Towards Rule Learning Approaches to Instance-based Ontology Matching

Frederik Janssen\textsuperscript{1}, Faraz Fallahi\textsuperscript{2}, Jan Noessner\textsuperscript{3}, Heiko Paulheim\textsuperscript{1}

\textsuperscript{1} Knowledge Engineering Group, TU Darmstadt
\textsuperscript{2} ontoprise GmbH, Karlsruhe, Germany
\textsuperscript{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**

- **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

- **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.

\[
\text{DBpedia-owl:ProtectedArea} \leftarrow \text{yago:Park} \\
\text{yago:Park} \leftarrow \text{DBpedia-owl:ProtectedArea} \\
\Rightarrow \text{DBpedia-owl:ProtectedArea} \equiv \text{yago:Park}
\]
Case Study 1
Preliminary Results

- **Data set:** manual partial mapping between DBpedia and YAGO

- approach is able to find complex matchings, such as
  \[ \geq 1 \text{DBpedia-owl:name} \sqsubseteq \text{yago:Person} \]
Case Study 2
Refining mappings by separate-and-conquer rule learning

- **Given:**
  - two ontologies $O_1$ and $O_2$ 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

Dataset from ontology $O_1$

@relation car
@attribute acceleration {low, medium, high} @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, high, medium, medium
low, low, high, high, low
low, low, high, low, medium
medium, high, high, low, low
medium, high, high, medium, medium
low, high, high, medium, high
...

Learn $\downarrow$ rules

$r_{1,1}$: milesPerGallon = medium $\leftarrow$ convenience $\wedge$ Rating = high $\wedge$ acceleration = high

$r_{1,2}$: milesPerGallon = high $\leftarrow$ acceleration = medium $\wedge$ cargoCapacity = low

Dataset from ontology $O_2$

@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, high, high, low, medium
medium, high, high, high, medium
medium, high, high, high, medium
low, high, high, medium, high
...

Learn $\downarrow$ rules

mpg = medium $\leftarrow$ convenience = high $\wedge$ acceleration = high

numberOfExtras = high $\leftarrow$ convenience = high $\wedge$ passengerSpace = high
Case Study 2
Refining mappings by separate-and-conquer rule learning

![Diagram showing the process of mapping refinement]

- **Dataset from ontology $O_1$**
  - @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}

- **Dataset from ontology $O_2$**
  - @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}

**Learn ↓ rules**

- $r_{1,1}$: $\text{milesPerGallon}=\text{medium} \leftarrow \text{convenience-Rating}=\text{high} \land \text{acceleration}=\text{high}$
- $r_{1,2}$: $\text{milesPerGallon}=\text{high} \leftarrow \text{acceleration}=\text{medium} \land \text{cargoCapacity}=\text{low}$
- $\text{mpg}=\text{medium} \leftarrow \text{convenience}=\text{high} \land \text{acceleration}=\text{high}$
- $\text{numberOfExtras}=\text{high} \leftarrow \text{convenience}=\text{high} \land \text{passengerSpace}=\text{high}$
Case Study 2
Refining mappings by separate-and-conquer rule learning

![Diagram showing the process of refining mappings using separate-and-conquer rule learning.]

**Dataset from ontology $O_1$**

- @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}

**Dataset from ontology $O_2$**

- @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}

**Learned Rules**

$r_{1,1}$: milesPerGallon = medium $\leftarrow$ convenience $\wedge$ Rating = high $\wedge$ acceleration = high

$r_{1,2}$: milesPerGallon = high $\leftarrow$ acceleration = high $\wedge$ cargoCapacity = low

$\text{mpg} = \text{medium} \leftarrow \text{convenience} = \text{high} \land \text{acceleration} = \text{high}$

$\text{numberOfExtras} = \text{high} \leftarrow \text{convenience} = \text{high} \land \text{passengerSpace} = \text{high}$
Case Study 2
Refining mappings by separate-and-conquer rule learning

▶ Idea:
  ▶ similar rule sets → mapping candidate

▶ possible similarity measures:

\[
sim_R(R, R') = \frac{\sum_{\sim r, i, j} \left( \frac{tp(r_1, i) + tp(r_2, j)}{|D_1| + |D_2|} \right) \geq \theta}{|D_1| + |D_2|}
\]

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

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 \( \theta \) 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?