Updating Ontology Alignment on the Relation Level based on Ontology Evolution

Adrianna Kozierkiewicz[a] and Marcin Pietranik[b]
Faculty of Computer Science and Management, Wroclaw University of Science and Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland

Keywords: Ontology Alignment, Ontology Evolution, Knowledge Management.

Abstract: Ontologies are becoming a popular and convenient way for knowledge representation - they can store information about objects and relations between them. However, nothing is constant and new information may appear, therefore those alterations should be reflected both in an ontology as well as in alignment between two ontologies if the knowledge about some domain is distributed in many sources. In the literature, it is possible to find approaches devoted to tracking changes in ontologies, but tools for updating ontology alignment are limited, especially devoted to the level of relations. It became a motivation for this work, thus, the aim of this paper is split into two parts. Firstly, we will present a criterion that tells us that modification in an ontology on the relation level is significant, and how it influences the maintained alignment (also on the relation level). Next, an algorithm for simple revalidation of existing mappings is proposed.

1 INTRODUCTION

Ontology alignment is a widely discussed and researched topic. It addresses a seemingly simple problem of designating a mapping between two heterogeneous ontologies. Such mapping can be used to "translate" content of one ontology to the content of some other ontology. This possibility is invaluable when some kind of communication of two independently developed information systems is expected. One cannot expect that such a system would utilize a shared ontology as a backbone of their knowledge bases. Different systems have different business requirements and enforcing a common ontology would eventually make it nearly impossible to maintain ((Abbes and Gargouri, 2017), (Wang et al., 2020)).

Designating a bridge between two ontologies (although simple to understand) is a very difficult task in terms of both its semantic and computational complexity. Informally speaking the task is to select parts of two (or more) ontologies that express the same parts of a modeled universe of discourse. In the literature, it is easy to find a plethora of different procedures that address this problem ((Algergawy et al., 2018)). However, in modern applications ((Kiourtis et al., 2019)) no one can expect that the underlying ontologies will not change in time. Such a situation may potentially result in invalidation of the designated alignment, which results in breaking the communication between interacting information systems. This is especially visible on the level of relations, which is frequently omitted. None of the analyzed publications (which is presented in further parts of the article) focus strictly on relations neither in the context of their evolution, neither on the level of their mappings.

For the better explanation of the considered in this paper, problem let us follow Figure 1 which is an easy example of evolving ontology and presents two ontologies $O_2^{(n)}$ and $O-1$ in a state $m-1$ and m. On the upper side of the picture between ontologies O_1 and O_2 the alignment on the relational level has been designated. In both ontologies, the relation *look after* appears and mapping between them is obvious. Relation *is mother* is only less general than relation *is parent* and between them also mapping has been detected. On the bottom side of the picture ontology O_1 has been evolved. The relation *look after* has been deleted which influences the current alignment (the mapping can no longer exist). The new relation *has* has appeared, however, it can not be connected with any relation from ontology O_2. The relation *is parent*

[a] https://orcid.org/0000-0001-8445-3979
[b] https://orcid.org/0000-0003-4255-889X

Kozierkiewicz, A. and Pietranik, M.
Updating Ontology Alignment on the Relation Level based on Ontology Evolution.
DOI: 10.5220/0009142002410248
In *Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2020)*, pages 241-248
ISBN: 978-989-758-421-3

has been changed by the concept modification, however, it has not affected the alignment.

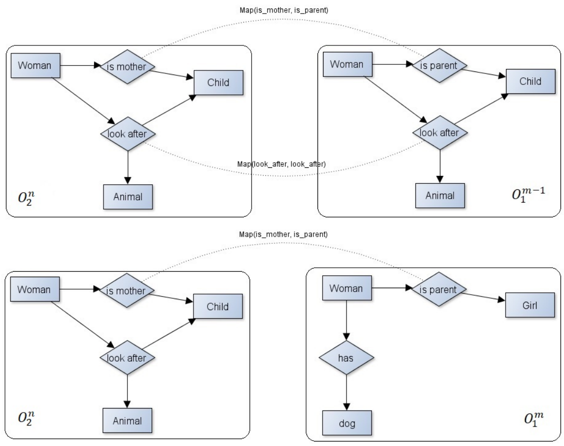

Figure 1: Examples of Ontologies Evaluation.

The obvious solution for the above problem is to relaunch the accepted mapping procedure from scratch for updated ontologies whenever they change. Such an approach would give good results but entails bearing the cost enforced by the mapping procedure. In the article, we propose a different approach. We claim that it is possible to update the possessed alignment between two ontologies based solely on the analysis of the applied changes. This task can be further decomposed into two elements.

The first of the aforementioned subproblems is deciding whether or not the maintained alignment should be checked for its validity. We claim that not all alterations that are introduced to some ontology during its evolution are equally significant and not all of them result in a situation in which some elements of the alignment become stale or invalid. For example, some small changes of some concept's label are not as influential as a major update of its attributes and relations it is connected by with other concepts. Since in this paper we are addressing only the level of relations, formally, this task can be described as follows: *For a given ontology O in its two consecutive states in time, denoted as $O^{(m)}$ and $O^{(m+1)}$, one should determine a function Ψ_R representing the degree of significance to which relations within it have been changed in time.*

The second part is performing the actual revalidation. This task involves solving three problems: *(i)* deleting stale mappings, *(ii)* revalidating and updating existing mappings, and *(iii)* adding new mappings. Formally, it can be defined as: *For a given source ontology $O_1^{(m)}$ in a moment in time denoted as m, a target ontology $O_2^{(n)}$ in a moment in time denoted as n, and an alignment of their relations denoted as $Align(O_1^{(m)}, O_2^{(n)})$, one should provide algorithms which can update this alignment if the source ontology significantly evolves from the state $O_1^{(m)}$ to the state $O_1^{(m+1)}$ according to applied changes.*

The article is structured as follows. In Section 2 an overview of related works is given. Section 3 contains basic notions and definitions utilized in the paper. Section 4 provides the main contribution. It has been split into two subparts. The first contains a definition of the developed function Ψ_R that can be used to indicate whether or not the alignment of relations potentially needs revalidating. The second part provides algorithms for such revalidation. The developed procedures have been experimentally verified - the obtained are described in Section 5. Our upcoming research and a summary are given in Section 6.

2 RELATED WORKS

It is possible to find many ontology alignment tools, however, their efficiency may highly differ. The choice of the best alignment tool is supported since 2004 by the Ontology Alignment Evaluation Initiative (OAEI). OAEI coordinates an international initiative to evaluate, compare and improve the tools for ontology mapping and alignment by providing standardized benchmark datasets.

Strong competitors of multiple OAEI editions were tools like ALIN (Da Silva Jomar and Kate, 2018), AML (Faria et al., 2018), Kepler (Kachroudi et al., 2018), LogMap (Jiménez-Ruiz et al., 2018), SANOM (Mohammadi et al., 2018). However, the of the alignment system of object properties matching, which can be understood as relations in terms of OWL representation ((Hitzler et al., 2009)), lags significantly behind that on class and instance matching (Cheatham et al., 2018). The average precision and recall measures for class alignment obtained by the top 2016 OAEI competitors on the Conference track equal 0.82 and 0.65, respectively. For object properties alignment these measures are severely lower: 0.45 and 0.1, respectively.

The presented results in (Cheatham et al., 2018) allow us to conclude that the accuracy of the verified tools is almost three times worse for data and object property in comparison with class alignment. Additionally, some of the presented systems do not generate any matches involving object properties at all. This means that the alignment system' designers focus their efforts mainly on class and instance level. Therefore, the problem of determining reliable mappings for object properties is still open.

The systems mentioned above, allow the user to determine a new alignment for given ontologies. This paper is devoted to updating alignment in case of evolution input one or more ontologies. It is rather obvious, that we can use tools directly dedicated to an ontology mapping and perform a process of determining the alignment from the beginning for the modified ontologies. Such a solution is obviously cost- and time-consuming. In the literature, it is possible to find some approaches which track ontology evolution (i.e. (Abbes and Gargouri, 2017), (Noy et al., 2002)), however, not many solutions are devoted to updating the predesignated mappings.

Some of the tools for updating ontology alignments focus only on part of ontologies, especially on concepts level. In (Dos Reis and Yamamoto, 2019), (Dos Reis, 2018) authors proposed techniques to refine a set of established mappings based on the evolution of ontologies. Their algorithms process the context of new concepts in both ontologies to find new matches between concepts. In other words, they suggest new correspondences with the new version of the ontology without applying a matching operation to the whole ontologies.

Similarly in (Thenmozhi and Vivekanandan, 2012), the authors proposed a new approach which is a semi-automatic way update mapping. The ontology was constantly monitoring for any changes and updating in the log, however, only information concerning concepts are stored. Next, the proposed solution removes staleness from the alignment based on predefined rules.

It is possible to find complex tools for updating ontology alignments like (Khattak et al., 2015). The authors proposed an approach that employs the analysis of ontology change history. These logged changes are later used with extensions to the existing mapping systems during the reconciliation of the mapping process. The three change types have been distinguished like create (such as ClassAddition, PropertyAddition, and IndividualAddition), update (such as ClassRenaming, PropertyRenaming, and IndividualRenaming) and delete (such as ClassDeletion, PropertyDeletion, and IndividualDeletion).

In (Hartung et al., 2013) COnto-Diff tool was described. Although COnto-Difff determined an expressive and invertible diff evolution mapping between two versions of the same ontology, authors noticed, that it can also be used to semi-automatically adapt annotations and ontology mappings after ontology modifications. The solution took into account the insertion and deletion of relationships.

In (Dos Reis et al., 2013) a set of mapping adaptation actions to maintain mappings up-to-date based on ontology change operations of different natures have been proposed. Two types of ontology change operations: atomic and complex have been defined. Based on them an overall adaptation of mappings according to the revision of knowledge in ontology was proposed.

Noteworthy is the fact, that all of the described methods concentrate on the concept level. This has two implications. First, the level of relations is frequently omitted. None of the analyzed publications focus strictly on relations neither in the context of their evolution, neither on the level of their mappings. Secondly, overviewed solutions of ontology mapping evolution look inside of the content of evolving source concepts, and not holistically analyze ontologies. Ontologies have dynamic nature, which directly impacts mappings established between concepts, relations, and instances from different ontologies. Although many pieces of research are aware of the need to track the changes in ontologies, the problem of maintaining the actual mappings between input ontologies is still open. These remarks became a motivation behind this paper, which addresses the problem of updating ontology alignment strictly on the relation level.

3 BASIC NOTIONS

The solution proposed in this paper for updating ontology alignment is based on our formal model of an ontology defined as a quintuple:

$$O = (C, H, R^C, I, R^I) \tag{1}$$

where: C is a set of concepts; H is a concepts' hierarchy; R^C is a set of relations between concepts $R^C = \{r_1^C, r_2^C, ..., r_n^C\}$, $n \in N$ (natural numbers), such that $r_i^C \in R^C$ ($i \in [1,n]$) is a subset of $C \times C$; I is a set of instances' identifiers; $R^I = \{r_1^I, r_2^I, ..., r_n^I\}$ is a set of relations between concepts' instances.

In the following article we focus on the relation level, therefore, provided definitions concern only elements related to this level. By D_R we define a set containing atomic descriptions of relations. Subsequently, we define a sub-language of the sentence calculus built from elements of D_R and logic operators of conjunction, disjunction and negation. We denote it as L_s^R. It is used within a function that assigns semantics to relations from the set R^C. This function has the following signature:

$$S_R : R^C \to L_s^R \tag{2}$$

As a consequence, we can define formal criteria for relationships between relations:

- *equivalency* between relations r and r' (denoted as $r \equiv r'$) occurs only if a sentence $S_R(r) \iff S_R(r')$ is a tautology
- a relation r' is more general than the relation r (denoted as $r' \leftarrow r$) if a sentence $S_R(r) \implies S_R(r')$ is a tautology
- *contradiction* between relations r and r' (denoted as $r \sim r'$) occurs only if a sentence $\neg(S_R(r) \wedge S_R(r'))$ is a tautology

The formal definition of an ontology on the relation level allows us to track changes in the ontology. The ontology evolution is based on a notion of a timeline, which can be treated as an ordered set of discrete moments in time. It can be defined as $\overline{TL} = \{t_n | n \in N\}$. $TL(O)$ is a subset of this timeline, containing only those moments from $\overline{TL}$ during which the ontology O has been changed. By $O^{(m)} = (C^{(m)}, H^{(m)}, R^{C(m)}, I^{(m)}, R^{I(m)})$ we denote the ontology O in a given moment in time $t_m \in TL(O)$. ($O^{(m-1)} \prec O^{(m)}$ means that $O^{(m)}$ is a subsequent version of O than $O^{(m-1)}$.

In order to compare two states of a single ontology O in our previous publication (Kozierkiewicz and Pietranik, 2019) we introduced a function $diff_R$ which, when fed with two successive states $O^{(m-1)}$ and $O^{(m)}$ of one single ontology (such that $O^{(m-1)} \prec O^{(m)}$), generates three sets containing relations added, deleted and altered. Formally, these sets are defined below:

1. $new_{R^C}(R^{C(m-1)}, R^{C(m)}) = \left\{ r | r \in R^{C(m)} \wedge r \notin R^{C(m-1)} \right\}$
2. $del_{R^C}(R^{C(m-1)}, R^{C(m)}) = \left\{ r | r \in R^{C(m-1)} \wedge r \notin R^{C(m)} \right\}$
3. $alt_{R^C}(R^{C(m-1)}, R^{C(m)}) = \left\{ (r^{(m-1)}, r^{(m)}) | r^{(m-1)} \in R^{C(m-1)} \wedge r^{(m)} \in R^{C(m)} \wedge (r^{(m-1)} \oplus r^{(m)} \neq \phi \vee S_R(r^{(m-1)}) \neq S_R(r^{(m)})) \right\}$

The first two descriptors are self-explanatory. The last one represents a situation where some pairs of concepts within such relation have been added or removed or some alterations to a semantic of relations expressed using the defined function S_R ($\oplus$ represent the exclusive or operator). For a detailed description of the holistic approach to managing ontologies in time please refer to (Kozierkiewicz and Pietranik, 2019).

Having two ontologies $O_1 = (C_1, H_1, R^{C_1}, I_1, R^{I_1})$ and $O_2 = (C_2, H_2, R^{C_2}, I_2, R^{I_2})$ a relations alignment between them is a set:

$$AL_R(O_1, O_2) = \{(r_1, r_2, \lambda_R(r_1, r_2), \tilde{r}) | \; r_1 \in R^{C_1} \wedge r_2 \in R^{C_2} \wedge \lambda_R(r_1, r_2) \geq T_R\} \quad (3)$$

where: λ_R is a degree to which relation r_1 can be aligned to relation r_2, $\tilde{r}$ represents the type of mapping (equivalency, generalization etc.) and T_R is some assumed threshold. For short we write $(r_1, r_2) \in AL_R(O_1, O_2)$ to indicate that a relation r_1 can be aligned to relation r_2 to some degree higher that the assumed threshold T_R by equivalency relationship.

The set from Equation 3 fulfills the requirement for alignment completeness which states that if some relation r_1 taken from the first ontology O_1 is more general than some other relation r_2, then they can be both mapped to a concept r' from the second ontology O_2. Formally it can be defined as follows:

$$\neg\exists (r_1, r_2) \in R^{C_1} \times R^{C_2} : \lambda_R(r_1, r_2) \geq T_R \; \wedge (r_1, r_2, \lambda_R(r_1, r_2), \tilde{r}) \notin AL_R(O_1, O_2) \quad (4)$$

4 UPDATING ONTOLOGY ALIGNMENT ON A RELATION LEVEL

4.1 The Degree of Change Significance on the Ontology Relation Level

Having an ontology O in its two subsequent states $O^{(m-1)}$ and $O^{(m)}$, such that $O^{(m-1)} \prec O^{(m)}$, and a relations difference function $diff_R$ we can define **the degree of significancy to which relations within a given ontology have been changed in time**. This function utilizes a function d_s which takes as an input two logical sentences and returns a distance between them based on the analysis of their elements taken from the set D_R defined in the previous section. Formally, it can be defined as follows:

$$\Psi_R : R^{C(m-1)} \times R^{C(m)} \to [0,1] \quad (5)$$

$$\Psi_R(R^{C(m-1)}, R^{C(m)}) = \frac{|new_{R^C}(R^{C(m-1)}, R^{C(m)})| + |del_{R^C}(R^{C(m-1)}, R^{C(m)})|}{|R^{C(m)}| + |del_{R^C}(R^{C(m-1)}, R^{C(m)})|} + \frac{\sum\limits_{(r_1, r_2) \in alt_{R^C}(R^{C(m-1)}, R^{C(m)})} d_s(S_R(r_1), S_R(r_2))}{|R^{C(m)}| + |del_{R^C}(R^{C(m-1)}, R^{C(m)})|} \quad (6)$$

The function above meets the following two postulates:

- **P1.** $\Psi_R(R^{C(m-1)}, R^{C(m)}) = 0 \iff diff_R(R^{C(m-1)} \times R^{C(m)}) = \left\langle \phi, \phi, \phi \right\rangle$
- **P2.** $\Psi_R(R^{C(m-1)}, R^{C(m)}) = 1 \iff del_R(R^{C(m-1)}, R^{C(m)}) = R^{C(m-1)} \wedge \wedge\, alt_R(R^{C(m-1)}, R^{C(m)}) = \phi$

P1 addresses a situation in which no alterations on the relation level have been applied - no new relations appeared, no relation was removed, no relation changed. Then, the significance of a change is minimal. **P2** describes an opposite situation, in which the change significance is maximal. It occurs when the ontology has been completely modified on the relation level - every relation from the earlier state has been deleted and every relation in a later state is new.

Having the function defined in Equation 6 it is straightforward to confront it with some assumed threshold. It allows deciding whether or not the alignment designated between the tracked ontology and some other ontology needs revalidation. If such a necessity appears, algorithms presented in the next section should be utilized.

4.2 Procedures for Updating Ontology Alignment on Relation Level

Having two ontologies $O_1^{(m-1)} = (C_1^{(m-1)}, H_1^{(m-1)}, R^{C_1(m-1)}, I_1^{(m-1)}, R^{I_1(m-1)})$ and $O_2^{(n)} = (C_2^{(n)}, H_2^{(n)}, R^{C_2(n)}, I_2^{(n)}, R^{I_2(n)})$, a relations alignment between them given as a set $AL_R(O_1^{(m-1)}, O_2^{(n)})$, a description of updates applied to O_1 given as a function $diff(O_1^{(m-1)}, O_1^{(m)})$, an algorithm that updates the alignment $AL_R(O_1^{(m-1)}, O_2^{(n)})$ to its new state $AL_R(O_1^{(m)}, O_2^{(n)})$ can be split into three separate scenarios:

1. deleting stale mappings
2. revalidating and updating existing mappings
3. adding new mappings

The first procedure presented on Algorithm 1 is very simple. It addresses the most basic situation when some relations have been removed from the source ontology. In this case, all related mappings (designated in Line 2) should also be discarded (which is done in Line 3), because they connect something which doesn't exist anymore.

The last two scenarios both end with searching for new mappings (due to adding new relations to the source ontology or modifying existing relations in such a way that new mappings may be added). Due to this, both can be implemented within the same procedure presented on Algorithm 2. It starts with accepting the unmodified alignment of ontologies in their earlier states (*m-1* and *n*) as its initial state (Line 2). Then, (Line 3) it designates a set of alignments that refer to relations modified in the later state (*m*). Elements of this set are later confronted (Line 4) with some assumed threshold - if it is not exceeded then such element is removed from the alignment. Subsequently, in Line 5, the algorithm generates a set of altered relations from the source ontology that was not mapped into any relation in the target ontology. This set is then combined with a set of relations newly added to the source ontology (Line 6).

Algorithm 1: Removing Stale Mappings of Deleted Relations from the Existing Alignment.

Input : $AL_R(O_1^{(m-1)}, O_2^{(n)}), diff_R(O_1^{(m-1)}, O_1^{(m)})$
Output: $AL_R(O_1^{(m)}, O_2^{(n)})$

1 **begin**
2 $\quad \widetilde{del} := \{(r_1, r_2, \lambda_R(r_1, r_2), \tilde{r}) | (r_1, r_2, \lambda_R(r_1, r_2), \tilde{r}) \in AL_R(O_1^{(m-1)}, O_2^{(n)}) \wedge r_1 \in del_R(R^{C_1(m-1)}, R^{C_1(m)})\}$;
3 $\quad AL_R(O_1^{(m)}, O_2^{(n)}) := AL_R(O_1^{(m-1)}, O_2^{(n)}) \setminus \widetilde{del}$;
4 $\quad$ **return** $AL_R(O_1^{(m)}, O_2^{(n)})$;
5 **end**

Eventually, the algorithm enters (Line 7) a subprocedure that finds new potential mappings. At this point it is possible to use any kind of available alignment procedure. The algorithm is fully agnostic and this step is easily swappable. However, for the demonstration purposes, in this paper we use an alignment procedure developed in our previous publications ((Pietranik and Nguyen, 2014)) that is based on comparing relation semantics from Equation 2 using a d_s distance function that is also used in Equation 6. Basically, the algorithm traverses through the set of relation available in the target ontology (Line 8) and checks if the degree to which a relation from the source ontology can be aligned to the relation from the target ontology exceeds the assumed threshold (Line 9). If yes, then the new mapping is added to the final alignment (Line 10). The sub-procedure ends with complementing the relation alignment to fulfill the completeness postulate from Equation 4. In Line 13 the set of relations that are less or more general than the current relation is designated. Then, in Line 14, the procedure checks if the current relation has any mappings. If this is the case, then appropriate mappings are added for related relations (Line 15). This step is done to avoid searching for mappings for relations that are connected with a generalization relationship with the current one and to automatically add them to the resulting set of mappings. Therefore, omitting unnecessary calculations.

The presented procedure is quite complex in terms of a number of iterations. However, ontology alignments usually do not contain many mappings connecting relations (as shown in Section 2). These sets are frequently very limited in terms of their size. It is not uncommon that they contain only a few mappings (or even none), therefore, the presented procedure for small alterations of the source ontology can become very useful.

Algorithm 2: Revalidating the Existing Alignment and Adding New Mappings.

Input : $AL_R(O_1^{(m-1)}, O_2^{(n)}), diff_R(O_1^{(m-1)}, O_1^{(m)})$
Output: $AL_R(O_1^{(m)}, O_2^{(n)})$
1 **begin**
2 $AL_R(O_1^{(m)}, O_2^{(n)}) := AL_R(O_1^{(m-1)}, O_2^{(n)})$;
3 $\widetilde{alt} = \{(r_1, r_2, \lambda_R(r_1, r_2), \tilde{r}) | (r_1, r_2, \lambda_R(r_1, r_2), \tilde{r}) \in AL_R(O_1^{(m-1)}, O_2^{(n)}) \wedge r \in alt_{R^C}(R^{C_1(m-1)}, R^{C_1(m)})\}$
4 $AL_R(O_1^{(m)}, O_2^{(n)}) := AL_R(O_1^{(m)}, O_2^{(n)}) \setminus \{(r_1, r_2, \lambda_R(r_1, r_2), \tilde{r}) | (r_1, r_2, \lambda_R(r_1, r_2), \tilde{r}) \in \widetilde{alt} \wedge \lambda_R(r_1, r_2) < T_R\}$;
5 $\widetilde{alt^+} := \{r | r \in alt_{R^C}(R^{C_1(m-1)}, R^{C_1(m)}) \wedge \neg\exists (r, r_2, \lambda_R(r, r_2), \tilde{r}) \in AL_R(O_1^{(m-1)}, O_2^{(n)})\}$
6 $\widetilde{new} = \widetilde{alt^+} \cup new_{R^C}(R^{C_1(m-1)}, R^{C_1(m)})$
7 **for** $r \in \widetilde{new}$ **do**
8 **for** $r' \in R^{C_2(n)}$ **do**
9 **if** $\lambda_R(r, r') \geq T_R$ **then**
10 $AL_R(O_1^{(m)}, O_2^{(n)}) := AL_R(O_1^{(m)}, O_2^{(n)}) \cup \{(r, r', \lambda_R(r, r'), \tilde{r})\}$
11 **end**
12 **end**
13 **if** $\exists r_2 \in R^{C_1(m)} : ((r_2 \leftarrow r) \vee (r \leftarrow r_2)) \wedge (r \neq r_2)$ **then**
14 **if** $\exists (r, r', \lambda_R(r, r'), \tilde{r}) \in AL_R(O_1^{(m-1)}, O_2^{(n)})$ **then**
15 $AL_R(O_1^{(m)}, O_2^{(n)}) := AL_R(O_1^{(m)}, O_2^{(n)}) \cup \{(r_2, r', \lambda_R(r_2, r'), \tilde{r})\}$
16 **end**
17 **end**
18 **end**
19 **return** $AL_R(O_1^{(m)}, O_2^{(n)})$;
20 **end**

5 EXPERIMENTAL VERIFICATION

Our procedure for updating ontology alignment on the relation level has been experimentally verified. For this task benchmark datasets provided by Ontology Alignment Evaluation Initiative (OAEI) have been used. From all available datasets "the Conference Track" has been chosen which describes the domain of organizing conferences.

Updating alignment on the relation level has been compared with mapping determined by using LogMap (Jiménez-Ruiz and Grau, 2011). LogMap is an ontology alignment and alignment repair system which earned high positions in subsequent OAEI campaigns. However, it is worthy of notice, that the accuracy of current ontology mapping systems on property (in our work called as the relation) alignment is not very high. As it was mentioned before, in (Cheatham et al., 2018) authors show that the average precision and recall measures for properties alignment equal 0.45 and 0.17, respectively. For LogMap these measures are correspondingly higher: 0.62 and 0.28. However, the presented results demonstrate that the mappings tool on the relation level are not efficient and they are required to be improved.

In the first part of our experiment, we have chosen from the datasets a source ontology (called *CMT*) and a target ontology (called *MyReview*). The source ontology has been modified randomly. For the comparison of both tested methods, we have used an *accuracy* measure. It is calculated as the number of common links (relations' mappings) between two ontologies divided by the number of all connections found by both methods:

$$\left(\frac{|Align_{LogMap}(O_1^{(m+1)}, O_2^{(n)}) \cap Align(O_1^{(m+1)}, O_2^{(n)})|}{|Align_{LogMap}(O_1^{(m+1)}, O_2^{(n)}) \cup Align(O_1^{(m+1)}, O_2^{(n)})|}\right. \quad (7)$$

where $Align_{LogMap}$ is a set of mappings created by LogMap, and $Align$ is a set of mappings determined by our updating procedure

Our approach described in Section 4.2 distinguishes three ontology evolution on the relational level which can influence existing mappings timeliness: adding new relations, removing existing relations and modification of existing relations. By modification of relations, we understand a process of adding or removing concepts connected by such relations or a process of changing the semantics of those relations. In our work, we have focused only on how adding new relations in a base ontology influence changes of alignments. The removing and modification of existing relations and their impact on updating alignment have been omitted, because the LogMap tool determines the mapping from the beginning. If any relation does not exist anymore it is obvious that any mapping does not appear. Thus, the efficiency of updating alignment in case of removing or modification of relations doing by the LogMap tool and our approach is perfect.

In the case of adding new relations, a divergence between the results of the mentioned approach is bigger. In the first part of our experiment, we verify how changes in base ontologies influence changes of alignments. At the beginning only a 10% new relations have been added and in the end, the number of relations in the source ontology has been doubled. The results clearly showed that the number of introduced modifications didn't have any significant impact on the value of accuracy - for each iteration we obtained a value between 75% and 83%.

The conducted experiment allows us to conclude that our approach and LogMap give similar alignments (in terms of the accuracy measure defined in Equation 7) of two ontologies. However, our approach found more correct connections between relations. As it was mentioned before, the efficiency of LogMap on the relational level is not sufficient.

Thus, the smaller number of mappings generated by LogMap in comparison with our solution confirmed the correctness of our assumptions and the results of the experiment and demonstrated the efficiency of our approach. Furthermore, updating an existing alignment is less expensive than building new mappings from the beginning which combined with the low computational complexity of our algorithm makes it a more attractive solution than existing so far.

In the second part of our experiment, we have not changed a target ontology. However, we have applied the same amount of modifications in different ontologies from the OAEI dataset which has been treated as a source ontology. Thus, we would like to verify how the same changes influence on updating alignment. The obtained results are presented in Table 1.

As in previous research, our approach found more correct links between relations than the LogMap tool. The accuracy of results obtained by both methods is usually very high (around 80 %) which proves the utility and correctness of our approach.

6 FUTURE WORKS AND SUMMARY

The paper is devoted to the problem of updating ontology alignment in the case of ontology evolution. This is still an open problem and not well investigated by other researches. So far, it is possible to find some tools dedicated to the mentioned problem. However, many solutions focus only on concept or instances level, and push the relation level to the background, treating it as less important.

In this paper, we defined the function which reflects the degree of significance to which relations within a given ontology have been changed in time. This function served as the criterion for deciding whether or not the designated alignment (between the tracked and some other ontology) needs updating. The main result of our work is an algorithm that revalidated the existing alignment by removing stale and adding new mappings.

Our method has been experimentally verified in comparison with the well-known tool called LogMap. The mappings updated by our method have been juxtaposed with mappings designated from the beginning by LogMap. The experiments show us that our approach returned a similar result like LogMap. However, our procedure found more correct link between relation than LogMap. It proves the correctness and efficiency of our approach.

In the nearest future work, we want to elaborate on the ontology alignment updating procedure for the instance level. If all levels of an ontology will be covered by proper alignment revalidating procedures then we will plan to implement and prepared the complex tool for ontology storing, tracking changes, knowledge integration, mappings determination and actualization in case of ontology evolution.

ACKNOWLEDGEMENTS

This research project was supported by grant No. 2017/26/D/ST6/00251 from the National Science Centre, Poland.

Table 1: Different Source Ontologies.

Source ontology	Number of relations in the source ontology	Number of maps found by the proposed approach	Number of maps found by LogMap	Accuracy
CMT	49	7	5	0.714
Cocus	35	6	4	0.667
Confious	52	7	6	0.857
ConfTool	13	5	4	0.8
Crs	15	9	8	0.89
Edas	30	8	8	1
Ekaw	33	5	4	0.8
Iasted	38	5	4	0.8
Linklings	31	5	4	0.8
Micro	17	7	7	1
OpenConf	24	9	8	0.89
PCS	24	11	5	0.455
Sigkdd	17	6	5	0.833
Sofsem	64	9	8	0.889

REFERENCES

Abbes, H. and Gargouri, F. (2017). Managing modular ontology evolution under big data integration. In *European, Mediterranean, and Middle Eastern Conference on Information Systems*, pages 17–28. Springer.

Algergawy, A. et al. (2018). Results of the ontology alignment evaluation initiative 2018. In *13th International Workshop on Ontology Matching co-located with the 17th ISWC (OM 2018)*, pages 76–116.

Cheatham, M., Pesquita, C., Oliveira, D., and McCurdy, H. B. (2018). The properties of property alignment on the semantic web. *IJMSO*, 13(1):42–56.

Da Silva Jomar, nd Baiao, F. A. and Kate, R. (2018). Alin results for oaei 2018. In *13th International Workshop on Ontology Matching co-located with the 17th ISWC (OM 2018)*, pages 117–124.

Dos Reis, J. C. (2018). Mapping refinement based on ontology evolution: A context-matching approach. In *ONTOBRAS*, pages 251–256.

Dos Reis, J. C., Dinh, D., Pruski, C., Da Silveira, M., and Reynaud-Delaître, C. (2013). Mapping adaptation actions for the automatic reconciliation of dynamic ontologies. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 599–608. ACM.

Dos Reis, J. C. and Yamamoto, V. E. (2019). Updating ontology alignments based on new concepts and their context. In *Revista dos Trabalhos de Iniciação Científica da UNICAMP, Campinas, SP*, volume 27.

Faria, D., Pesquita, C., Balasubramani, B. S., Tervo, T., Carriço, D., Garrilha, R., Couto, F. M., and Cruz, I. F. (2018). Results of aml participation in oaei 2018. In *13th International Workshop on Ontology Matching co-located with the 17th ISWC (OM 2018)*, pages 125–131.

Hartung, M., Groß, A., and Rahm, E. (2013). Conto–diff: generation of complex evolution mappings for life science ontologies. *Journal of biomedical informatics*, 46(1):15–32.

Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P. F., Rudolph, S., et al. (2009). Owl 2 web ontology language primer. *W3C recommendation*, 27(1):123.

Jiménez-Ruiz, E. and Grau, B. C. (2011). Logmap: Logic-based and scalable ontology matching. In *International Semantic Web Conference*, pages 273–288. Springer.

Jiménez-Ruiz, E., Grau, B. C., and Cross, V. (2018). Logmap family participation in the oaei 2018. In *13th International Workshop on Ontology Matching co-located with the 17th ISWC (OM 2018)*, pages 173–178.

Kachroudi, M., Diallo, G., and Yahia, S. B. (2018). Kepler at oaei 2018. In *13th International Workshop on Ontology Matching co-located with the 17th ISWC (OM 2018)*, pages 173–178.

Khattak, A. M., Pervez, Z., Khan, W. A., Khan, A. M., Latif, K., and Lee, S. (2015). Mapping evolution of dynamic web ontologies. *Information Sciences*, 303:101–119.

Kiourtis, A., Mavrogiorgou, A., Menychtas, A., Maglogiannis, I., and Kyriazis, D. (2019). Structurally mapping healthcare data to hl7 fhir through ontology alignment. *Journal of medical systems*, 43(3):62.

Kozierkiewicz, A. and Pietranik, M. (2019). A formal framework for the ontology evolution. In *Asian Conference on Intelligent Information and Database Systems*, pages 16–27. Springer.

Mohammadi, M., Hofman, W., and Tan, Y.-H. (2018). Sanom results for oaei 2018. In *13th International Workshop on Ontology Matching co-located with the 17th ISWC (OM 2018)*, pages 125–131.

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

Pietranik, M. and Nguyen, N. T. (2014). A multi-attribute based framework for ontology aligning. *Neurocomputing*, 146:276–290.

Thenmozhi, M. and Vivekanandan, K. (2012). A semi-automatic approach to update mapping for ontology evolution. In *Int. Conf. on Computational Intelligence and Information Technology, CIIT*, pages 319–324.

Wang, T., Gu, H., Wu, Z., and Gao, J. (2020). Multi-source knowledge integration based on machine learning algorithms for domain ontology. *Neural Computing and Applications*, 32(1):235–245.