Completing the is-a Structure of Biomedical Ontologies

Zlatan Dragisic[1,2], Patrick Lambrix[1,2], and Fang Wei-Kleiner[1]

[1] Department of Computer and Information Science, Linköping University, 581 83 Linköping, Sweden
[2] Swedish e-Science Research Centre, Sweden

Abstract. Ontologies in the biomedical domain are becoming a key element for data integration and search. The usefulness of the applications which use ontologies is often directly influenced by the quality of ontologies, as incorrect or incomplete ontologies might lead to wrong or incomplete results for the applications. Therefore, there is an increasing need for repairing defects in ontologies. In this paper we focus on completing ontologies. We provide an algorithm for completing the is-a structure in $\mathcal{EL}$ ontologies which covers many biomedical ontologies. Further, we present an implemented system based on the algorithm as well as an evaluation using three biomedical ontologies.

1 Introduction

With the increasing presence of biomedical data sources on the Internet more and more research effort is put into finding possible ways for integrating and searching such often heterogeneous sources. Semantic Web technologies such as ontologies, are becoming a key technology in this effort. Ontologies provide a means for modelling the domain of interest and they allow for information reuse, portability and sharing across multiple platforms. Efforts such as the Open Biological and Biomedical Ontologies (OBO) Foundry, BioPortal and Unified Medical Language System (UMLS) aim at providing repositories for biomedical ontologies and relations between these ontologies thus providing means for annotating and sharing biomedical data sources. Many of the ontologies in the biomedical domain can be represented using the $\mathcal{EL}$ description logic or small extensions thereof (e.g. [1] and the TONES Ontology Repository).

Developing ontologies is not an easy task, and often the resulting ontologies (including their is-a structures) are not complete. In addition to being problematic for the correct modelling of a domain, such incomplete ontologies also influence the quality of semantically-enabled applications. Incomplete ontologies when used in semantically-enabled applications can lead to valid conclusions being missed.

In ontology-based search, queries are refined and expanded by moving up and down the hierarchy of concepts. Incomplete structure in ontologies influences the quality of the search results. As an example, suppose we want to find articles in the MeSH Database of PubMed using the term *Scleral Diseases* in MeSH. By default the query will follow the hierarchy of MeSH and include more specific terms for searching, such as *Scleritis*. If the relation between *Scleral Diseases* and *Scleritis* is missing in MeSH, we will miss 922 articles in the search result, which is about 57% of the original result[1]. The structural information is also important information in ontology engineering

[1] PubMed accessed on 21-02-2014.

H. Galhardas and E. Rahm (Eds.): DILS 2014, LNBI 8574, pp. 66–80, 2014.

research. For instance, most current ontology alignment systems use structure-based strategies to find mappings between the terms in different ontologies (e.g. overview in [27]) and the modeling defects in the structure of the ontologies have an important influence on the quality of the ontology alignment results.

In this paper we tackle the problem of completing the is-a structure of ontologies. Completing the is-a structure requires adding new correct is-a relations to the ontology. We identify two cases for finding relations which need to be added to an ontology. In **case 1** missing is-a relations have been detected and the task is to find ways of making these detected is-a relations derivable in the ontology. There are many approaches to detect missing is-a relations, e.g., using linguistic or logical patterns or by using knowledge intrinsic to an ontology network (see Section 6). However, in general, these approaches do not detect *all* missing is-a relations and in several cases even only few. Therefore, we assume that we have obtained a set of missing is-a relations for a given ontology (but not necessarily all). In the case where our set of missing is-a relations contains *all* missing is-a relations, completing the ontology is easy. We just add all missing is-a relations to the ontology and a reasoner can compute all logical consequences. However, when the set of missing is-a relations does not contain all missing is-a relations - and this is the common case - there are different ways to complete the ontology. The easiest way is still to just add the missing is-a relations to the ontology. For instance, T in Figure 1 represents a small ontology inspired by Galen ontology (http://www.co-ode.org/galen/), that is relevant for our discussions. Assume that we have detected that Endocarditis $\sqsubseteq$ PathologicalPhenomenon and GranulomaProcess $\sqsubseteq$ NonNormalProcess are missing is-a relations (M in Figure 1). Obviously, adding these relations to the ontology will repair the missing is-a structure. However, there are other more interesting possibilities. For instance, adding Carditis $\sqsubseteq$ CardioVascularDisease and GranulomaProcess $\sqsubseteq$ PathologicalProcess also repairs the missing is-a structure. Further, these is-a relations are correct according to the domain and constitute new is-a relations (e.g. Carditis $\sqsubseteq$ CardioVascularDisease) that were not derivable from the ontology and not originally detected by the detection algorithm.[2] We also note that from a logical point of view, adding Carditis $\sqsubseteq$ Fracture and GranulomaProcess $\sqsubseteq$ NonNormalProcess also repairs the missing is-a structure. However, from the point of view of the domain, this solution is not correct. Therefore, as it is the case for all approaches for dealing with modeling defects, a domain expert needs to validate the logical solutions.

In **case 2** no missing is-a relations are given. In this case we investigate existing is-a relations in the ontology and try to find new ways of deriving these existing is-a relations. This might pinpoint to the necessity of adding new missing is-a relations to the ontology. As an example, let us assume that our ontology contains relations $T \cup M$ in Figure 1. If we assume now that we want to investigate new ways of deriving relations in M then obviously adding Carditis $\sqsubseteq$ CardioVascularDisease and GranulomaProcess $\sqsubseteq$ PathologicalProcess would be one possibility given that both are correct according to the domain.

The basic problem underlying the two cases can be formalized in the same way (Section 2.2).

[2] Therefore, the approach in this paper can also be seen as a detection method that takes already found missing is-a relations as input.

C = { GranulomaProcess, CardioVascularDisease, PathologicalPhenomenon, Fracture, Endocarditis, Carditis, InflammationProcess, PathologicalProcess, NonNormalProcess}

T = { CardioVascularDisease $\sqsubseteq$ PathologicalPhenomenon, Fracture $\sqsubseteq$ PathologicalPhenomenon, $\exists$hasAssociatedProcess.PathologicalProcess $\sqsubseteq$ PathologicalPhenomenon, Endocarditis $\sqsubseteq$ Carditis, Endocarditis $\sqsubseteq$ $\exists$hasAssociatedProcess.InflammationProcess, PathologicalProcess $\sqsubseteq$ NonNormalProcess }

M = { Endocarditis $\sqsubseteq$ PathologicalPhenomenon, GranulomaProcess $\sqsubseteq$ NonNormalProcess }

The following is-a relations are correct according to the domain, i.e., Or returns $true$ for:
GranulomaProcess $\sqsubseteq$ InflammationProcess, GranulomaProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ NonNormalProcess, CardioVascularDisease $\sqsubseteq$ PathologicalPhenomenon, Fracture $\sqsubseteq$ PathologicalPhenomenon, Endocarditis $\sqsubseteq$ PathologicalPhenomenon, Endocarditis $\sqsubseteq$ Carditis, Endocarditis $\sqsubseteq$ CardioVascularDisease, Carditis $\sqsubseteq$ PathologicalPhenomenon, Carditis $\sqsubseteq$ CardioVascularDisease, InflammationProcess $\sqsubseteq$ PathologicalProcess, InflammationProcess $\sqsubseteq$ NonNormalProcess, PathologicalProcess $\sqsubseteq$ NonNormalProcess.

Let $\mathcal{P}$ = GTAP(T, C, Or, M).

Fig. 1. Small example

The contributions of this paper are the following. We present an approach for completing the is-a structure of $\mathcal{EL}$ ontologies which aims at introducing new information to the ontology (Section 3). Together with the algorithm for completing the is-a structure we present an implemented system (Section 4). Next, we provide an evaluation of the system using three ontologies from the biomedical domain and discuss lessons learned. The paper concludes with the discussion of related work and possible future work (Sections 6 and 7). We continue with some necessary preliminaries in Section 2.

2 Preliminaries

2.1 The Description Logic $\mathcal{EL}$

Concept descriptions are constructed inductively from a set N_C of atomic concepts and a set N_R of atomic roles. The concept constructors are the top concept $\top$, conjunction, and existential restriction. The syntax of the different constructors can be found in Figure 2. An interpretation $\mathcal{I}$ consists of a non-empty set $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ which assigns to each atomic concept $A \in N_C$ a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, to each atomic role $r \in N_R$ a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is straightforwardly extended to complex concepts. An $\mathcal{EL}$ TBox[3] is a finite set of *general concept inclusions* (GCIs), whose syntax can be found in the lower part of Figure 2. An interpretation $\mathcal{I}$ is a *model* of a TBox T if for each GCI in T, the conditions given in the third column of Figure 2 are satisfied.

The main reasoning task for description logics is subsumption in which the problem is to decide for a TBox T and concepts C and D whether $T \models C \sqsubseteq D$. Subsumption in $\mathcal{EL}$ is polynomial.

[3] Named CBox in [1].

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

Fig. 2. $\mathcal{EL}$ Syntax and Semantics

2.2 Completing is-a Structure

The problem of completing the missing is-a structure in an ontology can be formalized as a generalized version of the TBox abduction problem [28].

We assume that our ontology is represented using a TBox T in $\mathcal{EL}$. Further, we have a set of missing is-a relations which are represented by a set M of atomic concept subsumptions. In *case 1* in the introduction, these missing is-a relations were detected. In *case 2* the elements in M are existing is-a relations in the ontology that are temporarily removed, and T represents the ontology that is obtained by removing the elements in M from the original ontology. (They can later be added again after completing the ontology.) To complete the is-a structure of an ontology, the ontology should be extended with a set S of atomic concept subsumptions (repair) such that the extended ontology entails the missing is-a relations. However, the added atomic concept subsumptions should be correct according to the domain. In general, the set of all atomic concept subsumptions that are correct according to the domain are not known beforehand. Indeed, if this set were given then we would only have to add this to the ontology. The common case, however, is that we do not have this set, but instead can rely on a domain expert that can decide whether an atomic concept subsumption is correct according to the domain. In our formalization the domain expert is represented by an oracle Or that when given an atomic concept subsumption, returns true or false. It is then required that for every atomic concept subsumption $s \in S$, we have that $Or(s) = true$. The following definition formalizes this.

Definition 1 (Generalized TBox Abduction). *(variant of [28])*
Let T be a TBox in $\mathcal{EL}$ and C be the set of all atomic concepts in T.
Let $M = \{A_i \sqsubseteq B_i \mid A_i, B_i \in C\}$ be a finite set of TBox assertions.
Let $Or : \{C_i \sqsubseteq D_i \mid C_i, D_i \in C\} \rightarrow \{true, false\}$.
A solution to the generalized TBox abduction problem (GTAP) (T, C, Or, M) is any finite set $S = \{E_i \sqsubseteq F_i \mid E_i, F_i \in C \wedge Or(E_i \sqsubseteq F_i) = true\}$ of TBox assertions, such that $T \cup S$ is consistent and $T \cup S \models M$.

We note that an additional condition could be enforced in the definition i.e. $\forall m \in M : Or(m) = true$. Regarding this condition, if some missing is-a relation is not correct according to the domain, it could still be possible to find a solution. However, in this case the domain expert makes mistakes in the judgement or T is not correct according to the domain. In practice, it is therefore advantageous to validate whether the missing is-a relations are correct according to the domain before repairing.

As an example, let us consider GTAP $\mathcal{P}$ as defined in Figure 1. Then a possible solution for $\mathcal{P}$ is $\{$Carditis $\sqsubseteq$ CardioVascularDisease, InflammationProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ InflammationProcess$\}$. Another possible solution is $\{$Carditis $\sqsubseteq$ CardioVascularDisease, GranulomaProcess $\sqsubseteq$ PathologicalProcess$\}$ as explained in Section 1.

There can be many solutions for a GTAP and, as explained in Section 1, not all solutions are equally interesting. Therefore, in [28] we proposed two preference criteria on the solutions. The first criterion is a criterion that is not used in other abduction problems, but that is particularly important for GTAP. In GTAP it is important to find solutions that add to the ontology as much information as possible that is correct according to the domain. Therefore, the first criterion prefers solutions that imply more information.

Definition 2 (More Informative). *Let S and S' be two solutions to the GTAP (T, C, Or, M). S is said to be* more informative *than S' iff $T \cup S \models S'$ and $T \cup S' \not\models S$.*

Further, we say that S is equally informative *as S' iff $T \cup S \models S'$ and $T \cup S' \models S$.*

Consider two solutions[4] to $\mathcal{P}$, S_1 = $\{$InflammationProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ InflammationProcess$\}$ and S_2 = $\{$InflammationProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ PathologicalProcess$\}$. In this case solution S_1 is more informative than S_2.

The second criterion is a classical criterion in abduction problems. It requires that no element in a solution is redundant.

Definition 3 (Subset Minimality). *A solution S to the GTAP (T, C, Or, M) is said to be subset minimal iff there is no proper subset $S' \subsetneq S$ such that S' is a solution.*

An example of a subset minimal solution for $\mathcal{P}$ is $\{$InflammationProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ InflammationProcess$\}$. On the other hand, solution $\{$Carditis $\sqsubseteq$ CardioVascularDisease, InflammationProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ InflammationProcess$\}$ is not subset minimal as it contains Carditis $\sqsubseteq$ CardioVascularDisease which is redundant for repairing the missing is-a relations.

Three different combinations of these criteria were identified and formalized in [28]. Solutions with higher level of informativeness and no redundancy are preferred and this is formalized by skyline optimality.

Definition 4 (Skyline Optimal). *A solution S to the GTAP (T, C, Or, M) is said to be skyline optimal iff there does not exist another solution S' such that S' is a proper subset of S and S' is equally informative as S.*

[4] Observe that both missing is-relations are derivable using S_1. GranulomaProcess $\sqsubseteq$ NonNormalProcess is derivable as GranulomaProcess $\sqsubseteq$ InflammationProcess (S_1), InflammationProcess $\sqsubseteq$ PathologicalProcess (S_1), and PathologicalProcess $\sqsubseteq$ NonNormalProcess (T). Endocarditis $\sqsubseteq$ PathologicalPhenomenon is derivable as Endocarditis $\sqsubseteq$ $\exists$hasAssociatedProcess.InflammationProcess (T), $\exists$hasAssociatedProcess.InflammationProcess $\sqsubseteq$ $\exists$hasAssociatedProcess.PathologicalProcess (S_1), and $\exists$hasAssociatedProcess.PathologicalProcess $\sqsubseteq$ PathologicalPhenomenon (T). Similarly for S_2.

For example, {InflammationProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ InflammationProcess, Carditis $\sqsubseteq$ CardioVascularDisease} is a skyline optimal solution for $\mathcal{P}$.

3 Algorithm

In this section we present an algorithm for completing the is-a structure (solving GTAP (T, C, Or, M)) in ontologies that are represented in $\mathcal{EL}$ and where the TBox is normalized as described in [1]. A normalized TBox T contains only axioms of the forms $A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B$, $A \sqsubseteq \exists r.B$, and $\exists r.A \sqsubseteq B$, where A, A_1, …, A_n and B are atomic concepts and r is a role. Further, based on lessons learned in [28], we require that the missing is-a relations are validated before the repairing and thus $\forall m \in M : Or(m) = true$. This, together with the fact that $\mathcal{EL}$ TBoxes are always consistent, gives us that M is a solution.

In general, we would like to find a solution for GTAP at the highest level of informativeness. However, this can only be *guaranteed* if we know *all* missing is-a relations. One way to obtain this is using a brute-force method and ask Or for every pair in $C \times C$ whether it is a correct is-a relation according to the domain or not. In practice, for large ontologies this is not feasible. Therefore, the algorithm in Algorithm 1 computes initially a skyline optimal solution for GTAP (T, C, Or, M) and iteratively tries to find other skyline optimal solutions at higher levels of informativeness. As M is a solution, the algorithm will always return a result. The result can be a subset minimal solution that is a subset of M or a solution that is more informative than M.

The basic step in the algorithm (*RepairSingleIsa*) computes a solution for a GTAP with one missing is-a relation (i.e. GTAP $(T, C, Or, \{E \sqsubseteq F\})$ in the following way. First, superconcepts of E are collected in a *Source* set and subconcepts of F are collected in a *Target* set (lines 3 and 4). *Source* contains expressions of the forms A and $\exists r.A$ while *Target* contains expressions of the forms A, $A_1 \sqcap \ldots \sqcap A_n$ and $\exists r.A$ where A, A_1, …, A_n are atomic concepts and r is a role. Adding an is-a relation between an element in Source and an element in Target to the ontology would make $E \sqsubseteq F$ derivable (and thus this gives us logical solutions, but not necessarily solutions that are correct according to the domain). As we are interested in solutions containing is-a relations between atomic concepts, we check for every pair (A,B) $\in$ Source $\times$ Target whether A and B are atomic concepts and $Or(A \sqsubseteq B) = true$ (i.e. correct according to the domain). If so, then this is a possible solution for GTAP $(T, C, Or, \{E \sqsubseteq F\})$. However, if the current solution already contains is-a relations that would lead to the entailment of $A \sqsubseteq B$ then we do not use $A \sqsubseteq B$ (8-9). Otherwise we use $A \sqsubseteq B$ and remove elements from the current solution that would be entailed if $A \sqsubseteq B$ is used (10-12). Further, in the case where A is of the form $\exists r.N$ and B is of the form $\exists r.O$, then making $N \sqsubseteq O$ derivable would also make $A \sqsubseteq B$ derivable (13-14). It is clear that for the result of *RepairSingleIsa*, i.e. Sol, the following holds: $T \cup Sol \models E \sqsubseteq F$ and $\forall s \in Sol : Or(s) = true$. Together with the fact that $\mathcal{EL}$ TBoxes are consistent, this leads to the fact that Sol is a solution of GTAP $(T, C, Or, \{E \sqsubseteq F\})$.

In *RepairMultipleIsa* the algorithm collects for each missing is-a relation a solution from *RepairSingleIsa* and takes the union of these. Therefore, the following holds for

```
1  Procedure RepairSingleIsa begin
       Input: E ⊑ F, T, Or, C
       Output: Solution for GTAP (T, C, Or, {E ⊑ F})
2      Sol := ∅;
3      Source := find superconcepts of E;
4      Target := find subconcepts of F;
5      foreach A ∈ Source do
6          foreach B ∈ Target do
7              if A and B are atomic concepts & A ⊑ B ∈ Or then
8                  if there exists K ⊑ L ∈ Sol such that T ⊨ A ⊑ K and T ⊨ L ⊑ B then
9                      do nothing;
10                 else
11                     remove every K ⊑ L ∈ Sol s.t. T ⊨ K ⊑ A and T ⊨ B ⊑ L;
12                     Sol := Sol ∪ {A ⊑ B};
13             else if A is of the form ∃r.N & B is of the form ∃r.O then
14                 Sol := Sol ∪ RepairSingleIsa(N ⊑ O, T, Or, C);
15     return Sol;
16 Procedure RepairMultipleIsa begin
       Input: M, T, Or, C
       Output: Solution for GTAP (T, C, Or, M)
17     foreach E_i ⊑ F_i ∈ M do
18         SingleSol_i := RepairSingleIsa(E_i ⊑ F_i, T, Or, C);
19     Solution := ⋃_i SingleSol_i;
20     remove redundancy in Solution within same level of informativeness;
21     return Solution;
22 Procedure Repair begin
       Input: M, T, Or, C
       Output: Solution for GTAP (T, C, Or, M)
23     Missing := M;
24     Solution := RepairMultipleIsa(Missing, T, Or, C);
25     Final-Solution := Solution;
26     while Solution ≠ Missing do
27         Missing := Solution;
28         Solution := RepairMultipleIsa(Missing, T ∪ Missing, Or, C);
29         Final-Solution := Final-Solution ∪ Solution;
30         remove redundancy in Final-Solution within same level of informativeness;
31     return Final-Solution;
```

Algorithm 1. Solving GTAP

Solution in line 19: $T \cup Solution \models M$ and $\forall s \in Solution : Or(s) = true$. Together with the fact that $\mathcal{EL}$ TBoxes are consistent, this leads to the fact that Solution is a solution of GTAP (T, C, Or, M). Further, in line 20, we remove redundancy while keeping the same level of informativeness, and thus obtain a skyline optimal solution. (In the case where there are several ways to remove redundancy, one is chosen, as the extended ontologies will be equivalent in the sense that they entail the same statements.)

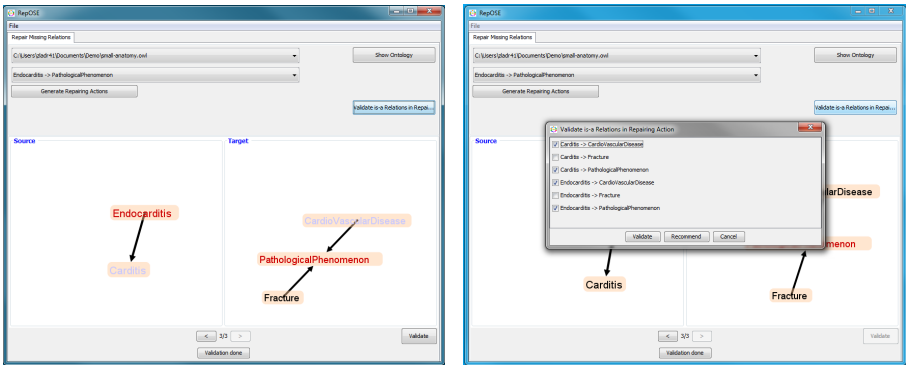
In *Repair* we try to improve the result from *RepairMultipleIsa* by trying to find a skyline optimal solution at a higher level of informativeness. Given that any element in the solution of *RepairMultipleIsa* that is not in M can be considered as a new missing is-a relation (which was not detected earlier), we can try to find additional more informative ways of repairing by solving a new GTAP problem for these new missing is-a relations (and continue as long as new missing is-a relations are detected). As a (skyline optimal) solution for the new GTAP is also a (skyline optimal) solution of the original GTAP, the solution found in *Repair* is a skyline optimal solution for the original GTAP.

As an example run consider the GTAP in Figure 1. For a given ontology and set of missing is-a relations, the algorithm will first find solutions for repairing individual missing is-a relations using *RepairSingleIsA*. For the missing is-

a relation Endocarditis $\sqsubseteq$ PathologicalPhenomenon the following is-a relations provide logical solutions for repairing the missing is-a relation: Endocarditis $\sqsubseteq$ PathologicalPhenomenon, Endocarditis $\sqsubseteq$ Fracture, Endocarditis $\sqsubseteq$ CardioVascularDisease, Carditis $\sqsubseteq$ PathologicalPhenomenon, Carditis $\sqsubseteq$ Fracture, Carditis $\sqsubseteq$ CardioVascularDisease as well as InflammationProcess $\sqsubseteq$ PathologicalProcess. As the first one is the missing is-a relation which was already validated, only the other six is-a relations are presented to the oracle for validation. Out of these six Endocarditis $\sqsubseteq$ Fracture and Carditis $\sqsubseteq$ Fracture are not correct according to the domain and are therefore not included in solutions. Further, relations Endocarditis $\sqsubseteq$ CardioVascularDisease, Endocarditis $\sqsubseteq$ PathologicalPhenomenon, Carditis $\sqsubseteq$ PathologicalPhenomenon are removed given it is possible to entail them from the ontology together with the remaining relations. Therefore, after validation, *RepairSingleIsA* returns $\{$InflammationProcess $\sqsubseteq$ PathologicalProcess, Carditis $\sqsubseteq$ CardioVascularDisease$\}$. The same process is repeated for the second missing is-a relation GranulomaProcess $\sqsubseteq$ NonNormalProcess. In this case the following is-a relations provide logical solutions for repairing the missing is-a relation: GranulomaProcess $\sqsubseteq$ NonNormalProcess and GranulomaProcess $\sqsubseteq$ PathologicalProcess. GranulomaProcess $\sqsubseteq$ NonNormalProcess is the missing is-a relation and was already validated as correct according to the domain. GranulomaProcess $\sqsubseteq$ PathologicalProcess is presented to the oracle and validated as correct according to the domain. As GranulomaProcess $\sqsubseteq$ NonNormalProcess can be entailed from the ontology together with GranulomaProcess $\sqsubseteq$ PathologicalProcess, *RepairSingleIsA* returns $\{$GranulomaProcess $\sqsubseteq$ PathologicalProcess$\}$. The solutions for the single is-a relations are then combined to form a solution for the set of missing is-a relations. In our case, there are no redundant relations and therefore *RepairMultipleIsA* returns $\{$InflammationProcess $\sqsubseteq$ PathologicalProcess, Carditis $\sqsubseteq$ CardioVascularDisease, GranulomaProcess $\sqsubseteq$ PathologicalProcess$\}$. We note that this is a skyline optimal solution. In *Repair* the system tries to improve the acquired solution. This time the oracle is presented with a total of 13 relations for validation out of which only one is validated to be correct, i.e. GranulomaProcess $\sqsubseteq$ InflammationProcess. This is added to the solution. Given this new is-a relation, GranulomaProcess $\sqsubseteq$ PathologicalProces is removed from the solution as it can now be entailed from the ontology and GranulomaProcess $\sqsubseteq$ InflammationProcess. The new solution is $\{$InflammationProcess $\sqsubseteq$ PathologicalProcess, Carditis $\sqsubseteq$ CardioVascularDisease, GranulomaProcess $\sqsubseteq$ InflammationProcess$\}$. This is again a skyline optimal solution and it is more informative than the previous solution. As new missing is-a relations were detected, the repairing is run for the third time. However, in this run the solution is not improved and thus the algorithm outputs the final result. We note that in this example we found a skyline optimal solution that is also solution with the highest level of informativeness. In general, however, it is not possible to know whether the solution is of the highest level of informativeness without checking every possible is-a relation between atomic concepts in the ontology.

4 System

We have implemented a system for completing the missing is-a structure in $\mathcal{EL}$ ontologies based on the algorithm in Algorithm 1. The input to the system is a an ontology and a set of validated missing is-a relations. The output is a solution to GTAP (called



(a) Repairing using Source and Target sets. (b) Validating is-a relations in a repairing action.

Fig. 3. System screenshots

a *repairing action*). The system was implemented in Java and uses the ELK reasoner (version 0.4.1) [21] to detect implicit entailments in the ontology. The system is semi-automatic and requires interaction with a user which is a domain expert serving as an oracle and who decides whether an is-a relation is correct according to the domain.

Once the ontology and the set of missing is-a relations are loaded, the user starts the debugging process by pressing the button `Generate Repairing Actions`. The system then removes redundant is-a relations and the non-redundant missing is-a relations are shown in a drop-down list allowing the user to switch between missing is-a relations. Additional relations acquired from lines 13 and 14 in the algorithm (Algorithm 1) are also included in the drop-down list. It is also possible to scroll between relations using the arrow buttons in the bottom part of the screen.

After selecting an is-a relation from the list, the user is presented with the Source and the Target set for that is-a relation. The user then needs to choose relations which are correct according to the domain for that is-a relation. Missing is-a relations are automatically validated to be correct according to the domain while the relations that were acquired from lines 13 and 14 in the algorithm have to be explicitly validated by the user.

In Figure 3(a) the user is presented with the Source and the Target set for the missing is-a relation Endocarditis $\sqsubseteq$ PathologicalPhenomenon (concepts in the missing is-a relation are marked in red). In this case the user has selected $\{$Carditis $\sqsubseteq$ CardioVascularDisease$\}$ as a repairing action for the missing is-a relation (concepts marked in purple) and needs to confirm this by clicking the `Validate` button.

The user also has the option to check which relations have been validated so far and which relations can be validated, by clicking the `Validate Is-a Relations` button. In the pop-up window that appears the user can validate new relations, remove validations from already validated relations as well as ask for a recommendation by clicking the `Recommend` button (Figure 3(b)). Recommendations are acquired by querying external sources (currently, WordNet, UMLS Methathesaurus and Uberon).

The validation phase is ended by clicking on the `Validation Done` button. The system then calculates the consequences of the chosen repairing actions and presents the user with a new set of is-a relations that need to be repaired. The validation phase and consequent computations represent one iteration of the Repair procedure in Algorithm 1. If the repairing did not change between two iterations the system outputs the repairing.

At any point the user can save validated relations from the "File" menu which makes it possible to do debugging accross multiple sessions.

5 Experiments

We have run several experiments on an Intel Core i7-2620M Processor at 3.07 GHz with 4 GB RAM under Windows 7 Professional and Java 1.7 compiler. The experiments cover the two cases from the introduction. In all experiments the validation phase took the most time while the computations between iterations took less than 10 seconds.

The results are summarized in Figures 4 - 5. The 'It' columns represent the different iterations of Repair in Algorithm 1. The 'Missing' rows give the number of missing is-a relations in each iteration. Such a missing is-a relation can be repaired by adding itself ('Repaired by itself'), by adding other is-a relations that were not derivable in the ontology and thus represent new knowledge added to the ontology ('Repaired using new knowledge'). The 'New relations' row shows how many new is-a relations were added to the ontology. When such relations were found using $\exists$ (lines 13 and 14 in the algorithm), then the number of such relations is shown in parentheses. We note that for iteration $i + 1$ the number of missing is-a relations is the number of new relations from iteration i plus the number of missing is-a relations repaired by themselves from iteration i if there are no redundant relations. We also note that in the *last* iteration all missing is-a relations from that iteration are always repaired by themselves and these represent the final repairing action.

5.1 Case 1 Experiment – OAEI Anatomy

We debugged the two ontologies from the Anatomy track at the 2013 Ontology Alignment Evaluation Initiative, i.e. Mouse Anatomy ontology (AMA) containing 2744 concepts and a fragment of NCI human anatomy ontology (NCI-A) containing 3304 concepts. The input missing is-a relations for these two experiments were a set of 94 and 58 missing is-a relations, respectively, for AMA and NCI-A. These missing is-a relations were obtained by using a logic-based approach using an alignment between AMA and NCI-A [25] to generate candidate missing is-a relations which were then validated by a domain expert to obtain actual missing is-a relations. Therefore, this experiment is related to *case 1*.

Mouse Anatomy. The results for debugging AMA are given in Figure 4(a). Three iterations were required to reach the final solution. Out of 94 initial missing is-a relations 37 were repaired by repairing actions which add new knowledge to the ontology while 57 were repaired using only the missing is-a relation itself. There were no derivable

	It1	It2	It3
Missing	94	101	101
Repaired by itself	57	98	101
Repaired using new knowledge	37	3	0
New relations	44	3	0

(a) Results for debugging AMA - Mouse Anatomy ontology.

	It1	It2	It3
Missing	58	55	54
Repaired by itself	49	50	54
Repaired using new knowledge	9	5	0
New relations	6	4	0

(b) Results for debugging NCI-A - Human Anatomy ontology.

Fig. 4. OAEI experiments

relations. In total 44 new and non-redundant relations were added to the ontology in the first iteration. Out of 37 relations which were repaired by adding new relations, 22 had more than 1 non-redundant relation in the repairing action. For example, the missing is-a relation wrist joint $\sqsubseteq$ joint is repaired by a repairing action $\{$limb joint $\sqsubseteq$ joint, wrist joint $\sqsubseteq$ synovial joint$\}$.

The set of missing is-a relations in the second iteration contains 101 relations, i.e. 57 relations which were repaired by adding the missing is-a relation itself and 44 newly added relations. In this iteration, 3 is-a relations were repaired by adding new knowledge to the ontology. All 3 of these is-a relations are is-a relations which were added in the previous iteration. For example, is-a relation wrist joint $\sqsubseteq$ synovial joint is repaired by a repairing action $\{$wrist joint $\sqsubseteq$ hand joint$\}$ which is possible given that the is-a relation metacarpo-phalangeal joint $\sqsubseteq$ joint from the initial set of missing is-a relations was repaired by a repairing action $\{$hand joint $\sqsubseteq$ synovial joint, limb joint $\sqsubseteq$ joint$\}$ in the first iteration. Finally, the set of missing is-a relations containing 101 is-a relations in the third iteration is also the solution for the initial set of missing is-a relations given that no new relations were added in the third iteration.

NCI – Human Anatomy. The initial set of missing is-a relations contained 58 relations for the NCI-A ontology. Out of these 58 relations in the first iteration 9 were repaired by adding relations which introduce new knowledge to the ontology. In total 6 new is-a relations were added and 4 missing is-a relations were derivable.

In the second iteration, 5 out of 55 is-a relations were repaired by adding new relations while repairing actions for the 50 other is-a relations were unchanged. All 5 is-a relations which were repaired by adding new relations to the ontology are is-a relations which were repaired by repairing actions containing only the missing is-a relation from the first iteration. This exemplifies why it is beneficial to consider already repaired is-a relations in subsequent iterations as Source and Target sets for some missing is-a relations can change and more informative solutions might be identified. The input to the third iteration is a set of 54 is-a relations and given that no changes were made, these relations are the final solution.

5.2 Case 2 Experiment – Biotop

This experiment relates to Case 2. In this experiment we used the Biotop ontology from the 2013 OWL Reasoner Evaluation Workshop dataset containing 280 concepts

	It1	It2	It3	It4
Missing	47	41	42	41
Repaired by itself	19	31	38	41
Repaired using new knowledge	28	10	4	0
New relations	26(3)	11	3(1)	0

Fig. 5. Results for debugging the Biotop ontology

and 42 object properties. For the set of missing is-a relations we randomly selected 47 is-a relations. Then the ontology was modified by removing is-a relations which would make the selected is-a relations derivable. The unmodified ontology was used as domain knowledge in the experiment. The results for debugging Biotop ontology are presented in Figure 5.

The debugging process took 4 iterations. In the first iteration 28 relations were repaired by adding new relations. In total 26 new relations were added in the first iteration using axioms containing $\exists$ expressions. For example, for missing is-a relation GreatApe $\sqsubseteq$ Primate we have a repairing action $\{$FamilyHominidaeQuality $\sqsubseteq$ OrderPrimatesQuality$\}$ given that the ontology contains axioms GreatApe $\sqsubseteq \exists$hasInherence.FamilyHominidaeQuality and $\exists$hasInherence.OrderPrimatesQuality $\sqsubseteq$ Primate.

The input to the second iteration contained 41 non-redundant is-a relations (4 redundant is-a relations were removed from the solution in iteration 1). In total 10 is-a relations were repaired by adding new is-a relations. Out of these 10 repaired is-a relations, 5 are relations from the initial set of missing is-a relations while the other 5 are relations which were added in the first iteration. For example, is-a relation Atom $\sqsubseteq$ Entity from the initial set of missing relations can be repaired with $\{$Atom $\sqsubseteq$ MaterialEntity$\}$ given that MaterialEntity $\sqsubseteq$ Entity was added in the previous iteration.

In the third iteration, the input contained 42 is-a relations. In total 4 is-a relations (3 from the initial set of missing is-a relations and 1 from iteration 1) were repaired by adding 3 new relations. Out of the 3 new relations 1 is acquired using axioms containing $\exists$ expressions. Finally, in the fourth iteration no new relations were added and the system outputs the solution.

5.3 Lessons Learned

The experiments have shown the usefulness of our approach. In each of the cases, whether missing is-a relations were identified, or whether we investigated existing is-a relations, our approach identified new information to be added to the ontologies.

The experiments have also shown that the iterative approach to repairing missing is-a relations is beneficial as in all our experiments additional relations were added to the ontology in subsequent iterations. Running the system on already repaired is-a relations gives the opportunity to identify new repairing actions which introduce new knowledge to the ontology. An example of this is found in the BioTop experiment where is-a relations from the initial set of missing is-a relations were repaired by more informative solutions in the third iteration.

Currently, the system removes redundant is-a relations from a solution after every iteration. This step is crucial for producing skyline optimal solutions. However, in situations where an is-a relation is repaired by a relation acquired from the axioms containing $\exists$ expressions it might be advantageous to keep also the missing is-a relation in subsequent iterations even though it is redundant. The reason for this is that the Source set and the Target set for the missing is-a relation might get updated in later iterations and therefore new repairing actions might be identified. One way to solve this is to make it possible in the system to show these missing is-a relations with their Source and Target sets but not to include them in the solution unless they are repaired using new knowledge. For example, let us assume that the missing is-a relation Human $\sqsubseteq$ Primate was repaired in one iteration by a repairing action $\{$Human $\sqsubseteq$ Primate, SpeciesHomoSapiensQuality $\sqsubseteq$ OrderPrimatesQuality$\}$ in which case the second relation was found using $\exists$. In the next iteration the relation GreatApe $\sqsubseteq$ Primate was added to the ontology. If the system removed redundant relation Human $\sqsubseteq$ Primate then relation Human $\sqsubseteq$ GreatApe would not be detected as a possible repairing action for Human $\sqsubseteq$ Primate.

6 Related Work

There is not much work on the *completing of missing is-a structure*. In [26,25] this was addressed in the setting of taxonomies where the problem as well as some preference criteria were defined. Further, an algorithm was given and an implemented system was proposed. We note that the algorithm presented in this paper can be restricted to taxonomies and in that case finds more informative solutions than [26]. A later version of the [26] system, presented in [24], also deals with semantic defects, and was used for debugging ontologies related to a project for the Swedish National Food Agency [15]. An extension dealing with both ontology debugging and ontology alignment is described in [16]. In [23] the problem was formalized as an abduction problem and an algorithm was given for finding solutions for $\mathcal{ALC}$ acyclic terminologies. In [28] we extended the previous formalization by formalizing the role of the domain expert as well as by introducing preference criteria for the solutions to the problem. There is no other work yet on *GTAP*. There is some work on TBox abduction. [14] proposes an automata-based approach to TBox abduction in $\mathcal{EL}$. It is based on a reduction to the axiom pinpointing problem which is then solved with automata-based methods.

Further, there is work that addresses *related topics* but not directly the problem that is addressed in this paper. There is much work on the *detection of missing (is-a) relations* in e.g. ontology learning [4] or evolution [12], using linguistic [13] and logical [6] patterns, or by using knowledge intrinsic to an ontology network [26,15]. As mentioned before, these approaches, in general, do not detect all missing is-a relations. There is also much work on a dual problem to the one addressed in this paper, i.e. the *debugging of semantic defects.* Most of the work on debugging semantic defects aims at identifying and removing logical contradictions from an ontology [11,31,20,19,10], from mappings between ontologies [29,32,17,30] or ontologies in a network [18,15].

Finally, there is also work on other *abductive reasoning problems in (simple) description logics* including concept abduction [5,2,7] and ABox abduction [8,22,3] as defined in [9].

7 Conclusions

In this paper we presented an approach for completing the is-a structure of $\mathcal{EL}$ ontologies. Many biomedical ontologies can be represented by $\mathcal{EL}$ or a small extension thereof. We have also presented an implemented system and evaluated our approach on three biomedical ontologies. The evaluation has shown the usefulness of the system as in all experiments new is-a relations have been identified.

There are a number of directions for future work. We will investigate approaches for more expressive representation languages as well as different preference criteria. Further, we want to investigate methods for dealing with inconsistency and incoherence as well as incompleteness.

Acknowledgments. We thank the Swedish Research Council (Vetenskapsrådet), the Swedish e-Science Research Centre (SeRC) and the Swedish National Graduate School in Computer Science for financial support.

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: 19th Int. Joint Conf. on Artificial Intelligence, pp. 364–369 (2005)
2. Bienvenu, M.: Complexity of abduction in the $\mathcal{EL}$ family of lightweight description logics. In: 11th Int. Conf. on Principles of Knowledge Representation and Reasoning, pp. 220–230 (2008)
3. Calvanese, D., Ortiz, M., Simkus, M., Stefanoni, G.: The complexity of explaining negative query answers in DL-Lite. In: 13th Int. Conf. on Principles of Knowledge Representation and Reasoning, pp. 583–587 (2012)
4. Cimiano, P., Buitelaar, P., Magnini, B.: Ontology Learning from Text: Methods, Evaluation and Applications. IOS Press (2005)
5. Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., Mongiello, M.: A uniform tableaux-based approach to concept abduction and contraction in $\mathcal{ALN}$. In: Int. Workshop on Description Logics, pp. 158–167 (2004)
6. Corcho, O., Roussey, C., Vilches, L.M., Pérez, I.: Pattern-based OWL ontology debugging guidelines. In: Workshop on Ontology Patterns, pp. 68–82 (2009)
7. Donini, F., Colucci, S., Di Noia, T., Sciasco, E.D.: A tableaux-based method for computing least common subsumers for expressive description logics. In: 21st Int. Joint Conf. on Artificial Intelligence, pp. 739–745 (2009)
8. Du, J., Qi, G., Shen, Y.-D., Pan, J.: Towards practical abox abduction in large OWL DL ontologies. In: 25th AAAI Conf. on Artificial Intelligence, pp. 1160–1165 (2011)
9. Elsenbroich, C., Kutz, O., Sattler, U.: A case for abductive reasoning over ontologies. In: OWL: Experiences and Directions (2006)
10. Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: Ontology Change: Classification and Survey. Knowledge Engineering Review 23(2), 117–152 (2008)
11. Haase, P., Stojanovic, L.: Consistent Evolution of OWL Ontologies. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 182–197. Springer, Heidelberg (2005)
12. Hartung, M., Terwilliger, J., Rahm, E.: Recent advances in schema and ontology evolution. In: Schema Matching and Mapping, pp. 149–190 (2011)
13. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In: 14th Int. Conf. on Computational Linguistics, pp. 539–545 (1992)

14. Hubauer, T., Lamparter, S., Pirker, M.: Automata-based abduction for tractable diagnosis. In: Int. Workshop on Description Logics, pp. 360–371 (2010)
15. Ivanova, V., Bergman, J.L., Hammerling, U., Lambrix, P.: Debugging taxonomies and their alignments: the ToxOntology - MeSH use case. In: 1st Int. Workshop on Debugging Ontologies and Ontology Mappings, pp. 25–36 (2012)
16. Ivanova, V., Lambrix, P.: A unified approach for aligning taxonomies and debugging taxonomies and their alignments. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) ESWC 2013. LNCS, vol. 7882, pp. 1–15. Springer, Heidelberg (2013)
17. Ji, Q., Haase, P., Qi, G., Hitzler, P., Stadtmüller, S.: RaDON — repair and diagnosis in ontology networks. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 863–867. Springer, Heidelberg (2009)
18. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Ontology integration using mappings: Towards getting the right logical consequences. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 173–187. Springer, Heidelberg (2009)
19. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B.: Repairing unsatisfiable concepts in OWL ontologies. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 170–184. Springer, Heidelberg (2006)
20. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.: Debugging Unsatisfiable Classes in OWL Ontologies. J. of Web Semantics 3(4), 268–293 (2006)
21. Kazakov, Y., Krötzsch, M., Simančík, F.: Concurrent classification of $\mathcal{EL}$ ontologies. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 305–320. Springer, Heidelberg (2011)
22. Klarman, S., Endriss, U., Schlobach, S.: Abox abduction in the description logic $\mathcal{ALC}$. J. of Automated Reasoning 46, 43–80 (2011)
23. Lambrix, P., Dragisic, Z., Ivanova, V.: Get my pizza right: Repairing missing is-a relations in $\mathcal{ALC}$ ontologies. In: Takeda, H., Qu, Y., Mizoguchi, R., Kitamura, Y. (eds.) JIST 2012. LNCS, vol. 7774, pp. 17–32. Springer, Heidelberg (2013)
24. Lambrix, P., Ivanova, V.: A unified approach for debugging is-a structure and mappings in networked taxonomies. J. of Biomedical Semantics 4, 10 (2013)
25. Lambrix, P., Liu, Q.: Debugging the missing is-a structure within taxonomies networked by partial reference alignments. Data & Knowledge Engineering 86, 179–205 (2013)
26. Lambrix, P., Liu, Q., Tan, H.: Repairing the Missing is-a Structure of Ontologies. In: Gómez-Pérez, A., Yu, Y., Ding, Y. (eds.) ASWC 2009. LNCS, vol. 5926, pp. 76–90. Springer, Heidelberg (2009)
27. Lambrix, P., Strömbäck, L., Tan, H.: Information Integration in Bioinformatics with Ontologies and Standards (chapter 8). In: Bry, F., Małuszyński, J. (eds.) Semantic Techniques for the Web. LNCS, vol. 5500, pp. 343–376. Springer, Heidelberg (2009)
28. Lambrix, P., Wei-Kleiner, F., Dragisic, Z., Ivanova, V.: Repairing missing is-a structure in ontologies is an abductive reasoning problem. In: 2nd Int. Workshop on Debugging Ontologies and Ontology Mappings, pp. 33–44 (2013)
29. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Repairing Ontology Mappings. In: 22nd Nat. Conf. on Artificial Intelligence, pp. 1408–1413 (2007)
30. Qi, G., Ji, Q., Haase, P.: A conflict-based operator for mapping revision. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 521–536. Springer, Heidelberg (2009)
31. Schlobach, S.: Debugging and Semantic Clarification by Pinpointing. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 226–240. Springer, Heidelberg (2005)
32. Wang, P., Xu, B.: Debugging ontology mappings: a static approach. Computing and Informatics 27, 21–36 (2008)