LogMap 2.0: towards logic-based, scalable and interactive ontology matching

Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Yujiao Zhou
Department of Computer Science
University of Oxford
Wolfson Building, Parks Road, OX1 3QD, UK
{ernesto,berg,yujiao.zhou}@cs.ox.ac.uk

ABSTRACT

In this paper we present a much improved version of LogMap, a highly scalable ontology matching system with 'built-in' reasoning and diagnosis capabilities. LogMap 2.0 is not only more scalable and robust than its predecessor, but it also provides the necessary infrastructure for domain experts to interactively contribute to the matching process.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms

Algorithms

1. INTRODUCTION

Despite the impressive state of the art, large-scale biomedical ontologies still pose serious challenges to existing ontology matching tools [11, 3].

Insufficient scalability. Existing matching tools can efficiently deal with small ontologies (i.e. less than 500 classes); however, medium-sized and large-scale ontologies are still beyond the reach of most existing systems. For example, the input ontologies in the largest test case of the OAEI 2011 initiative contain 2000-3000 classes, and only 6 out of 16 tools were able to process these ontologies [2].

Logical inconsistencies. OWL ontologies have well-defined semantics based on first-order logic, and mappings are commonly represented as OWL class axioms. Many existing tools, however, disregard the semantics of the input ontologies; thus, they are unable to detect and repair inconsistencies that logically follow from the union of the input ontologies and the computed mappings. Although there is a growing interesting in applying reasoning techniques to ontology matching (e.g., [4, 10, 9]), reasoning is known to severely aggravate the scalability problem.

Curation. Manual curation of ontology mappings is costly and time consuming, especially if the input ontologies are large, and millions of candidate mappings can be found. Several tools have been recently developed to assist the user in the curation of an input set of mappings [6, 8]; these tools, however, do not scale for large ontologies and mapping sets.

The LogMap tool[1] successfully addresses the first two aforementioned challenges [5, 7].

In this paper we present LogMap 2.0, a much improved version of LogMap. LogMap 2.0 is not only more scalable and robust than its predecessor, but it also provides the necessary infrastructure for domain experts to interactively contribute to the matching process, thus facilitating curation.

2. THE ARCHITECTURE OF LOGMAP 2.0

The main steps in LogMap 2.0 are summarised in Table 1. The steps marked with a tickmark (✓) are those where user intervention is possible. We next briefly describe each step.[2]

Overlapping estimation. LogMap 2.0 over-estimates the set of possible mappings using a very efficient lexical algorithm. Essentially, such (rough) over-estimation consists of the pairs of entities whose labels have a similar lexical component (e.g., NCI:CommonCarotidArteryBranch and FMA:BranchOfCommonCochlearArtery). LogMap 2.0 then uses module extraction techniques [1] to compute the modules $\mathcal{O}_1'$ and $\mathcal{O}_2'$ of the input ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ for the entities involved in these possible mappings, thus considerably reducing the size of the matching problem.

Lexical indexation. LogMap 2.0 indexes the labels of the classes in $\mathcal{O}_1'$ and $\mathcal{O}_2'$ as well as their lexical variations. Then, it constructs an 'inverted' lexical index for each of these ontologies (see [5] for details), which will be exploited to efficiently compute an initial set of candidate mappings.

Computation of Candidate Mappings. LogMap 2.0 efficiently computes a set of initial *candidate mappings* by intersecting the inverted indices of $\mathcal{O}_1'$ and $\mathcal{O}_2'$ (see [5] for details). Entities involved in such candidate mappings have a very high lexical similarity. Unlike its predecessor, LogMap 2.0 heuristically splits candidate mappings into two groups $\mathcal{M}^{act}$ and $\mathcal{M}^{?}$. The set $\mathcal{M}^{act}$ contains the candidate mappings that are likely to be "correct" (e.g., FMA:CarpalBone $\equiv$

[1] `http://www.cs.ox.ac.uk/isg/projects/LogMap/`

[2] Note that a set of initial mappings can be given as input, although such initial set will be empty in many cases.

Input: $\mathcal{O}_1$, $\mathcal{O}_2$: input ontologies; $\mathcal{M}$: input mappings
Output: $\mathcal{M}$: mappings; $\mathcal{O}_1^{'}$, $\mathcal{O}_2^{'}$: fragments.

1: $\langle \mathcal{O}_1^{'}, \mathcal{O}_2^{'} \rangle := \mathsf{OverlappingEstimation}(\mathcal{O}_1, \mathcal{O}_2)$
2: Compute lexical indexation of $\mathcal{O}_1^{'}$ and $\mathcal{O}_2^{'}$
3: $\langle \mathcal{M}^{act}, \mathcal{M}^{?} \rangle := \mathsf{CandidateMappings}(\mathcal{O}_1^{'}, \mathcal{O}_2^{'})$
4: $\mathcal{M} := \mathcal{M} \cup \mathsf{Repair}(\mathcal{M}, \mathcal{M}^{act})$
5: $\mathcal{M}^{act} := \emptyset$
6: Compute structural indexation for $\mathcal{O}_1^{'}$, $\mathcal{O}_2^{'}$ and $\mathcal{M}$
7: Extract mappings $\mathcal{M}^{\perp} \subseteq \mathcal{M}^{?}$ in conflict with $\mathcal{M}$
8: $\mathcal{M}^{?} := \mathcal{M}^{?} \setminus \mathcal{M}^{\perp}$
9: Compute partial order of $\mathcal{M}^{?}$
10: **if** ((✓) Interactive Process) and $\mathcal{M}^{?} \neq \emptyset$ **then**
11: (✓) Select $\mathcal{M}^{act} \subseteq$ top-k mappings in $\mathcal{M}^{?}$
12: $\mathcal{M}^{?} := \mathcal{M}^{?} \setminus \mathcal{M}^{act}$
13: Go to Step (4)
14: **else**
15: Automatically select $\mathcal{M}^{act} \subseteq \mathcal{M}^{?}$
16: $\mathcal{M} := \mathcal{M} \cup \mathsf{Repair}(\mathcal{M}, \mathcal{M}^{act})$
17: **end if**
18: **return** $\mathcal{M}$, $\mathcal{O}_1^{'}$ and $\mathcal{O}_2^{'}$

Table 1: LogMap 2.0 interactive method

NCI:Carpal_Bone where both classes are classified as *bones* in their respective ontologies); in contrast, $\mathcal{M}^{?}$ contains the mappings that may require expert curation (e.g., the classes FMA:Trapezoid and NCI:Trapezoid are lexically equivalent, however FMA:Trapezoid is classified as *bone* whereas NCI:Trapezoid is a *polygon*).

Mapping repair. LogMap 2.0 projects the (classified) input ontologies into Horn propositional logic. The mapping sets $\mathcal{M}$ (output or 'fixed' mappings computed thus far) and $\mathcal{M}^{act}$ ('active' mappings) are also represented in Horn propositional logic. LogMap 2.0 implements a sound and highly scalable (but possibly incomplete) reasoning algorithm for detecting unsatisfiable classes. Each unsatisfiable class is repaired using a diagnosis algorithm that only deletes active mappings. The remaining active mappings are then considered as 'fixed', and are hence included in $\mathcal{M}$.

Structural indexation. The classified ontologies together with the mappings in $\mathcal{M}$ are indexed using an interval labelling schema (see [5] for details). This index significantly reduces the cost of computing typical queries (e.g. ancestor-descendant relationships, disjointness between classes) over large class hierarchies.

Conflict detection w.r.t. the structural index. Structural indexation allows to efficiently detect whether two given classes are disjoint. Thus, LogMap 2.0 automatically discards mappings from $\mathcal{M}^{?}$ aligning entities that are known to be disjoint and will obviously lead to logical errors.

User intervention. LogMap 2.0 uses several heuristics to construct a partial order of the set $\mathcal{M}^{?}$, which determines the order in which mappings in $\mathcal{M}^{?}$ are presented to the human expert for approval or rejection. After each expert decision, LogMap 2.0 prunes the set $\mathcal{M}^{?}$ as much as possible and updates the partial order accordingly. This interactive process continues until $\mathcal{M}^{?}$ is empty or the user decides to conclude the manual curation. In the latter case, LogMap 2.0 will use heuristics similar to those implemented in LogMap 1.0 to make the remaining decisions automatically.

3. PRELIMINARY RESULTS

We have conducted preliminary experiments with the ontologies FMA (version 2.0) and NCI (version 08.05*d*) to provide an upper bound to the number of questions asked to the human expert. Furthermore, we also show that, even without human intervention, LogMap 2.0 improves the results obtained by LogMap 1.0 on these ontologies.

LogMap 2.0 identifies 2,692 candidate mappings between FMA and NCI (step 3 in Table 1) where $\mathcal{M}^{act} = 2,051$ and $\mathcal{M}^{?} = 641$. In the *repair step* 61 mappings from $\mathcal{M}^{act}$ are discarded. In the *conflict detection step*, 56 mappings from $\mathcal{M}^{?}$ are also discarded since they are in conflict with $\mathcal{M}$. Thus, in the interactive process, the user would need to assess at most 585 mappings. LogMap 2.0 gives an ouput of 2,575 mappings which have a precision of 0.887, a recall of 0.794 and a F-measure of 0.838, which improves our previous results over FMA and NCI in terms of precision and F-measure (see [5]).

The interactive features of LogMap 2.0 are still at a very early stage of development. We believe, however, that our preliminary experiments are encouraging and they can be significantly improved.

4. REFERENCES

[1] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Just the right amount: extracting modules from ontologies. In *Proc. of WWW*, 2007.

[2] J. Euzenat et al. First results of the Ontology Alignment Evaluation Initiative 2011. *Proc. of the International Workshop on Ontology Matching*, 2011.

[3] J. Euzenat, C. Meilicke, H. Stuckenschmidt, P. Shvaiko, and C. Trojahn. Ontology Alignment Evaluation Initiative: six years of experience. *J Data Semantics*, 2011.

[4] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka. Ontology matching with semantic verification. *J of Web Semantics*, 7(3):235–251, 2009.

[5] E. Jiménez-Ruiz and B. Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *10th International Semantic Web Conference*, 2011.

[6] E. Jimenez-Ruiz, B. Cuenca Grau, I. Horrocks, and R. Berlanga. Ontology integration using mappings: Towards getting the right logical consequences. In *Proc. of European Semantic Web Conference*, 2009.

[7] E. Jimenez-Ruiz, A. Morant, and B. Cuenca Grau. LogMap results in the OAEI 2011. In *Proc. of the International Workshop on Ontology Matching*, 2011.

[8] C. Meilicke, H. Stuckenschmidt, and O. Sváb-Zamazal. A reasoning-based support tool for ontology mapping evaluation. In *The 6th European Semantic Web Conference (ESWC)*, 2009.

[9] J. Noessner and M. Niepert. CODI: Combinatorial optimization for data integration. Results for OAEI 2010. In *Proc. of OM Workshop*, 2010.

[10] Q. Reul and J. Z. Pan. KOSIMap: Use of description logic reasoning to align heterogeneous ontologies. In *Proc. of DL Workshop*, 2010.

[11] P. Shvaiko and J. Euzenat. Ten challenges for ontology matching. In *On the Move to Meaningful Internet Systems (OTM Conferences)*, 2008.