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):

  writtenBy ≡ hasAuthor, Person ⊒ Female

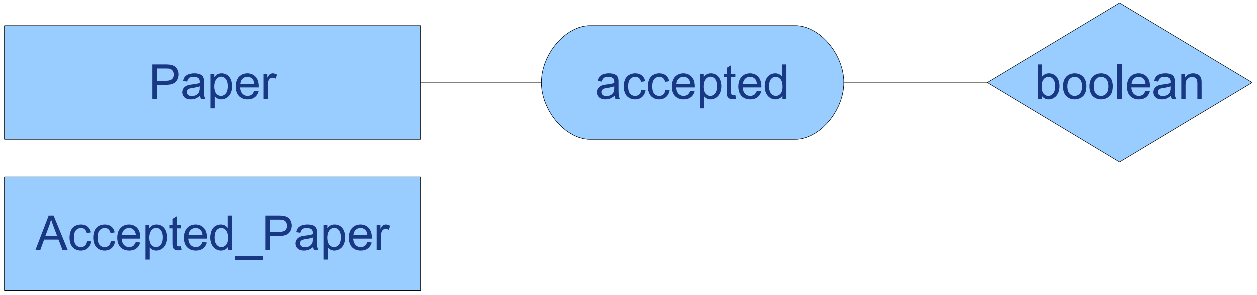
- Not enough due to heterogeneity
- Need for complex correspondences
- Complex: at least one non-atomic entity

Outline

- Introduction
- Problem Statement
- Complex Correspondence Patterns
- Pattern Detection
- Experimental Results
- Summary & Future Work

Problem Statement

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



Paper

accepted

boolean

Accepted_Paper

- Some work already done (database, machine learning)

Complex Correspondence Patterns

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

CAT: Class by Attribute Type Pattern

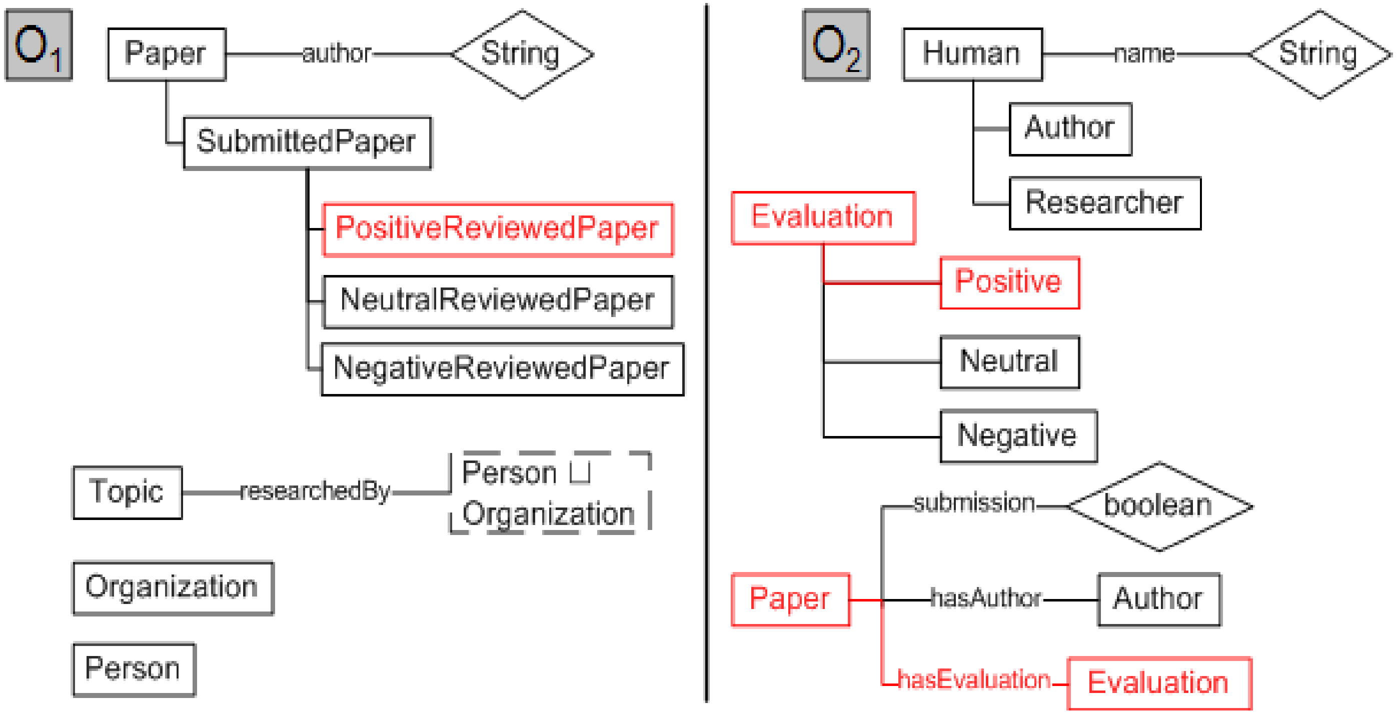
CAT^{-1}: Class by Inverse Attribute Type Pattern

CAV: Class by Attribute Value Pattern

PC: Property Chain Pattern

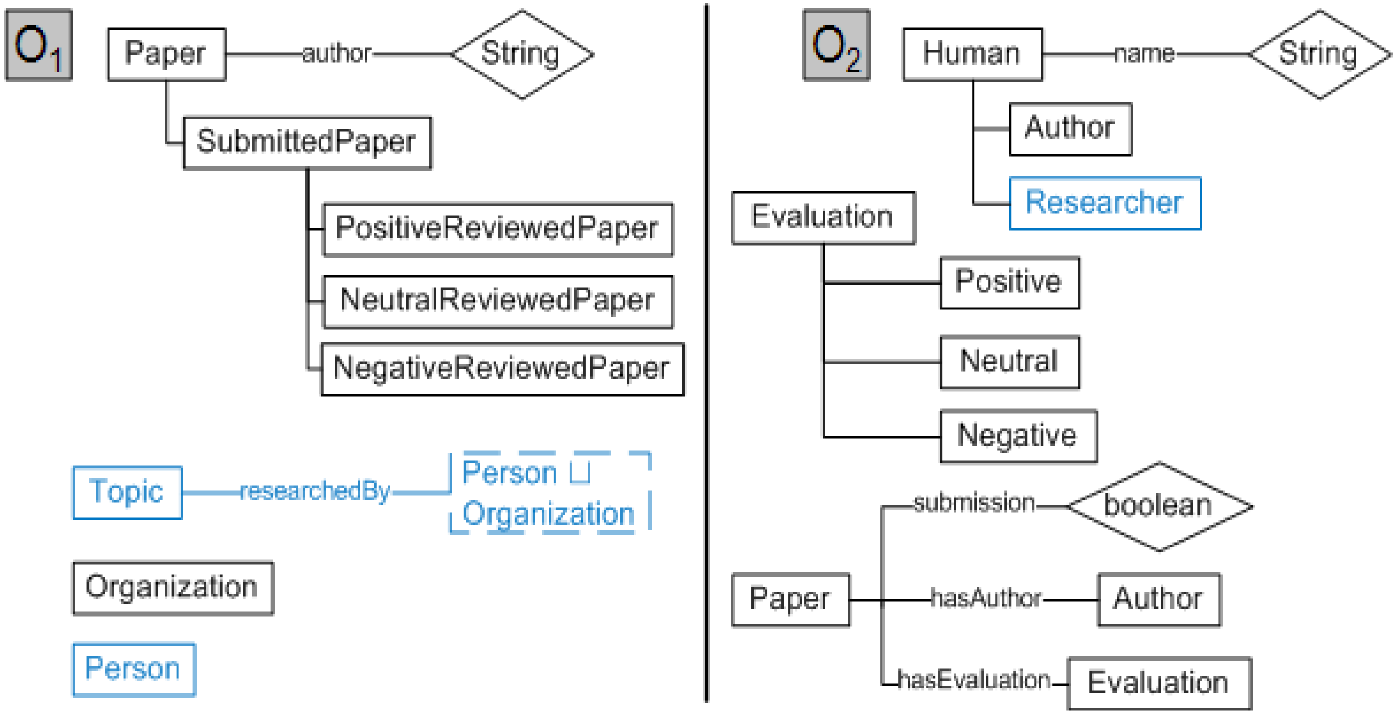
- CAT and CAT^{-1} are in the patterns library (F. Scharffe)

CAT



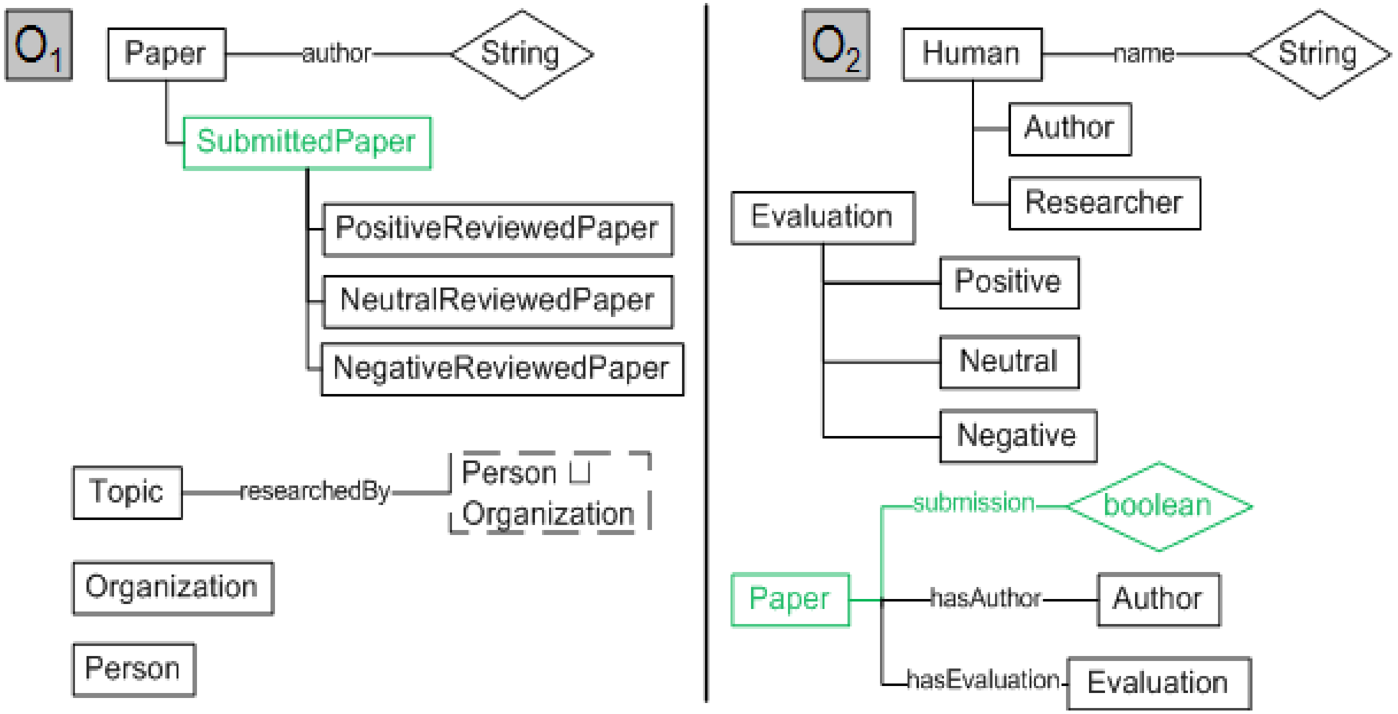
$$PositiveReviewedPaper \equiv \exists hasEvaluation.Positive$$

CAT^{-1}



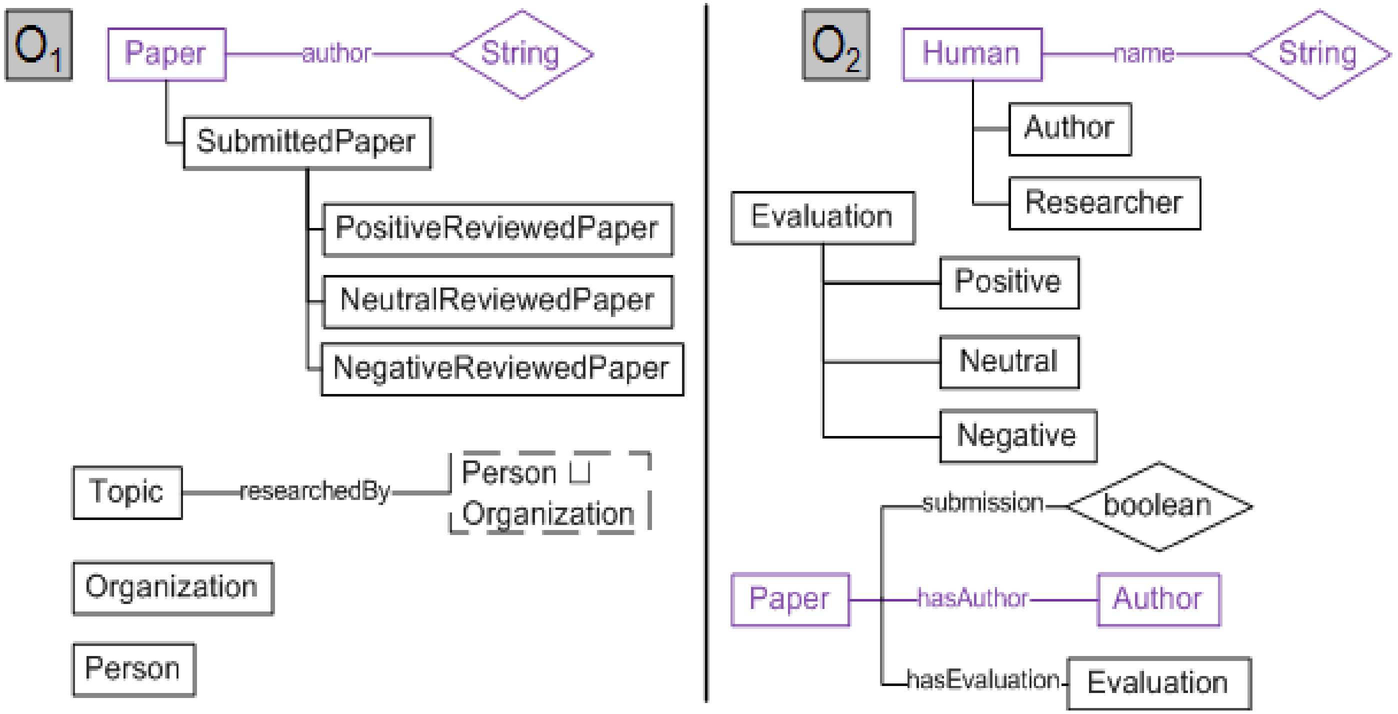
$$\text{Person} \sqcap \ \exists \text{researchedBy}^{-1}.\text{T} \equiv \text{Researcher}$$

CAV



SubmittedPaper ≡ ∃submission.{true}

PC



author ≡ hasAuthor o name

Pattern Detection

- Conjunction of conditions
- Combining simple existing techniques

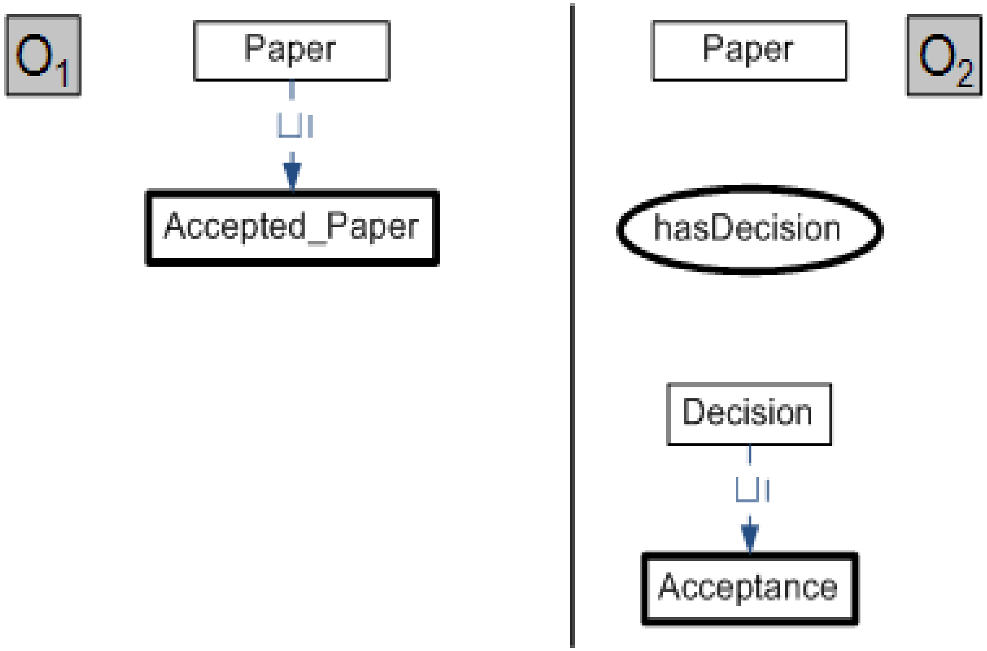
structural methods: hierarchy, disjointness, domain, range

linguistic methods: similarity (Levenshtein), head noun, first part

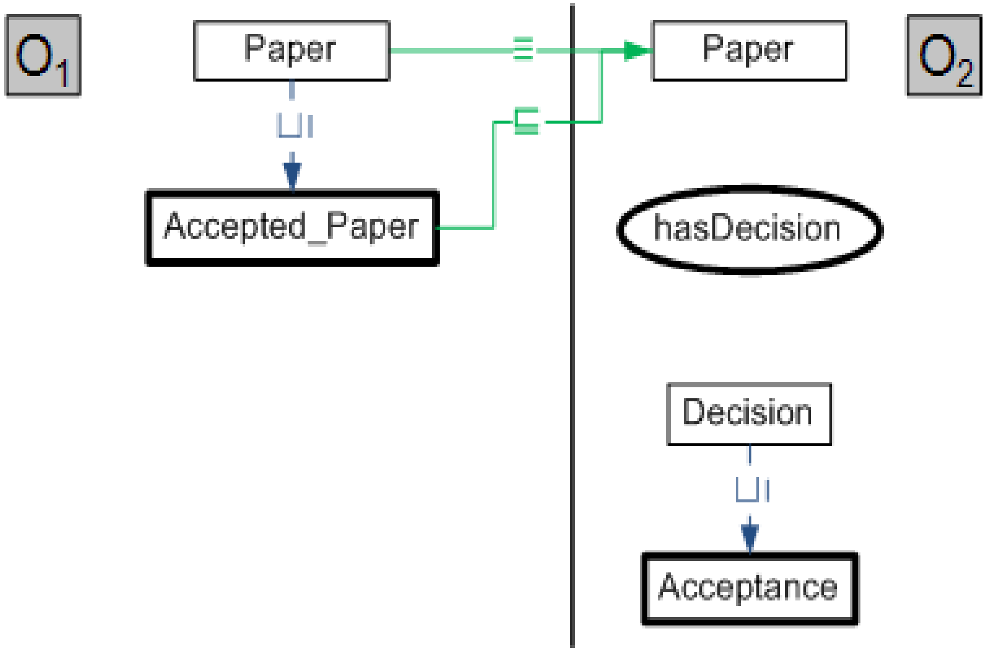
data type compability

- State-of-the-art input alignment required
- Quality depends on quality of input alignment

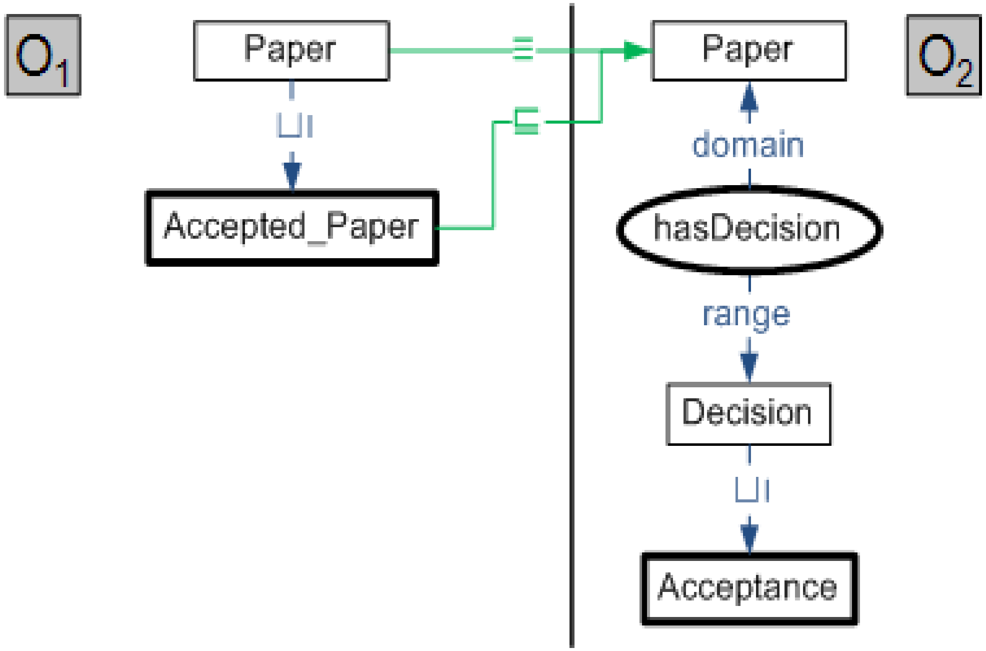
CAT Example


O_1
Paper
$\sqsupseteq$
Accepted_Paper
Paper
O_2
hasDecision
Decision
$\sqsupseteq$
Acceptance

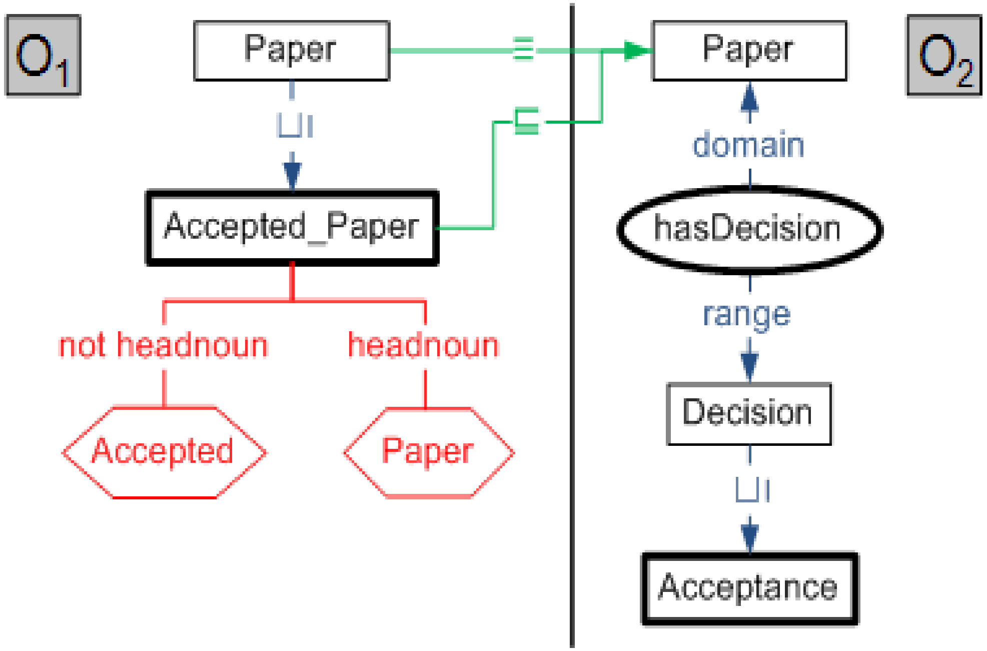
CAT Example


O1
Paper
⊒
Accepted_Paper
≡
⊑
Paper
O2
hasDecision
Decision
⊒
Acceptance

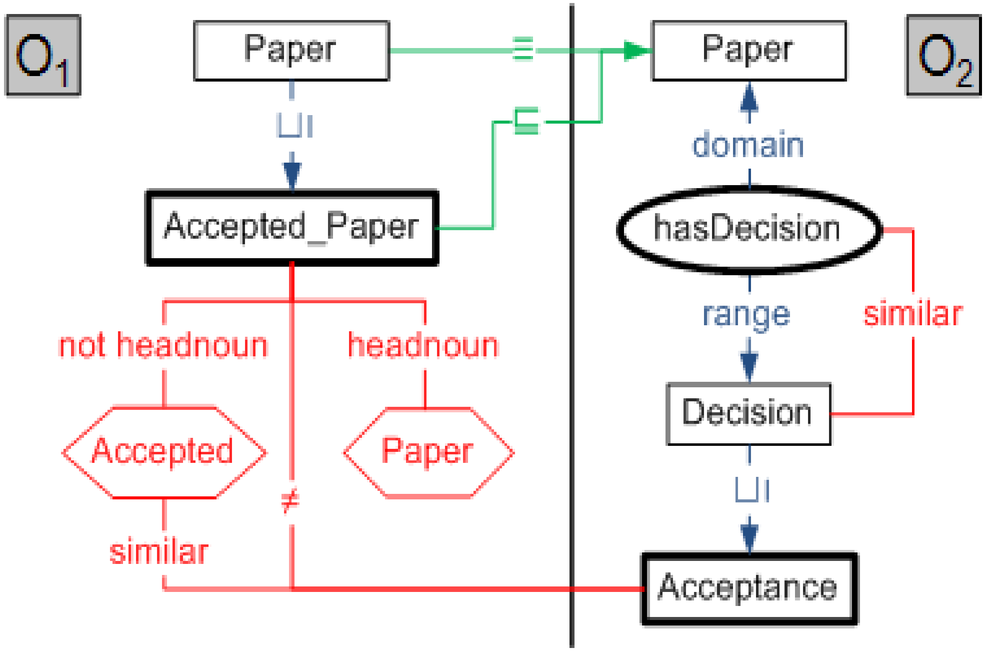
CAT Example


O1
Paper
⊑
Accepted_Paper
≡
⊑
Paper
O2
domain
hasDecision
range
Decision
⊑
Acceptance

CAT Example


O_1
Paper
⊑
Accepted_Paper
not headnoun
headnoun
Accepted
Paper
≡
⊑
Paper
O_2
domain
hasDecision
range
Decision
⊑
Acceptance

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

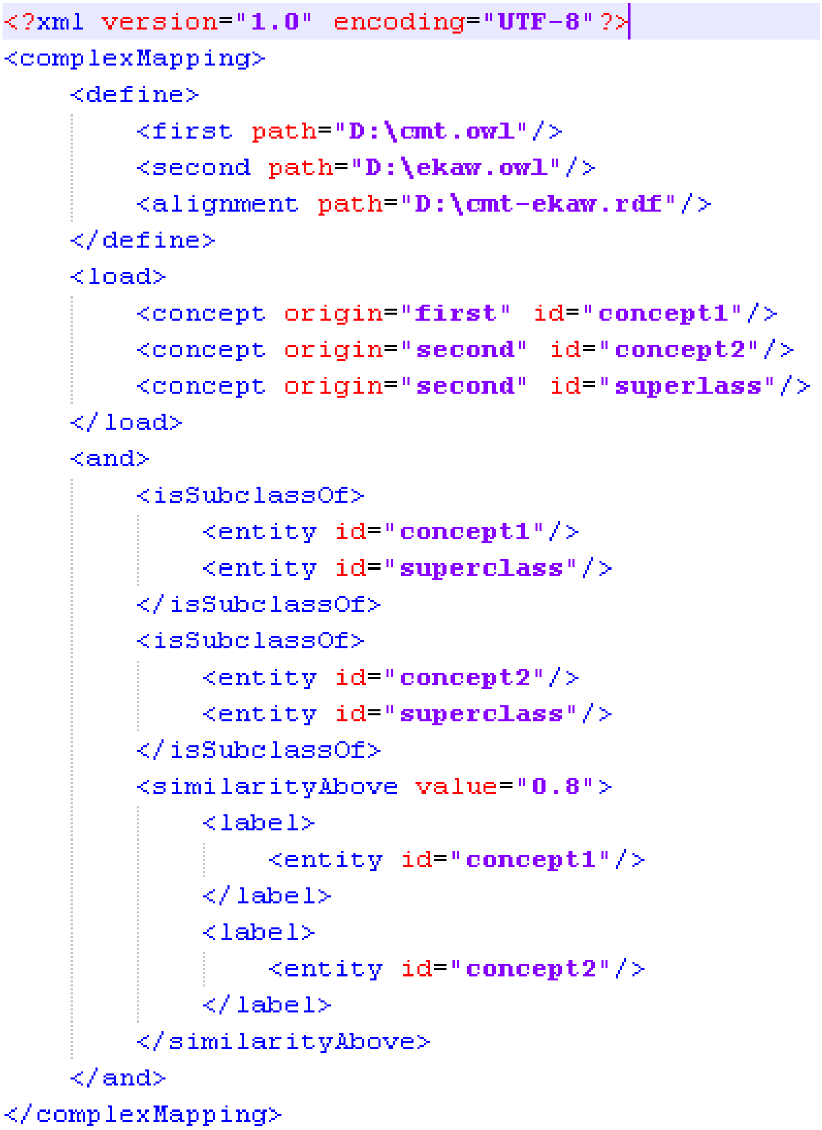
	True positives				False positives			
Threshold	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9
CAT&CAT^{-1}	10	8	7	0	24	14	11	2
PC	18	18	18	18	21	16	14	8
Σ	28	26	25	18	45	30	25	10

- 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 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="superlass"/>
    </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 ≡ concept2

Future Work

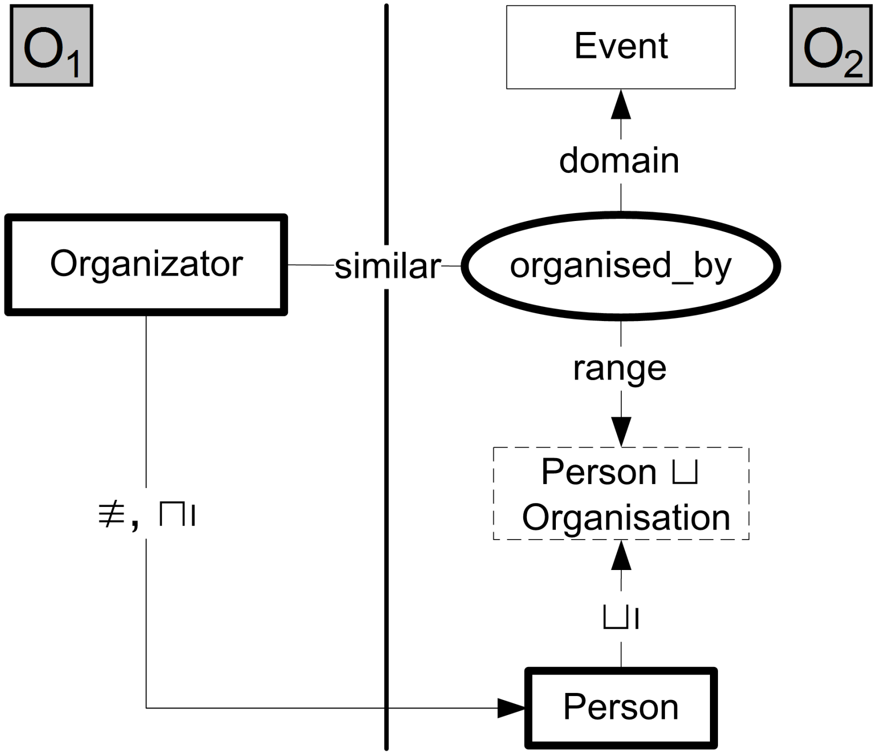
- We try to develop XML language for pattern detection
  - Finding new types
  - Extensible conditions
  - Available for other users

- Open problem:
  - Evaluation foundation

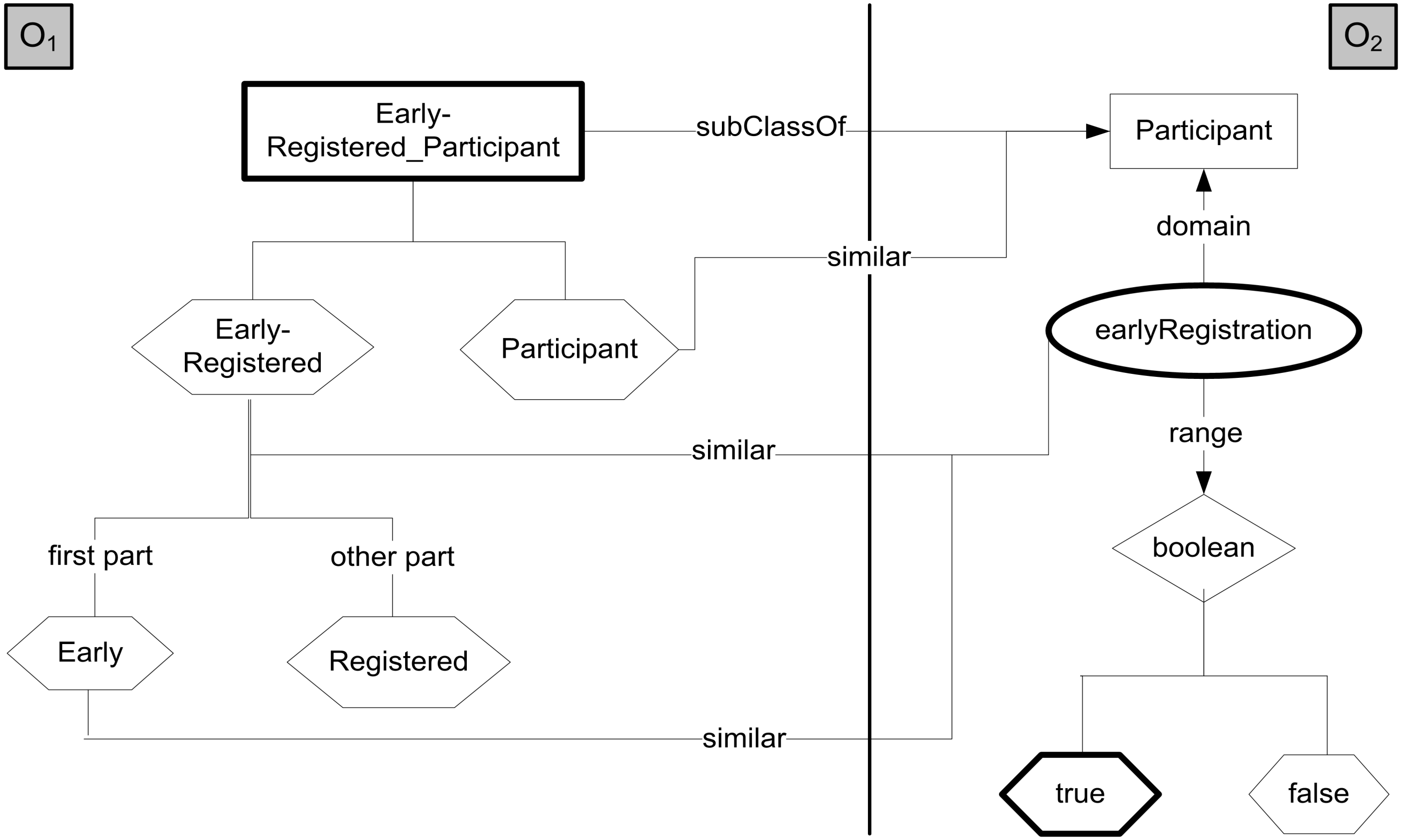
Thank you!

Questions?

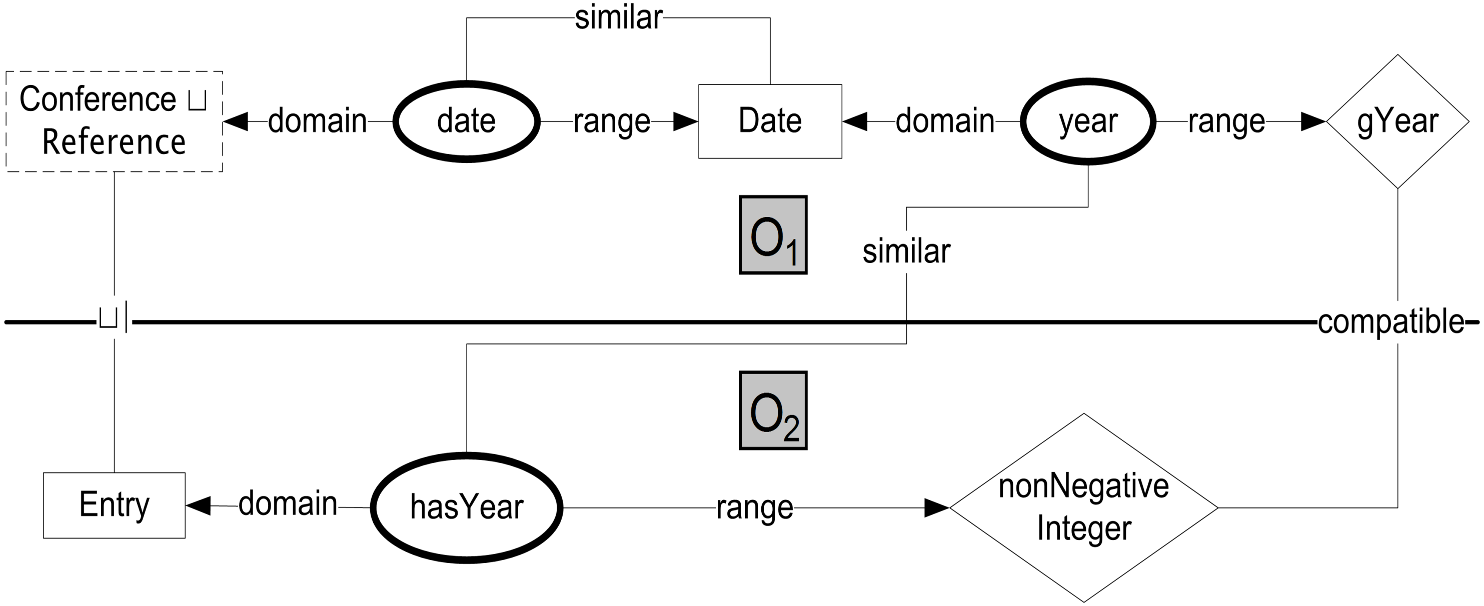
CAT^{-1} Example



CAV Example


O_1
O_2
Early-Registered_Participant
subClassOf
Participant
domain
similar
Early-Registered
Participant
earlyRegistration
similar
range
first part
other part
boolean
Early
Registered
similar
true
false

PC Example


similar
Conference ⊔ Reference
domain
date
range
Date
domain
year
range
gYear
O1
similar
⊔|
compatible
O2
Entry
domain
hasYear
range
nonNegative Integer