A System for Debugging Missing Is-a Structure in $\mathcal{EL}$ Ontologies

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

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

Abstract. With the increased use of ontologies in semantically-enabled applications, the issue of debugging defects in ontologies has become increasingly important. These defects can lead to wrong or incomplete results for the semantically-enabled applications. Debugging consists of the phases of detection and repairing. In this paper we introduce a system for repairing a particular kind of defects, i.e. missing relations in the is-a hierarchy of $\mathcal{EL}$ ontologies.

1 Introduction

Developing ontologies is a difficult task and it is often the case that the ontologies are incomplete or incorrect. More and more ontologies are used in semantically-enabled applications. Defects in these ontologies can cause incomplete or incorrect results so ontology debugging is a crucial step for acquiring high-quality results in these applications.

In this demonstration paper, we focus on missing is-a relations which are a type of modelling defects. This type of defects requires domain knowledge to detect and resolve. We consider ontologies that are represented by description logics (DLs), more specifically represented by TBoxes in $\mathcal{EL}$. $\mathcal{EL}$ is highly relevant for the representation of lightweight ontologies. For instance, several of the major ontologies in the biomedical domain, e.g., SNOMED[1] and Gene Ontology [1], can be represented in $\mathcal{EL}$ or small extensions thereof [2].

In this demonstation paper we briefly introduce the system introduced in [4]. We describe the system (Section 2) and an example run (Section 3). For the theory, the algorithm as well as more detailed discussion of the experiments we refer to [4]. In Section 4 we introduce the demonstration.

2 Approach

Debugging missing is-a structure consists of two phases, detection and repair. In the detection phase, missing is-a relations are identified while in the repair phase the idea is to make these identified missing is-a relations derivable in the ontology. If all missing is-a relations were identified in the detection phase, the repair phase would be straightforward as only adding these is-a relations is required. However, in general, detection

[1] http://www.ihtsdo.org/snomed-ct/

algorithms do not detect all missing is-a relations and in most cases only few. In cases when only some missing is-a relations are detected there are different approaches for repairing missing is-a structure.

In our setting we assume that our ontology is represented using a TBox T. Further, a detection algorithm or a domain expert has provided a set M of missing is-a relations (but not necessarily all) for the ontology. Then we want to identify a set of is-a relations S such that $T \cup S \models M$. We require that relations in S and M are is-a relations between named concepts as well as that the is-a relations in S should be correct according to the domain. In general, the set of all is-a relations using concepts in T that are correct according to the domain is not known beforehand. If this set was given then we would only have to add this to the ontology. The common case is that we do not have this set, but instead can rely on a domain expert that can decide whether an is-a relation is correct according to the domain. The role of the domain expert can be formalized by an oracle function that returns true or false given an is-a relation. The formal definitions of the problem can be found in [4].

While our earlier work focused on taxonomies [7, 5], in this work we focus on repairing missing is-a relations in $\mathcal{EL}$ ontologies. A TBox in $\mathcal{EL}$ ontologies is a finite set of general concept inclusions of the form $C \sqsubseteq D$ where C and D represent concept descriptions. Concept descriptions in $\mathcal{EL}$ are inductively formed using concept names, role names and concepts constructors which include the top concept, conjuction and existential restriction. In our approach for repairing missing is-a relations we require that the TBox is normalized as described in [2]. 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, $A_1, \ldots, A_n$ and B are concept names and r is a role.

Given that we are dealing with normalized $\mathcal{EL}$ ontologies the algorithm for repairing missing is-a relations uses the following intuitions. Given missing is-a relation $A \sqsubseteq B$:

1. if $A \sqsubseteq C$ and $D \sqsubseteq B$ are derivable from the ontology, then adding $C \sqsubseteq D$ would make the missing is-a relation derivable. Therefore, to acquire possible logical solutions we form two sets, Source and Target, containing the superconcepts of A and the subconcepts of B, respectively. Any is-a relation $C \sqsubseteq D$ such that $C \in$ Source and $D \in$ Target would be a logical solution for repairing $A \sqsubseteq B$.
2. if the ontology contains axioms $A \sqsubseteq \exists r.C$ and $\exists r.D \sqsubseteq B$ then adding is-a relation $C \sqsubseteq D$ would make $A \sqsubseteq B$ derivable.
3. if the ontology contains axioms $A \sqsubseteq \exists r.C$, $\exists r.D \sqsubseteq B$ and is-a relations $C \sqsubseteq F$ and $G \sqsubseteq D$ are derivable in the ontology then $F \sqsubseteq G$ would be a logical solution for the missing is-a relation $A \sqsubseteq B$. This intuition corresponds to generating Source and Target sets for the identified logical solution in the second intuition.

Following the above intuitions we identify logical solutions but not necessarily solutions that are correct according to the domain. Therefore, it is necessary to validate logical solutions with respect to the domain. The repair for the complete set of missing is-a relations is formed by taking the union of repairs for individual missing is-a relations. Any element in the repair for the complete set of missing is-a relations which is not in the initial set of missing is-a relations can be considered as a new missing is-a relation (which was not detected earlier). These new missing is-a relations can then be used as input for a new iteration of the process, thus possibly finding additional solutions.

We have implemented a system for repairing missing is-a structure in $\mathcal{EL}$ ontologies based on the described approach. The input to the system is a set of missing is-a relations which have been validated to be correct according to the domain. The repairing process is semi-automatic and requires interaction with the user who acts as an oracle and decides whether an is-a relation is correct according to the domain. The system has been implemented in Java and uses the ELK reasoner (version 0.4.1) [6] to calculate implicit entailments in the ontology.

3 Use

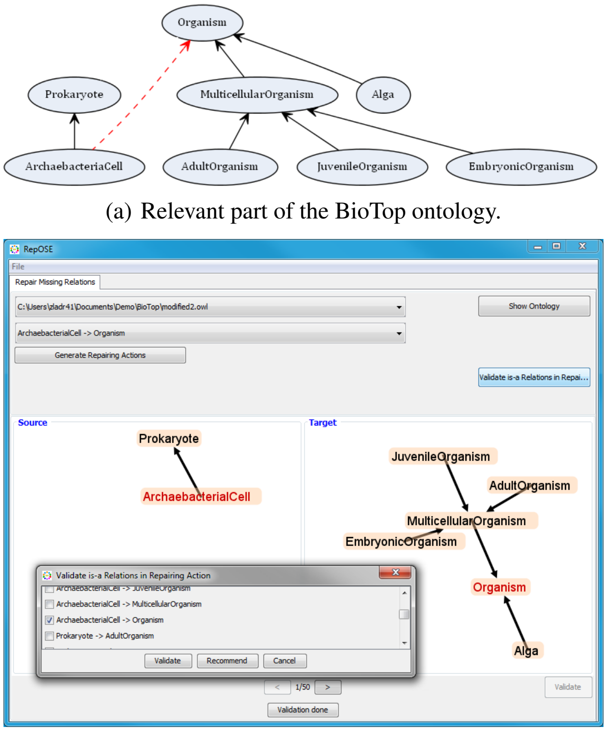
In order to demonstrate the use of the system, let us consider the process of repairing the BioTop ontology from the 2013 OWL Reasoner Evaluation Workshop [4]. The ontology contains 280 concepts and 42 object properties. The set of missing is-a relations consists of 47 is-a relations which were randomly selected in the ontology. Then the ontology was modified by removing relations from the ontology which would make the selected is-a relations derivable. The unmodified ontology has been used as domain knowledge.

The repairing process starts with the user loading the ontology and missing is-a relations into the system and pressing the button `Generate Repairing Actions`. The system then generates Source and Target sets according to intuition 1 and intuitions 2/3. The loaded missing is-a relations are shown in a drop down list allowing the user to easily switch between missing is-a relations. After selecting one of the missing is-a relations, the system shows Source and Target sets for that is-a relation. To repair the missing is-a relation the user needs to choose is-a relations which are correct according to the domain for that is-a relation. This is done by selecting one element from the Source set and one element from the Target set and pressing the `Validate` button thus validating the is-a relation as a repairing action. The system allows multiple repairing actions for each missing is-a relation.

In the BioTop use case the system generates Source and Target sets for 50 is-a relations, 47 according to intuition 1 and 3 according to intuitions 2/3. An example of a Source and Target set generated according to intuition 1 is given in Figure 1(b). Given that is-a relation ArchaebacteriaCell $\sqsubseteq$ Organism is in the input set of missing is-a relations the is-a relation is automatically validated to be correct according to the domain. In this case, the domain expert will also validate is-a relation Prokaryote $\sqsubseteq$ Organism as correct thus introducing new knowledge to the ontology.

An example of Source and Target sets acquired following the intuitions 2/3 is shown in Figure 2(b). In this case, we have the Source and Target set for the is-a relation SpeciesHomoSapiensQuality $\sqsubseteq$ FamilyHominidaeQuality which is a logical solution for the missing is-a relation Human $\sqsubseteq$ GreatApe given that the ontology contains axioms Human $\sqsubseteq$ $\exists$hasInherence.SpeciesHomoSapiensQuality and $\exists$hasInherence.FamilyHominidaeQuality $\sqsubseteq$ GreatApe (Figure 2(a)). Unlike the previous example, the is-a relation SpeciesHomoSapiensQuality $\sqsubseteq$ FamilyHominidaeQuality has to be validated explicitly by the domain expert as it was found using intuitions

2/3. In addition, is-a relation GenusHomoQuality $\sqsubseteq$ FamilyHominidaeQuality will also be validated by the domain expert as it is correct according to the domain.



(a) Relevant part of the BioTop ontology.

C:\Users\zladr41\Documents\Demo\BioTop\modified2.owl

ArchaebacterialCell -> Organism

Validate is-a Relations in Repairing Action

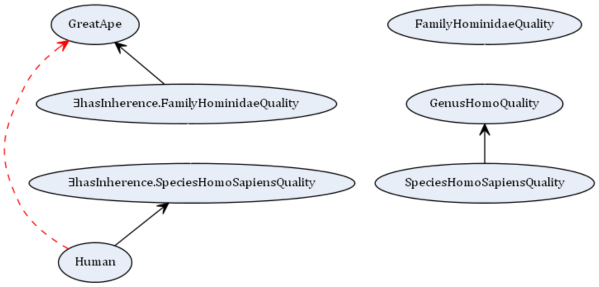
□ ArchaebacterialCell -> JuvenileOrganism
□ ArchaebacterialCell -> MulticellularOrganism
☑ ArchaebacterialCell -> Organism
□ Prokaryote -> AdultOrganism

(b) Screenshot from the system

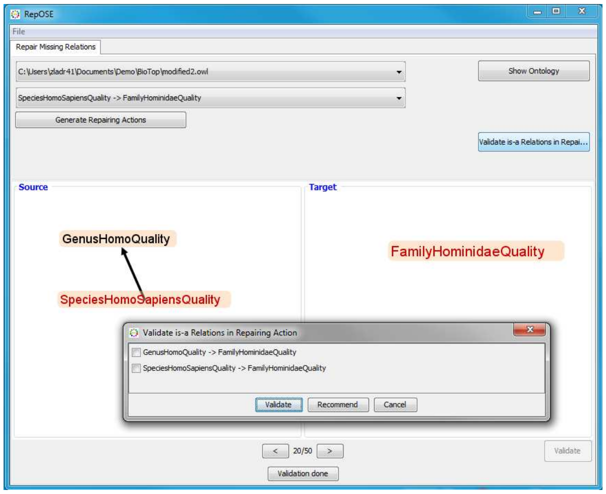
Fig. 1: Repairing ArchaebacteriaCell $\sqsubseteq$ Organism.

Clicking the `Validate Is-a Relations in Repairing Action` button opens a pop-up window where the user has a possibility to check validated is-a relations or see which relations can be validated. On this screen the user can also do the actual validation or remove already validated relations. By clicking on the `Recommend` button the system will recommend correct is-a relations by querying external sources. Currently the recommendations are acquired from WordNet, UMLS Methathesaurus and Uberon. In Figures 1(b) and 2(b) the validation panel for the is-a relations ArchaebacteriaCell $\sqsubseteq$ Organism and SpeciesHomoSapiensQuality $\sqsubseteq$ FamilyHominidaeQuality, respectively, is given.

The validation phase is ended by clicking on the `Validation Done` button. The user has a possibility to end validation phase at any point. If the user has not dealt with some missing is-a relation then the repairing for that is-a relation would be the missing is-a relation itself (as the missing is-a relations are automatically validated to be correct). The system then calculates a repair for the complete set of missing is-a


GreatApe
FamilyHominidaeQuality
∃hasInherence.FamilyHominidaeQuality
GenusHomoQuality
∃hasInherence.SpeciesHomoSapiensQuality
SpeciesHomoSapiensQuality
Human

(a) Relevant part of the BioTop ontology.



C:\Users\zladr41\Documents\Demo\BioTop\modified2.owl

SpeciesHomoSapiensQuality -> FamilyHominidaeQuality

Validate is-a Relations in Repairing Action

- □ GenusHomoQuality -> FamilyHominidaeQuality
- □ SpeciesHomoSapiensQuality -> FamilyHominidaeQuality

20/50

(b) Screenshot from the system

Fig. 2: Repairing Human ⊑ GreatApe.

relations. This repair is then used as input in the next iteration of the repairing process. If the repairing did not change between iterations, the system outputs the final solution.

In our example, 28 relations are repaired by adding a total of 26 new relations out of which 3 are acquired using intuitions 2/3. The remaining 19 missing is-a relations are repaired by adding the missing is-a relation itself. Before the start of the second iteration the system calculates a new set of non-redundant is-a relations from the union of repairing actions from the first iteration. In total 41 new non-redundant is-a relations are identified (4 redundant is-a relations are removed from the solution in iteration 1). In the next iteration the user is presented with Source and Target sets for a total of 64 is-a relations out of which 23 correspond to repairing actions acquired using intuitions 2/3. However, none of these 23 is-a relations are identified to be correct according to the domain. In this iteration 10 is-a relations are repaired by adding new is-a relations. Four out of these 10 is-a relations are from the initial set of missing is-a relations while others were added in the first iteration. For example, relation Virus $\sqsubseteq$ StructuredBiologicalEntity is repaired by adding relation Virus $\sqsubseteq$ Organism given that the relation Organism $\sqsubseteq$ StructuredBiologicalEntity was added in the first iteration. Figure 3(a) shows the relevant part of the BioTop ontology for the missing is-a relation Virus $\sqsubseteq$ StructuredBiologicalEntity with is-a relations added in the previous iteration marked in green. A screenshot from the system with Source and Target set for the missing is-a relation is given in Figure 3(b).

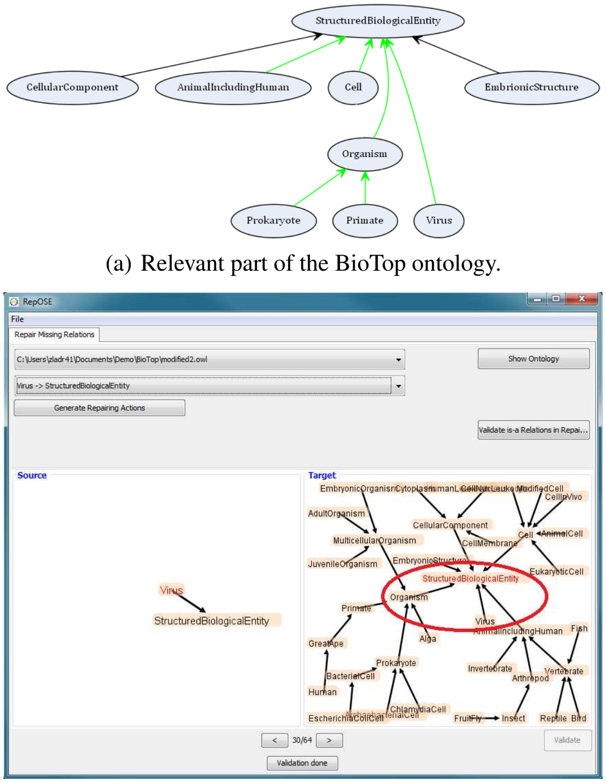
In the third iteration, the user is presented with Source and Target sets for 65 is-a relations out of which 42 are non-redundant is-a relations from the union of repairing actions in the second iteration and 23 are is-a relations which represent repairing actions acquired using intuitions 2/3. Out of these 23 is-a relations only one is validated to be correct according to the domain. Additionally, 2 relations are added in this iteration repairing a total of 4 is-a relations. Out of these 4 repaired is-a relations 3 are from the initial set of missing is-a relations while 1 is from the first iteration.

Finally, in the fourth iteration no new relations are added and the system outputs the solution.

Given that validation can be a time consuming task for large ontologies, the system also implements sessions thus allowing the user to repair ontologies across multiple sessions. To accommodate this, the system implements mechanisms for saving currently validated relations as well as loading previously stored validated relations.

4 Demonstration

In the demonstration we will show two use cases from [4]. The first use case is the one described in Section 3. For the second use case we use Mouse anatomy (AMA) and a fragment of NCI human anatomy ontology (NCI-A) from the Anatomy track of the 2013 Ontology Alignment Evaluation Initiative [3]. The set of missing is-a relations for these two experiments were obtained using a logic-based approach presented in [7] which uses an alignment between these two ontologies to generate possible missing is-a relations which are then validated by a domain expert. The set of missing is-a relations consists of 94 is-a relations for the AMA ontology and 58 for the NCI-A ontology. The missing is-a relations were repaired by adding 101 is-a relations to AMA and 54 is-a



C:\Users\zladr41\Documents\Demo\BioTop\modified2.owl

Virus -> StructuredBiologicalEntity

(a) Relevant part of the BioTop ontology.

(b) Screenshot from the system

Fig. 3: Repairing Virus ⊑ StructuredBiologicalEntity.

relations to NCI-A. Out of 101 is-a relations in the repair for AMA 47 represent new is-a relations which do not appear in the initial set of missing is-a relations. In the case of NCI-A 10 new is-a relations were added.

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. M Ashburner, CA Ball, JA Blake, D Botstein, H Butler, JM Cherry, AP Davis, K Dolinski, SS Dwight, JT Eppig, MA Harris, DP Hill, L Issel-Tarver, A Kasarskis, S Lewis, JC Matese, JE Richardson, M Ringwald, GM Rubin, and G Sherlock. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, 25(1):25–29, 2000.
2. F Baader, S Brandt, and C Lutz. Pushing the $\mathcal{EL}$ envelope. In *19th Int Joint Conf on Artificial Intelligence*, pages 364–369, 2005.
3. B Cuenca Grau, Z Dragisic, K Eckert, J Euzenat, A Ferrara, R Granada, V Ivanova, E Jiménez-Ruiz, AO Kempf, P Lambrix, A Nikolov, H Paulheim, D Ritze, F Scharffe, P Shvaiko, C Trojahn, and O Zamazal. Results of the ontology alignment evaluation initiative 2013. In *Proc. 8th workshop on ontology matching (OM)*, pages 61–100, 2013.
4. Z Dragisic, P Lambrix, and F Wei-Kleiner. Completing the is-a structure of biomedical ontologies. In *10th International Conference on Data Integration in the Life Sciences*, 2014.
5. V Ivanova and P Lambrix. A unified approach for aligning taxonomies and debugging taxonomies and their alignments. In *10th Extended Semantic Web Conf*, pages 1–15, 2013.
6. Y Kazakov, M Krötzsch, and F Simančík. Concurrent classification of $\mathcal{EL}$ ontologies. In *10th Int Semantic Web Conf*, pages 305–320. 2011.
7. P Lambrix and Q Liu. Debugging the missing is-a structure within taxonomies networked by partial reference alignments. *Data & Knowledge Engineering*, 86:179–205, 2013.