Knowledge Bases. In: Proc. of the 6th International Andrei Ershov Memorial Conference on Perspectives of Systems Informatics. PSI’06, Springer-Verlag (2007) 55–69
13. Rieß, C., Heino, N., Tramp, S., Auer, S.: EvoPat - Pattern-based Evolution and Refactoring of RDF Knowledge Bases. In: Proc. of the 9th International Semantic Web Conference. ISWC’10, Berlin, Heidelberg, Springer-Verlag (2010) 647–662
14. Djedidi, R., Aufaure, M.A.: ONTO-EVOAL an Ontology Evolution Approach Guided by Pattern Modeling and Quality Evaluation. In: Proc. FoIKS’10, Berlin, Heidelberg, Springer-Verlag (2010) 286–305
15. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies* **43** (1995) 907 – 928
16. Cheatham, M., Hitzler, P.: String similarity metrics for ontology alignment. In Alani, H., Kagal, L., *et. al.*, eds.: *ISWC 2013*. Volume 8219 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2013) 294–309