A pattern-based ontology matching approach for detecting complex correspondences

Dominique Ritze
Christian Meilicke
Ondřej Šváb-Zamazal
Heiner Stuckenschmidt
Introduction

• Ontology Matching often limited to find simple correspondences (between atomic entities):

  \[
  \text{writtenBy} \equiv \text{hasAuthor}, \ \text{Person} \supseteq \text{Female}
  \]

• Not enough due to heterogeneity

• Need for complex correspondences

• Complex: at least one non-atomic entity
A pattern-based ontology matching approach for detecting complex correspondences

Outline

- Introduction
- Problem Statement
- Complex Correspondence Patterns
- Pattern Detection
- Experimental Results
- Summary & Future Work
A pattern-based ontology matching approach for detecting complex correspondences

Problem Statement

- Semantic heterogeneity
- Different vocabulary, granularity, model styles
- Example:

  - Some work already done (database, machine learning)
A pattern-based ontology matching approach for detecting complex correspondences

**Complex Correspondence Patterns**

- Searched manually examples in OAEI Benchmark & Conference
- Chose four patterns which have been implemented

**CAT:** Class by Attribute Type Pattern

**CAT^{-1}**: Class by Inverse Attribute Type Pattern

**CAV:** Class by Attribute Value Pattern

**PC:** Property Chain Pattern

- CAT and CAT^{-1} are in the patterns library (F. Scharffe)
A pattern-based ontology matching approach for detecting complex correspondences

PositiveReviewedPaper ≡ ∃hasEvaluation. Positive
A pattern-based ontology matching approach for detecting complex correspondences

\[ \text{Person} \cap \exists \text{researchedBy}^{-1}.T \equiv \text{Researcher} \]
A pattern-based ontology matching approach for detecting complex correspondences

SubmittedPaper ≡ ∃submission.{true}
A pattern-based ontology matching approach for detecting complex correspondences

author ≡ hasAuthor o name
A pattern-based ontology matching approach for detecting complex correspondences

Pattern Detection

- Conjunction of conditions
- Combining simple existing techniques
  - Structural methods: hierarchy, disjointness, domain, range
  - Linguistic methods: similarity (Levenshtein), head noun, first part
  - Data type compatibility
- State-of-the-art input alignment required
- Quality depends on quality of input alignment
A pattern-based ontology matching approach for detecting complex correspondences

CAT Example
A pattern-based ontology matching approach for detecting complex correspondences

CAT Example
A pattern-based ontology matching approach for detecting complex correspondences
A pattern-based ontology matching approach for detecting complex correspondences

CAT Example
A pattern-based ontology matching approach for detecting complex correspondences

CAT Example

Accepted_Paper ≡ ∃hasDecision.Acceptance
Experimental Results

- Tested on OAEI Benchmark & Conference and another Conference set
- Thresholds for similarity decisions
- New correspondences found in the second Conference set

<table>
<thead>
<tr>
<th>Threshold</th>
<th>True positives</th>
<th>False positives</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td>Threshold</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAT&amp;CAT^{-1}</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>PC</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>Σ</td>
<td>28</td>
<td>26</td>
</tr>
</tbody>
</table>

- Increased overall number of property corres. by 11%, concept by 3%
Summary

- Need for complex correspondences
- Example for every detected pattern
- One pattern detection presented as example of CAT
- Results showed number of correct/incorrect correspondences
- Difficult to evaluate
- Much harder to find than simple correspondences
A pattern-based ontology matching approach for detecting complex correspondences

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<complexMapping>
    <define>
        <first path="D:\cmt.owl"/>
        <second path="D:\ekaw.owl"/>
        <alignment path="D:\cmt-ekaw.rdf"/>
    </define>
    <load>
        <concept origin="first" id="concept1"/>
        <concept origin="second" id="concept2"/>
        <concept origin="second" id="superclass"/>
    </load>
    <and>
        <isSubclassOf>
            <entity id="concept1"/>
            <entity id="superclass"/>
        </isSubclassOf>
        <isSubclassOf>
            <entity id="concept2"/>
            <entity id="superclass"/>
        </isSubclassOf>
        <SimilarityAbove value="0.8">
            <label>
                <entity id="concept1"/>
            </label>
            <label>
                <entity id="concept2"/>
            </label>
        </SimilarityAbove>
    </and>
</complexMapping>

concept1 \equiv \text{concept2}
Future Work

• We try to develop XML language for pattern detection
  • Finding new types
  • Extensible conditions
  • Available for other users

• Open problem:
  • Evaluation foundation
Thank you!

Questions?
A pattern-based ontology matching approach for detecting complex correspondences

**CAT\(^{-1}\) Example**

- **O\(_1\)** Event
  - domain
  - range
  - organised_by
  - Person
  - Organisation

- **O\(_2\)**
  - similar
  - Organizer

\(\neq, \sqsubseteq\)
A pattern-based ontology matching approach for detecting complex correspondences

CAV Example
A pattern-based ontology matching approach for detecting complex correspondences