Towards Rule Learning Approaches to Instance-based Ontology Matching

Frederik Janssen[1], Faraz Fallahi[2]
Jan Noessner[3], and Heiko Paulheim[1]

[1] Knowledge Engineering Group,
TU Darmstadt, Hochschulstrasse 10, 64289 Darmstadt, Germany
{janssen,paulheim}@ke.tu-darmstadt.de
[2] ontoprise GmbH,
An der Raumfabrik 29, 76227 Karlsruhe, Germany
fallahi@ontoprise.de
[3] KR & KM Research Group
University of Mannheim, B6 26, 68159 Mannheim, Germany
jan@informatik.uni-mannheim.de

Abstract. Ontology matching approaches have mostly worked on the schema level so far. With the advent of Linked Open Data and the availability of a massive amount of instance information, instance-based approaches become possible. This position paper discusses approaches and challenges for using those instances as input for machine learning algorithms, with a focus on rule learning algorithms, as a means for ontology matching.

1 Introduction

Today, data integration results produced by various developed automated algorithms and systems are still error-prone. Therefore, the field of data integration is a major challenge of modern information technology and of significance to various research areas. Ontologies can play a key role in resolving semantic heterogeneity by providing a formal description of a domain.

Not only since the advent of Linked Open Data, ontology mappings have become an essential ingredient to ensure the interoperability of data sets using different ontologies. While Linked Open Data is much concerned about the linking of *instances*, mappings on the schema level, e.g. for federating queries across different data sets, are currently under-represented. At the same time, the vast amount of instance data in Linked Open Data allows for developing powerful instance-based approaches to ontology matching to complement the currently predominant schema-level approaches.

Utilizing matching algorithms with lexical distance measure and pattern recognition techniques for automatically finding mappings between ontologies do not always yield the expected results. Where lexical distance measure algorithms fail, machine learning algorithms which are based on symbolic representations can serve as a remedy, at least to some extent.

In the last years, a major research focus has been set on schema based solutions. Hence, a variety of state-of-the-art algorithms for alignment have been developed, which work mostly on traversing schemas and their structure, and only a few approaches explicitly use data about instances [1, 2]. In the scope of the *MappingAssistant* project [4], instance data has been utilized to repair and refine existing ontology alignments.

In this paper, we discuss two possible approaches for employing machine learning for instance-based ontology matching. The basic idea of both of them is to use rule learning to solve this problem. The main advantage of rule learning is its symbolic character, i.e., rules can be interpreted and compared to each other. In a first case study, we use association rule learners on the instance level to discover mappings. A second case study shows how separate-and-conquer rule learners can be employed to further refine mappings. Both case studies aim at demonstrating the ideas and exploring possible challenges.

2 Case Study 1 – Creating Mappings by Rule Learning

In this case study, we focus on cases where a set of instances is contained in different data sets – either identified by identical URIs or connected via `owl:sameAs` statements. While this is rather frequent in Linked Open Data, the OAEI datasets typically do not contain shared instances. Therefore, we have created our own dataset, using an excerpt from *DBpedia*. As DBpedia uses its own ontology as well as *YAGO* for classification, it provides an instance set using different ontologies, which can be used as a benchmark for instance-based ontology matching. To construct our dataset, we used an existing, publicly available partial 1:1 mapping[4] between the DBpedia and the YAGO ontology as a starting point. The dataset contains 169 simple mappings between DBpedia and YAGO. For that dataset, we retrieved all instances that contain at least one of the types in the mapping. That data set has a total of 231,635 instances.

We use association rule mining to discover ontologies, as suggested by [6]. Association rules are used, e.g., to discover sets of items that are likely to be bought together, in order to build recommender algorithms. Likewise, we use them to find out which classes (and other ontology constructs) frequently co-occur, and derive mappings from those co-occurences.

The portions of the two ontologies under consideration are essentially different. For example, the YAGO ontology relies mainly on a rich classification and contains very special classes such as `yago:IndustrialRockMusicGroups`, while the DBpedia relies on relations to describe such classes, e.g., as a `DBpedia-owl:Band` with a value of `DBpedia:IndustrialRock` for `DBpedia-owl:genre`[5]. Thus, the mapping between the two ontologies can only be expressed by a complex one:

$$\begin{aligned} & \texttt{yago:IndustrialRockMusicGroups} \\ \equiv\ & \texttt{DBpedia-owl:Band} \sqcap \exists \texttt{DBpedia-owl:genre}.\{\texttt{DBpedia:IndustrialRock}\} \end{aligned}$$

[4] `http://www.netestate.de/De/Loesungen/DBpedia-YAGO-Ontology-Matching`
[5] See `http://DBpedia.org/resource/Nine_Inch_Nails` for this example.

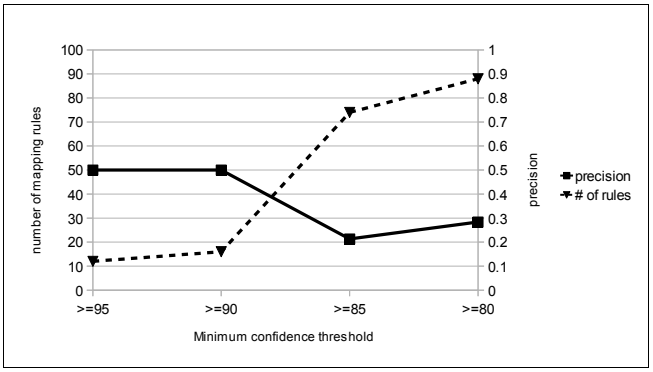

Fig. 1. Matching results for complex mappings

In a first experiment, we tried to infer the simple 1:1 mappings. Using the feature generation toolkit *FeGeLOD* [5], we added a boolean feature for each `rdf:type` property for all the instances in our data set, ending up with a total of 98,414 features. On this data set, we used an association rule mining algorithm to find rules expressing frequent co-occurring `rdf:type` statements, i.e., two classes in different ontologies that share many instances. For example, from two symmetrically occurring rules, we conclude a mapping as follows:

$$\texttt{DBpedia-owl:ProtectedArea} \leftarrow \texttt{yago:Park}$$
$$\texttt{yago:Park} \leftarrow \texttt{DBpedia-owl:ProtectedArea}$$
$$\Rightarrow \quad \texttt{DBpedia-owl:ProtectedArea} \equiv \texttt{yago:Park}$$

As discussed in [5], an F-Measure of up to 25% can be achieved with this naive approach. While this is not as much as state-of-the-art approaches, the mappings found are often hard to find with conventional approaches, as the above example shows.

In a second experiment, we tried to infer *complex* mappings between two classes. To that end, we have created a second dataset which comprises not only the type information as in the first case, but also boolean properties indicating whether there is an incoming or outgoing relation of a certain type. The resulting data set has 117,029 features. From that dataset, we were able to discover complex mappings, such as

$$\geq 1\texttt{DBpedia-owl:name} \sqsubseteq \texttt{yago:Person}$$

Since there is, to the best of our knowledge, no gold standard for complex mappings, we have evaluated our approach by counting the number of rules found, and manually determining the rules that are correct. The results are depicted in Fig. 1 for different thresholds. The experiments show that it is possible to employ association rule learning for discovering ontology mappings. Typical challenges include scalability to larger datasets, learning more expressive rules, and evaluation of complex mappings.

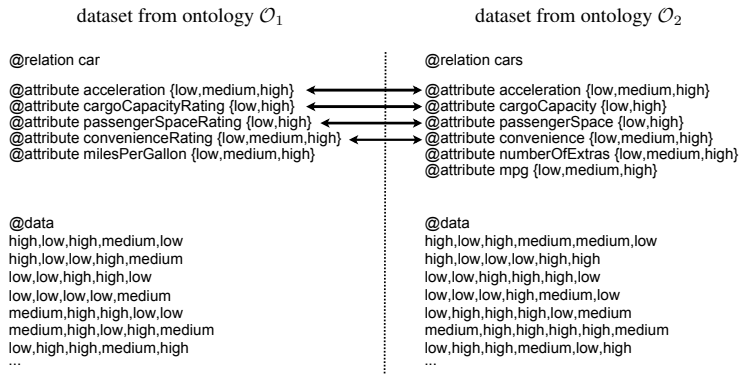

Fig. 2. Excerpt of the input for the separate-and-conquer approach

3 Case Study 2 – Extending Mappings by Rule Learning

In the second case study, our goal is to use classification rule learning to find non-trivial alignments between ontologies, which cannot be found through conventional schema-based matching techniques. We focus on finding additional property alignments, assuming that some alignments between properties defined in two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ are given as an initial mapping, as indicated by the arrows in Fig. 2. Note that in that example, the property `numberOfExtras` that is present in $\mathcal{O}_2$ is missing in $\mathcal{O}_1$.

The goal is to detect additional property alignments between $\mathcal{O}_1$ and $\mathcal{O}_2$. In the example, the alignment still left to detect is `milesPerGallon` $\equiv$ `mpg`.

For each unmapped property, we construct a dataset as input for a rule learner. The datasets are created by including all the instances available from the $\mathcal{A}$-boxes of the two ontologies, and using the unmapped property as a class attribute. In Fig. 2, three classification problems would be created: using `milesPerGallon` as the target class in the data set for $\mathcal{O}_1$, and using `numberOfExtras` and `mpg` as the target class in the data set for $\mathcal{O}_2$.

In our example, we learn rule sets for each non-aligned property q_1 and q_2, using a CN2-like rule learner. While for the illustration of the approach the actual learning algorithm is not that important (given that the output are rules), certainly the performance of the approach is strongly related to the used algorithm (cf. the discussion in Section 4). In the end, the rule learner outputs three rule sets, one for each non-aligned property, e.g., one ruleset R_1 for the property `milesPerGalllon` of $\mathcal{O}_1$, and two rulesets (R_2 and R_3) for the two properties that reside unmapped in $\mathcal{O}_2$ (`numberOfExtras` and `mpg`). The key idea of the approach is to compare the rules found on the instances of the two ontologies: if two rules comprise attribute-value tests for properties that are previously mapped, the consequence is that the target properties (the prediction

of the rules) then can be also mapped, because their characteristics are similar to a certain extent.

To illustrate such a mapping, in the following examples for similar and different rules are shown. The best case is that the rules are identical: $r_{1,1}$: `milesPerGallon=medium`$\leftarrow$`convenienceRating=high` $\wedge$ `acceleration=high.`
$r_{2,1}$: `mpg=medium`$\leftarrow$ `convenience=high` $\wedge$ `acceleration=high.` where $r_{i,j}$ stands for the j-th rule of rule set i. Rule $r_{1,1}$ tests whether the *convenienceRating* is *high* and the *acceleration* is *high*. Due to the predefined mappings, it can be concluded that both rules use the same properties with the same values and hence are similar (with confidence 1.0). Note that all numerical values of all properties are discretized in a preprocessing step for simplification issues. Therefore, rules that are exactly similar may not be so rare. Nevertheless, rules that are found on real world data may not be identical. An example for such different rules is given below.
$r_{1,3}$: `milesPerGallon=high`$\leftarrow$ `acceleration=medium` $\wedge$ `cargoCapacityRating=low.`
$r_{3,3}$: `numberOfExtras=high`$\leftarrow$ `convenience=high` $\wedge$ `passengerSpace=high.`

In that case, the rules are completely dissimilar. Thus, a similarity measure $sim_r(r, r')$ for a pair of rules r and r' is needed. This similarity of individual rules is used then to compute the similarity of the whole rule sets $sim_R(R, R')$. A naive choice to define the similarity of two rule sets could look as follows:

$$sim_R(R, R') = \frac{\sum_{sim_r(r_{1,i}, r_{2,j}) \geq \theta} {}^{(tp(r_{1,i}) + tp(r_{2,j}))}/_2}{{}^{(|D_1| + |D_2|)}/_2} \tag{1}$$

where $|D_i|$ is the total number of examples in dataset D_i, $tp(r)$ yields the number of *true positive* examples covered by rule r (i.e., the examples that are correctly covered by the rule r), and θ is a threshold that decides whether the rules are equal or not. The similarity of single rules can naively be defined by

$$sim_r(r, r') = \begin{cases} 1 \text{ if } r \text{ matches } r' \text{ exactly} \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

Clearly, in real world situations these definitions are too restrictive. Nevertheless, despite their simplicity, first experiments show that they are sufficient. These preliminary experiments with two ontologies containing a few instances show that the approach works. The two rule sets R_1 (rules for `milesPerGalllon`) and R_3 (rules for `mpg`) were identical. In contrast, the rule sets learned for `numberOfExtras` and `milesPerGallon` are completely different, since these are different properties. Thus, the overall similarity value for `numberOfExtras` and `milesPerGallon` is significantly lower than for `mpg` and `milesPerGallon`. Therefore, in the end, the mapping `milesPerGallon` $\equiv$ `mpg` was found while `numberOfExtras` is left unmapped.

The case study shows that rule learning can be used to derive additional mappings from those already found. Open challenges include the definition of metrics for rule comparison, the use of suitable rule learning heuristics [3], and the decomposition of the ontology matching problem into a binary classification problem.

4 Conclusion and Challenges

In this paper, we have discussed approaches of learning and refining non-trivial ontology alignments from instances through utilization of inductive rule learning algorithms. We have shown two case studies which examine how finding and refining ontology mappings can be reformulated as problems of association rule mining and separate-and-conquer rule learning, respectively. While the case studies show the feasibility of employing rule learning algorithms for ontology matching, there are quite a few open research issues and challenges.

The first case study has shown an approach which is especially suitable for discovering complex mappings. Suitable *benchmark datasets* for complex mappings do not exist at the moment. Generating those benchmark sets from Linked Open Data is a suitable option, which at the same time produces new scalability challenges. When dealing with instance-based matching, using a *reasoner in the loop* is also an interesting opportunity for refining mappings. Necessary *preprocessing steps*, e.g., for dealing with numerical data properties, are also a subject of future research.

The second case study has used the similarity of rule sets as a means of assessing validity of ontology mappings. Finding suitable similarity measures of rule sets is a challenge for future work. It may also be a valid direction to use other learning models (e.g. subgroup discovery, or even decision trees) and suitable heuristics, which may allow different and maybe better means for comparison.

Although those challenges are non-trivial and require further deep investigations, we are confident that rule learning is suitable means to instance-based ontology matching. Thus, we are confident that the approaches sketched in this paper will lead to the development of high-performance matching algorithms.

References

1. M. Ehrig and Y. Sure. Ontology Mapping - An Integrated Approach. In C. Bussler, J. Davis, D. Fensel, and R. Studer, editors, *Proceedings of the 1st European Semantic Web Symposium*, volume 3053, pages 76–91, Heraklion, Greece, 2004. Springer Verlag.
2. J. Huber, T. Sztyler, J. Nößner, and C. Meilicke. CODI: Combinatorial Optimization for Data Integration: Results for OAEI 2011. In *Proceedings of the 6th International Workshop on Ontology Matching*, Bonn, Germany, October 2011.
3. F. Janssen and J. Fürnkranz. On the Quest for Optimal Rule Learning Heuristics. *Machine Learning*, 78(3):343–379, March 2010. DOI 10.1007/s10994-009-5162-2.
4. J. Noessner, F. Fallahi, , E. Kiss, and H. Stuckenschmidt. Interactive data integration with mappingassistant. In *Demo Proceedings of the 10th International Semantic Web Conference (ISWC)*, Bonn, Germany, October 2011.
5. H. Paulheim and J. Fürnkranz. Unsupervised Feature Generation from Linked Open Data. In *2nd International Conference on Web Intelligence, Mining, and Semantics*, 2012. to appear.
6. J. Völker and M. Niepert. Statistical Schema Induction. In *Proceedings of the 8th Extended Semantic Web Conference: Linked Open Data Track*, pages 124–138, Berlin, Heidelberg, 2011. Springer-Verlag.