Learning Complex Ontology Alignments – A Challenge for ILP Research

Heiner Stuckenschmidt, Livia Predoiu, Christian Meilicke

Computer Science Institute
University Mannheim, Germany
{heiner,livia,christian}@informatik.uni-mannheim.de

Abstract. In this paper, we propose the task of learning complex logical mappings between ontologies as a challenging task for ILP research. We motivate the need for complex ontology mappings using an example, formally define the task of learning complex mappings and identify a number of challenges for research on this issue in terms of tractability and uncertainty handling.

1 Background: Ontology Alignment

The integration of information from heterogeneous sources is one of the major challenges of modern information technology. Researchers from different areas including databases, knowledge representation and more recently in semantic web technologies have addressed this problem. Ontologies have been identified as a key technology for resolving semantic heterogeneity by providing common terms as well as formal specifications of their intended meaning in some logic. In large distributed environments with a high number of different information sources, however, it is unlikely that people will agree on a single ontology as the basis for integrating information. Here, we often face a situation where multiple ontologies describing the very same domain co-exist. In such a situation, we first have to integrate the different ontologies before they can serve as a basis for integrating information.

A common way of integrating different ontologies describing the same or largely overlapping domains is to use formal representations of semantic correspondences between their concepts and relations - also referred to as 'ontology mappings'. Manual approaches for identifying semantic correspondences are often not feasible since real world ontologies, for example in the medical domain, often contain several thousand concepts. As a response to this problem, a number of automatic and semi-automatic tools for generating hypotheses about semantic correspondences have been developed (see [4] for an overview).

A serious limitation of almost all existing tools is the inability to identify complex mappings. In particular, most systems are only able to identify simple equivalence statements between class or relation names. The true semantic relation between elements of different ontologies, however, is often more complex. In the following section, we give an example that illustrates the need for complex mappings and the limitations of existing systems. We propose the automatic identification of complex mappings between ontologies as an interesting and relevant challenge for the ILP community. The problem is

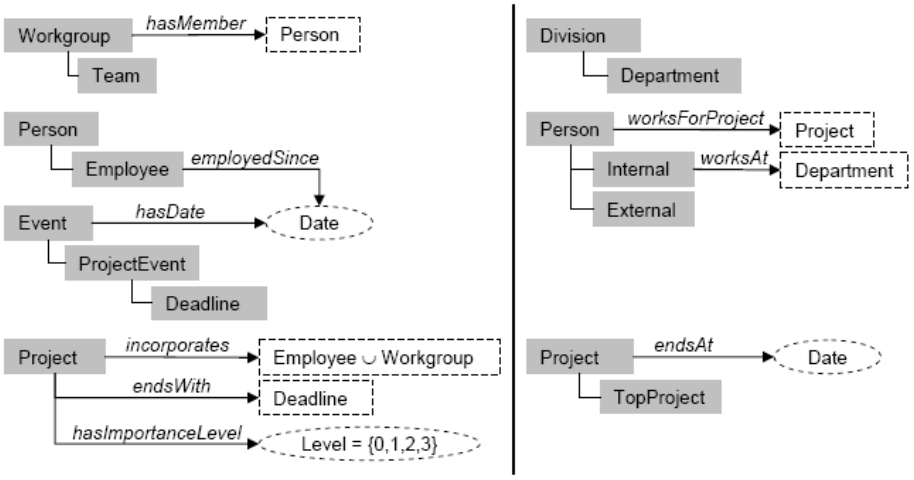

Fig. 1. An example of two ontology fragments describing employees and projects of a company. A labeled square represents a concept, a labeled ellipse a datatype, and a labeled arrow a role. The subsumption hierarchy of concepts is represented by indentation. Domain and range of a property are restricted to be the concepts connected by the accordant arrow.

of great practical importance as ontology matching is the Achilles heal of important research areas with a high potential impact, in particular the semantic web and enterprise application integration. Further, as we will argue below, the use of ILP as a paradigm for addressing the problem is a natural fit as the goal is to learn complex logical rules based on instances and background knowledge.

2 An Example Scenario

In the following example we focus on two ontologies describing human resources, projects and related topics. These ontologies are presented in figure 1. We refer to the ontologies as $\mathcal{O}_1$ (left side of the figure) and $\mathcal{O}_2$ (right side), and we use prefix $i\#$ to refer to the entities of $\mathcal{O}_i$. While both ontologies share some essential concepts, they differ especially with respect to the relations expressed via the properties. In particular these differences make the alignment process erroneous and require complex correspondences to express the correct semantic relations.

To better understand the capabilities of todays state of the art matching systems, we aligned these two ontologies with the Falcon-AO matching system [8], one of the top matching systems participating at the ontology alignment evaluation 2006 and 2007 [3]. As a result Falcon-AO generates two correspondences, namely $1\#\mathit{Project}(x) \leftrightarrow 2\#\mathit{Project}(x)$ and $1\#\mathit{Person}(x) \leftrightarrow 2\#\mathit{Person}(x)$. Are these correspondences sufficient to express the semantic relations that we might be interested in? Suppose that we would like to transfer instance data from $\mathcal{O}_1$ to $\mathcal{O}_2$. Which projects in $\mathcal{O}_1$ have to be classified as top projects in $\mathcal{O}_2$? These are projects with a high level of importance. We could for

example use the following rule for migrating these projects to $\mathcal{O}_2$.

$$2\#TopProject(x) \leftarrow 1\#Project(x) \wedge 1\#hasImportanceLevel(x, 3) \quad (1)$$

What about the deadline of a project? This relation is modeled via a single datatype property in $\mathcal{O}_2$ while we find a chain of properties in $\mathcal{O}_1$. Rule (2) represents this dependency.

$$2\#endsAt(x, z) \leftarrow 1\#endsWith(x, y) \wedge 1\#hasDate(y, z) \quad (2)$$

When we like to know which person are working in which projects, things are getting even more complicated, because the *1#incorporates* property relates both employees and workgroups to projects. We have to use the two rules to cope with the different modeling.

$$2\#worksForProject(x, z) \leftarrow 1\#incorporates(x, z) \wedge 1\#Employee(z) \quad (3)$$

$$2\#worksForProject(x, z) \leftarrow 1\#incorporates(x, y) \wedge 1\#hasMember(y, z) \quad (4)$$

We conclude that ontology alignment requires the use of complex and non-trivial correspondences. Otherwise the completeness of the alignment cannot be guaranteed and the semantic gap between different ontologies cannot be bridged in an appropriate way.

3 Formalization of the Problem

Based on [4] we can formalize the ontology matching problem as follows: Without loss of generality, we consider the case where we have two first-order theories or ontologies $\mathcal{O}_1 = T_1 \cup A_1$ and $\mathcal{O}_2 = T_2 \cup A_2$ given. The T component of an ontology determines the terminological knowledge definition and the A component determines the association of instances with predicates. Each of the ontologies $\mathcal{O}_i$ is represented in the language L_i and each of the ontologies has elements that can be defined by means of elements in the other ontology. Those elements are called the set of matchable elements $Q(\mathcal{O})$ of the ontology $\mathcal{O}$. Note that the set of matchable elements of an ontology depends of the other ontology that is involved in the matching process. The task of ontology matching is now to find correspondences between matchable elements in the two ontologies. Correspondences are 4-tuples $\{e_1, e_2, r, n\}$ such that

- n, a number between 0 and 1, expresses the degree of confidence in the correspondence.
- r is a relation between e_1 and e_2. We only consider implication between formulae as well as statements of the form $e_1 = e_2$ where e_1 and e_2 are constants. Statements of the latter form are called instance equivalences.
- each e_i is a formula represented in L_i. In the spirit of [4], we distinguish between three levels of expressivity. Given a level 0 correspondence, the formulae e_i consist simply of a single predicate. Level 1 corresponds to conjunctions of predicates on the right hand side of the implication relation $\leftarrow$ while the left hand side remains a single predicate. The final level 3 corresponds to arbitrary expressions in the languages L_i.

Based on these definitions, we can now more precisely define the learning task associated with the creation of complex ontology mappings as the ones described in the example above.

Definition 1 (Learning Task). *Given ontologies $\mathcal{O}_1 = T_1 \cup A_1$ and $\mathcal{O}_2 = T_2 \cup A_2$ in languages L_1 and L_2. Further given a mapping $\mathcal{M}$ that defines implications between predicates in T_1 and T_2 as well as equalities between constants in A_1 and A_2. Then the complex mapping learning problem is to find a set H of first order sentences such that:*

1. *Elements of H are of the form: $e_2 \leftarrow e_1$ where e_1 and e_2 are defined as above*
2. $(T_1 \cup T_2 \cup \mathcal{M}) \wedge A_1 \wedge H \models A_2$
3. $(\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{M}) \wedge H \not\models \bot$

From the point of view of Knowledge-Based Inductive Learning, the terminological part of the aligned ontologies together with the level 0 mappings and the instance equivalences play the role of background knowledge, while the instance information is used as training examples. More precisely, the hypothesis should explain the occurrence of instances in terms of A_2 based on their occurrence in A_1. Suppose for example that in A_2 we have instance *1#p-1762* with *1#Project*(*1#p-1762*) and *1#hasImportanceLevel*(*1#p-1762*, 3). If H contains equation 1 we can conclude that *2#TopProject*(*1#p-1762*). Further suppose that $\mathcal{M}$ contains instance equivalence *1#p-1762* = *2#P1762*. Finally, we can conclude that *2#TopProject*(*2#P1762*) and thus give a (partial) explanation of A_2. Further, we claim that the overall model consisting of the two ontologies, pre-existing and learned mappings is consistent to avoid solutions that trivially satisfy the second condition in the definition.

4 Problems and Challenges

The definition above seems to suggest that learning complex ontology mappings is quite straightforward as it can be phrased as a standard ILP learning problem. A closer look reveals, however, that there are a number of practical problems that make the task a challenging one. In this section we discuss three of these problems that we consider to be central to the endeavor and could be starting points for research in this area.

Tractability Work in inductive logic programming often focusses on supporting efficient subsets of first-order logic. In particular, there is a focus on first-order horn rules as a target language for learning and encoding background knowledge. In the context of ontologies, there is also some work concerned with languages that fall into this category (e.g. [6]). However, OWL-Lite and OWL-DL, the primary ontology languages, are based on expressive description logics $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$, respectively. It has been shown that disjunctive Datalog is needed to perform the kind of reasoning needed for testing the second condition of the definition [7]. Even worse, checking the consistency of the overall model cannot be done by a reduction to disjunctive Datalog but requires reasoning about a combined model consisting of description logic ontologies and rule-based mappings. It has been shown that reasoning for such models is highly intractable even for rather inexpressive ontology and rule languages. Recent

work on a first major revision of the OWL language (OWL 2.0) addresses this problem and proposes an integration of ontologies and rules that can be reduced to a very expressive but still decidable description logic $\mathcal{SROIQ}$ [5]. While this enables us in principle to test condition 3 in the definition, tractability is still a major issue claiming for highly optimized learning methods. While existing work on optimizing ILP seems to focus on the problem of dealing with large instance sets, learning ontology mappings comes with new challenges with respect to dealing with expressive models background knowledge that can also be very large - some ontologies contain tens of thousands of axioms.

Uncertainty The approach described above relies on the existence of an initial mapping between predicates and instances in the two ontologies. In order to determine these initial mappings an additional matching step is necessary (compare e.g. [10]). This problem which is referred to as entity and schema matching, respectively, is a research area in its own rights and a variety of methods have been proposed for this purpose. Most of these methods are based on weak criteria such as structural or linguistic similarity. As a result, the learning process has to cope with a significant degree of noise (recent papers report an F-Value of 70% to 90% for the instance matching task). For the problem of creating simple mappings as part of the background knowledge state of the art systems reach a performance of 60% to 90% on real world ontologies [3] which adds additional uncertainty into the learning process. This means that the development of highly robust learning methods is necessary to cope with the task. A possible way to go is to explicitly take the uncertainty introduced by entity and ontology into account [1]. As discussed above, mappings are annotated with a degree of confidence that can be interpreted in thee context of probabilistic ILP approaches [2]. This, however, requires that only probabilistic matching methods have been used to create simple mappings and entity correspondences. The majority of the existing approaches, however are based on different notions of similarity. Providing ways to exploit these similarities in the learning process would be a bit step forward.

Incompleteness and Inconsistency The third condition in our definition that claims the consistency of the overall model poses an additional and unexpected challenge to the learning task. We have shown that existing matching systems cannot guarantee that their result leads to a consistent model. This means that in many cases $(\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{M})$ is already inconsistent. As the logical languages currently used for ontologies and mappings are monotonic, the third condition will never be satisfied in many cases. If we still want to learn complex mappings, we first have to fix the inconsistencies in the simple mappings. This is normally done by removing mappings from $\mathcal{M}$ that cause the overall model to become inconsistent. Simply removing all potential causes of inconsistency, however, will in cases remove too many mappings reducing the degree of overlap between the two ontologies which in turn can be expected to have a negative impact on the accuracy of the learning result. We therefore have to find a way to only remove the 'right' mappings in the sense that the set of removed mappings is minimal and contains only such mappings that are 'wrong' in the sense that their content does not correspond to reality. There are first results in this direction that apply techniques from model-based

diagnoses to the debugging of inconsistent ontology mappings [9], but there is still a lot of space for improvement to get an optimal basis for learning complex mappings.

5 Summary and Conclusions

In this paper, we discussed the generation of complex ontology mappings as a challenging problem to be addressed by the ILP community. We think that this problem is interesting for ILP researchers because (1) the use of ILP for addressing this problem is a natural choice as the definition of the learning problem perfectly matches the ideas of ILP and because (2) a closer look reveals that the problem comes with some interesting challenges with respect to scalability and accuracy. In particular, ILP cannot be seen in isolation here, because the result of the learning phase is directly influenced by the quality of the instance and schema matching step. It is likely that there can be synergies between these two steps that have not been investigated so far, leaving space for interesting and challenging research on the border between ILP and semantic matching.

Acknowledgements: Research reported in this paper has been partially financed by the German Science Foundation (DFG) in the Emmy Noether Programme under contract STU 266/3-1.

References

1. A. Cali, T. Lukasiewicz, L. Predoiu, and H. Stuckenschmidt. Rule-based Approaches for Representing Probabilistic Ontology Mappings. In *Uncertainty Reasoning for the Semantic Web I*. Springer, to appear.
2. L. De Raedt, P. Frasconi, K. Kersting, and S. Muggleton. *Probabilistic Inductive Logic Programming - Theory and Applications*. Springer, 2008.
3. J. Euzenat, A. Isaac, C. Meilicke, P. Shvaiko, H. Stuckenschmidt, O. Svab, V. Svatek, W. van Hage, and M. Yatskevich. Results of the ontology alignment evaluation initiative 2007. In *Proceedings of the ISWC 2007 Workshop on Ontology Matching*, Busan, Korea, November 2007.
4. J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer, 2007.
5. B. C. Grau, B. Motik, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 Web Ontology Language: Profiles. (W3C Working Draft 11 April 2008). Technical report, 2008.
6. B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description Logic Programs: combining logic programs with description logic. In *Proc. of the 12th international conference on World Wide Web (WWW03)*, 2003.
7. U. Hustadt, B. Motik, and U. Sattler. Reasoning in Description Logics by a Reduction to Disjunctive Datalog. In *Journal of Automated Reasoning (JAR)*, 2007.
8. N. Jian, W. Hu, G. Cheng, and Y. Qu. Falcon-AO: Aligning ontologies with falcon. In *K-CAP Workshop on Integrating Ontologies*, 2005.
9. C. Meilicke, H. Stuckenschmidt, and A. Tamilin. Repairing ontology mappings. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, Vancouver, Canada, 2007.
10. H. Qin, D. Dou, and P. LePendu. Discovering executable semantic mappings between ontologies. In *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, volume 4803 of *LNCS*, pages 832–849, 2007.