Towards Rule Learning Approaches to Instance-based Ontology Matching

TECHNISCHE UNIVERSITÄT DARMSTADT

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

[1] Knowledge Engineering Group, TU Darmstadt
[2] ontoprise GmbH, Karlsruhe, Germany
[3] KR & KM Research Group, University of Mannheim, Germany

Outline

1. Motivation
2. Case Study 1 - Creating mappings by association rule mining
3. Case Study 2 - Refining mappings by separate-and-conquer rule learning
4. Conclusions and Challenges

Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Main problems of lexical distance measures or pattern recognition for ontology matching:
  - complex mappings cannot be found
  - in multi-lingual schemas there is no lexical similarity at all
- Remedy:
  - machine learning techniques with a focus on symbolic representations (such as rules)
- Advantages:
  - interpretability: enhanced methods for comparison and combination of rules and rule sets
  - capability of finding complex mappings
  - exploiting large-scale instance information, e.g. in LOD

Case Study 1

Creating mappings by association rule mining

TECHNISCHE UNIVERSITÄT DARMSTADT

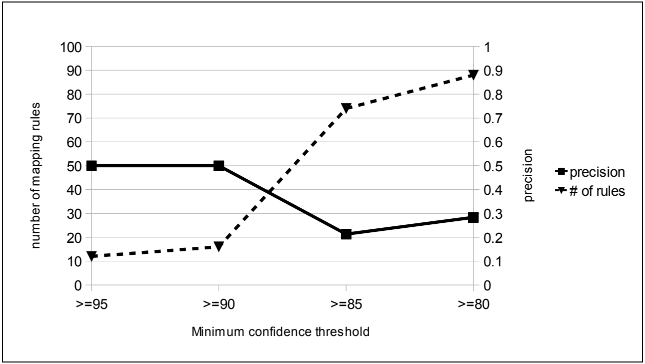
- Approach:
  - exploit instance information from LOD
  - basic idea: classes with similar instance sets are equal
  - use association rule learning to find mappings
  - using binary features for classes
  - conclude mappings for symmetrical rules, e.g.

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

Case Study 1

Preliminary Results

- Data set: manual partial mapping between DBpedia and YAGO



- approach is able to find complex matchings, such as $\geq 1\texttt{DBpedia-owl:name} \sqsubseteq \texttt{yago:Person}$

KE

Case Study 2

Refining mappings by separate-and-conquer rule learning

TECHNISCHE UNIVERSITÄT DARMSTADT

- Given:
  - two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ and some existing mappings, e.g., found by a lexical matcher
- Goal:
  - find additional mappings
- Approach:
  - create datasets for both ontologies using Linked Open Data
  - learn rule sets with the same algorithm on these two datasets for all unmapped entities
  - compute similarity between rule sets

Case Study 2

Refining mappings by separate-and-conquer rule learning

TECHNISCHE UNIVERSITÄT DARMSTADT

dataset from ontology $\mathcal{O}_1$	dataset from ontology $\mathcal{O}_2$
@relation car	@relation cars
@attribute acceleration {low,medium,high} ⟷	@attribute acceleration {low,medium,high}
@attribute cargoCapacityRating {low,high} ⟷	@attribute cargoCapacity {low,high}
@attribute passengerSpaceRating {low,high} ⟷	@attribute passengerSpace {low,high}
@attribute convenienceRating {low,medium,high} ⟷	@attribute convenience {low,medium,high}
@attribute milesPerGallon {low,medium,high}	@attribute mpg {low,medium,high}
@data	@data
high,low,high,medium,low	high,low,high,medium, low
high,low,low,high,medium	high,low,high,medium, low
low,low,high,high,low	low,high,high,low, low
low,low,low,low,medium	low,low,low,high, low
medium,high,high,low,low	low,high,high,high, medium
medium,high,low,high,medium	medium,high,high,high, medium
low,high,high,medium,high	low,high,high,medium,high
...	...
learn ↓ rules	learn ↓ rules

$r_{1,1}$: milesPerGallon=medium←convenience-Rating=high ∧ acceleration=high

$r_{1,2}$: milesPerGallon=high←accelearation=medium ∧ cargoCapacity=low

mpg=medium←convenience=high ∧ acceleration=high

numberOfExtras=high←convenience=high ∧ passengerSpace=high

Case Study 2

Refining mappings by separate-and-conquer rule learning

TECHNISCHE UNIVERSITÄT DARMSTADT

dataset from ontology $\mathcal{O}_1$

```
@relation car

@attribute acceleration {low,medium,high}
@attribute cargoCapacityRating {low,high}
@attribute passengerSpaceRating {low,high}
@attribute convenienceRating {low,medium,high}
@attribute milesPerGallon {low,medium,high}

@data
high,low,high,medium,low
high,low,low,high,medium
low,low,high,high,low
low,low,low,low,medium
medium,high,high,low,low
medium,high,low,high,medium
low,high,high,medium,high
...
```

dataset from ontology $\mathcal{O}_2$

```
@relation cars

@attribute acceleration {low,medium,high}
@attribute cargoCapacity {low,high}
@attribute passengerSpace {low,high}
@attribute convenience {low,medium,high}
@attribute mpg {low,medium,high}

@data
high,low,high,medium, low
high,low,high,medium, low
low,high,high,low, low
low,low,low,high, low
low,high,high,high, medium
medium,high,high,high, medium
low,high,high,medium,high
...
```

learn $\downarrow$ rules

$r_{1,1}$: milesPerGallon=medium←convenience-Rating=high ∧ acceleration=high

$r_{1,2}$: milesPerGallon=high←accelearation=medium ∧ cargoCapacity=low

learn $\downarrow$ rules

mpg=medium←convenience=high ∧ acceleration=high

numberOfExtras=high←convenience=high ∧ passengerSpace=high

Case Study 2

Refining mappings by separate-and-conquer rule learning

TECHNISCHE UNIVERSITÄT DARMSTADT

dataset from ontology $\mathcal{O}_1$	dataset from ontology $\mathcal{O}_2$
@relation car	@relation cars
@attribute acceleration {low,medium,high}	@attribute acceleration {low,medium,high}
@attribute cargoCapacityRating {low,high}	@attribute cargoCapacity {low,high}
@attribute passengerSpaceRating {low,high}	@attribute passengerSpace {low,high}
@attribute convenienceRating {low,medium,high}	@attribute convenience {low,medium,high}
@attribute milesPerGallon {low,medium,high}	@attribute mpg {low,medium,high}
@data	@data
high,low,high,medium,low	high,low,high,medium, low
high,low,low,high,medium	high,low,high,medium, low
low,low,high,high,low	low,high,high,low, low
low,low,low,low,medium	low,low,low,high, low
medium,high,high,low,low	low,high,high,high, medium
medium,high,low,high,medium	medium,high,high,high, medium
low,high,high,medium,high	low,high,high,medium,high
...	...

learn ↓ rules   learn ↓ rules

$r_{1,1}$: milesPerGallon=medium←convenience-Rating=high ∧ acceleration=high

$r_{1,2}$: milesPerGallon=high←accelearation=medium ∧ cargoCapacity=low

mpg=medium←convenience=high ∧ acceleration=high

numberOfExtras=high←convenience=high ∧ passengerSpace=high

Case Study 2

Refining mappings by separate-and-conquer rule learning

TECHNISCHE UNIVERSITÄT DARMSTADT

- Idea:
  - similar rule sets $\rightarrow$ mapping candidate
- possible similarity measures:

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

$$\text{e.g., with } sim_r(r, r') = \begin{cases} 1 \text{ if } r \text{ matches } r' \text{ exactly} \\ 0 \text{ otherwise} \end{cases}$$

where R, R': rule sets, $tp(r_{1,i})$: true positives of the i-th rule of ruleset 1, D_1, D_2: data sets, and θ is a similarity threshold

Conclusions and Challenges

- Conclusions
  - reformulation of ontology matching as problems of (association) rule learning
  - first experiments show that both approaches work
- Challenges
  - create suitable benchmark data sets for complex mappings
  - scaling up to the whole web of data
  - similarity measures for rules and rule sets
  - parameter tuning of rule learning algorithms
  - impact of different rule learning heuristics

Questions?