Tutorial on Ontology Matching

Pavel Shvaiko  Jérôme Euzenat

Trento, Italy
pavel@dit.unitn.it

Monbonnot, France
Jerome.Euzenat@inrialpes.fr

December 18, 2006
Goals of the tutorial

- Illustrate the role of ontology matching
- Provide an overview of basic matching techniques
- Demonstrate the use of basic matching techniques in state of the art systems
- Motivate future research
Outline

Matching problem

Classification

Basic techniques

Matching process

Systems

Conclusions
Outline

Matching problem
Classification
Basic techniques
Matching process
Systems
Conclusions
Matching operation takes as input ontologies, each consisting of a set of discrete entities (e.g., tables, XML elements, classes, properties) and determines as output the relationships (e.g., equivalence, subsumption) holding between these entities.
Motivation: two XML schemas
Motivation: two XML schemas
Motivation: two ontologies

Product
- DVD
- Book

Monograph
- Essay
  - Litterary critics
  - Politics
    - Biography
  - Autobiography
  - Literature
Motivation: two ontologies

Product
- price
- title
- doi
- creator
- topic

DVD
- author

Book

CD

Monograph
- isbn
- author
- title

Essay

Litterary critics

Politics

Biography

subject

Autobiography

Literature

Classification Basic techniques Matching process Systems Conclusions
Motivation: two ontologies

Product
- price
- title
- doi
- creator
- topic
- DVD
- Book
- CD
- author

Person

integer

string

uri

Monograph
- isbn
- author
- title
- Essay
- Litterary critics
- Politics
- Biography
- subject
- Autobiography
- Literature

Tutorial on Ontology Matching at SWAP-2006, Pisa, Italy
Motivation: two ontologies

- **Product**
  - price
  - title
  - doi
  - creator
  - topic
  - DVD
  - Book
  - CD
  - author

- **Person**
  - integer
  - string
  - uri

- **Monograph**
  - isbn
  - author
  - title
  - Essay
  - Litterary critics
  - Politics
  - Biography
  - subject
  - Literature
  - Autobiography
  - Writer
  - Human

- Example:
  - *Bertrand Russell: My life*
  - *Albert Camus: La chute*
Motivation: two ontologies

- Product
  - price
  - title
  - doi
  - creator
  - topic
- DVD
- Book
  - author
- CD

- Monograph
  - isbn
  - author
  - title
- Essay
- Litterary critics
- Politics
- Biography
- Writer

Bertrand Russell: My life
Albert Camus: La chute
Motivation: two ontologies
Matching problem Classification Basic techniques Matching process Systems Conclusions

Schema matching vs. ontology matching: differences

- Schemas often do not provide explicit semantics for their data
  - Relational schemas provide no generalization
### Schema matching vs. ontology matching: differences

- Schemas often do not provide explicit semantics for their data
  - Relational schemas provide no generalization
- Ontologies are logical systems that constrain the meaning
  - Ontology definitions as a set of logical axioms
Schema matching vs. ontology matching: commonalities

- Schemas and ontologies provide a vocabulary of terms that describes a domain of interest
Schema matching vs. ontology matching: commonalities

- Schemas and ontologies provide a vocabulary of terms that describes a domain of interest
- Schemas and ontologies constrain the meaning of terms used in the vocabulary
Schema matching vs. ontology matching: commonalities

- Schemas and ontologies provide a vocabulary of terms that describes a domain of interest
- Schemas and ontologies constrain the meaning of terms used in the vocabulary

Techniques developed for both problems are of a mutual benefit
Heterogeneity between ontologies can occur when

- different languages are used
- different terminologies are used
- different modeling is used
- ...
Heterogeneity between ontologies can occur when

- different languages are used
- different terminologies are used
- different modeling is used
- ...
Scope

- Reducing heterogeneity can be performed in 2 steps
  - Match, thereby determine the alignment
Scope

Reducing heterogeneity can be performed in 2 steps

- Match, thereby determine the alignment
- Process the alignment (merging, transforming, etc.)
Reducing heterogeneity can be performed in 2 steps
- Match, thereby determine the alignment
- Process the alignment (merging, transforming, etc.)
Scope

- Reducing heterogeneity can be performed in 2 steps
  - Match, thereby determine the alignment
  - Process the alignment (merging, transforming, etc.)
- When do we match?
  - Design time
  - Run time
Scope

- Reducing heterogeneity can be performed in 2 steps
  - Match, thereby determine the alignment
  - Process the alignment (merging, transforming, etc.)
- When do we match?
  - Design time
  - Run time
Correspondence

Definition (Correspondence)

Given two ontologies $O$ and $O'$, a correspondence $M$ between $O$ and $O'$ is a 5-uple: $⟨id, e, e', R, n⟩$ such that:

- $id$ is a unique identifier of the correspondence
Definition (Correspondence)

Given two ontologies $O$ and $O'$, a correspondence $M$ between $O$ and $O'$ is a 5-uple: $\langle id, e, e', R, n \rangle$ such that:

- $id$ is a unique identifier of the correspondence
- $e$ and $e'$ are entities of $O$ and $O'$ (e.g., XML elements, classes)
Correspondence

Definition (Correspondence)

Given two ontologies $O$ and $O'$, a correspondence $M$ between $O$ and $O'$ is a 5-uple: $\langle id, e, e', R, n \rangle$ such that:

- $id$ is a unique identifier of the correspondence
- $e$ and $e'$ are entities of $O$ and $O'$ (e.g., XML elements, classes)
- $R$ is a relation (e.g., equivalence ($=$), more general ($\sqsupseteq$), disjointness ($\perp$))
Correspondence

Definition (Correspondence)

Given two ontologies $O$ and $O'$, a correspondence $M$ between $O$ and $O'$ is a 5-uple: $\langle id, e, e', R, n \rangle$ such that:

- $id$ is a unique identifier of the correspondence
- $e$ and $e'$ are entities of $O$ and $O'$ (e.g., XML elements, classes)
- $R$ is a relation (e.g., equivalence ($=$), more general ($\sqsupseteq$), disjointness ($\bot$))
- $n$ is a confidence measure in some mathematical structure (typically in the [0,1] range)
Alignment

Definition (Alignment)

Given two ontologies $O$ and $O'$, an alignment $(A)$ between $O$ and $O'$:

- is a set of correspondences on $O$ and $O'$
Definition (Alignment)

Given two ontologies $O$ and $O'$, an alignment $(A)$ between $O$ and $O'$:

- is a set of correspondences on $O$ and $O'$
- with some cardinality: 1-1, 1-*, etc.
Definition (Alignment)

Given two ontologies $O$ and $O'$, an alignment ($A$) between $O$ and $O'$:

- is a set of correspondences on $O$ and $O'$
- with some cardinality: 1-1, 1-*, etc.
- some additional metadata (method, date, properties, etc.)
Matching process
Matching process

Diagram:

O → matching → O'
Matching process
Matching process

Diagram:

- O
- A
- O'

Matching

A'
Matching process

- $O$
- $A$
- $O'$
- $A'$

Parameters:

Matching:

Tutorial on Ontology Matching at SWAP-2006, Pisa, Italy
Matching process

- Parameters
- Resources

Matching process:
- \( O \)
- \( A \)
- \( O' \)
- \( A' \)

Diagram:
- \( O \) to \( A \) through parameters
- \( O' \) to \( A' \) through resources
- Matching process between \( A \) and \( A' \)
### Application domains

#### Traditional
- Ontology evolution
- Schema integration
- Catalog integration
- Data integration
Application domains

▶ **Traditional**
  ▶ Ontology evolution
  ▶ Schema integration
  ▶ Catalog integration
  ▶ Data integration

▶ **Emergent**
  ▶ P2P information sharing
  ▶ Agent communication
  ▶ Web service composition
  ▶ Query answering on the web
Application: catalog integration (simplified)
Application: catalog integration (simplified)

- **O** → **Matcher** → **O'**
- **DB** → **Matcher** → **DBPortal**
Application: catalog integration (simplified)
Application: catalog integration (simplified)
Application: catalog integration (simplified)
Applications: P2P information sharing
Applications: P2P information sharing
Applications: P2P information sharing
Applications: P2P information sharing

Diagram:
- Peer 1 (peer1) connects to A (Matcher) which connects to Generator.
- Generator connects to Peer 2 (peer2).
- A has a mediator between peer1 and peer2.
Applications: P2P information sharing

Diagram:

- **O** → **Matcher** → **A** → **Generator** → **mediator** → **query** → **peer1**
- **O'** → **Matcher** → **A** → **Generator** → **mediator** → **query** → **peer2**
Applications: P2P information sharing
## Applications: summary

<table>
<thead>
<tr>
<th>Application</th>
<th>instances</th>
<th>run time</th>
<th>automatic</th>
<th>correct</th>
<th>complete</th>
<th>operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ontology evolution</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
<td></td>
<td>transformation</td>
</tr>
<tr>
<td>Schema integration</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
<td></td>
<td>merging</td>
</tr>
<tr>
<td>Catalog integration</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
<td></td>
<td>data translation</td>
</tr>
<tr>
<td>Data integration</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
<td></td>
<td>query answering</td>
</tr>
<tr>
<td>P2P information sharing</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>query answering</td>
</tr>
<tr>
<td>Web service composition</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td>data mediation</td>
</tr>
<tr>
<td>Multi agent communication</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td>√</td>
<td>data translation</td>
</tr>
<tr>
<td>Query answering</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td>query reformulation</td>
</tr>
</tbody>
</table>
Outline

Matching problem

Classification

Basic techniques

Matching process

Systems

Conclusions
Matching dimensions

- Input dimensions
  - Underlying models (e.g., XML, OWL)
  - Schema-level vs. Instance-level
Matching dimensions

▶ Input dimensions
  ▶ Underlying models (e.g., XML, OWL)
  ▶ Schema-level vs. Instance-level

▶ Process dimensions
  ▶ Approximate vs. Exact
  ▶ Interpretation of the input
Matching dimensions

- **Input dimensions**
  - Underlying models (e.g., XML, OWL)
  - Schema-level vs. Instance-level

- **Process dimensions**
  - Approximate vs. Exact
  - Interpretation of the input

- **Output dimensions**
  - Cardinality (e.g., 1-1, 1-*)
  - Equivalence vs. Diverse relations (e.g., subsumption)
  - Graded vs. Absolute confidence
Matching dimensions

- **Input dimensions**
  - Underlying models (e.g., XML, OWL)
  - **Schema-level** vs. **Instance-level**

- **Process dimensions**
  - Approximate vs. Exact
  - Interpretation of the input

- **Output dimensions**
  - Cardinality (e.g., 1-1, 1-*)
  - Equivalence vs. Diverse relations (e.g., subsumption)
  - Graded vs. Absolute confidence
Three layers

- The upper layer
  - Granularity of match
  - Interpretation of the input information
Three layers

▶ The upper layer
  ▶ Granularity of match
  ▶ Interpretation of the input information

▶ The middle layer represents classes of elementary (basic) matching techniques
Three layers

- **The upper layer**
  - Granularity of match
  - Interpretation of the input information

- **The middle layer** represents classes of elementary (basic) matching techniques

- **The lower layer** is based on the kind of input which is used by elementary matching techniques
<table>
<thead>
<tr>
<th>Classification</th>
<th>Basic techniques</th>
<th>Matching process</th>
<th>Systems</th>
<th>Conclusions</th>
</tr>
</thead>
</table>

**Classification of schema-based techniques (simplified)**
Classification of schema-based techniques (simplified)
Classification of schema-based techniques (simplified)

Element-level
- Syntactic
  - String-based name, description similarity
  - Language-based tokenization, elimination
  - Linguistic resources lexicons, thesauri
- Constraint-based
  - type similarity, key properties
- Upper, domain specific formal ontologies
  - DOLCE, FMA

Structure-level
- Syntactic
  - Graph-based
    - graph homomorphism children, leaves
- External
  - Taxonomy-based
    - taxonomy structure
- Semantics
  - Repository of structures
    - structure metadata
  - Model-based
    - SAT solvers, DL reasoners
Classification of schema-based techniques (simplified)

- **Element-level**
  - Syntactic
  - External

- **Structure-level**
  - Syntactic
  - External
  - Semantics

**String-based**
- name, description similarity

**Language-based**
- tokenization, elimination

**Linguistic resources**
- lexicons, thesauri

**Constraint-based**
- type similarity, key properties

**Upper, domain specific formal ontologies**
- DOLCE, FMA

**Graph-based**
- graph homomorphism
- children, leaves

**Taxonomy-based**
- taxonomy structure

**Repository of structures**
- structure metadata

**Model-based**
- SAT solvers, DL reasoners

**Linguistic**
- Terminological

**Internal**
- Structural

**Relational**
- Semantic

Tutorial on Ontology Matching at SWAP-2006, Pisa, Italy
Classification of schema-based techniques (simplified)
Outline

Matching problem
Classification
Basic techniques
Matching process
Systems
Conclusions
Element-level techniques: String-based

- Prefix
  - takes as input two strings and checks whether the first string starts with the second one
  - `net = network`; but also `hot = hotel`

(e.g., COMA, SF, S-Match, OLA)
Element-level techniques: String-based

► Prefix
  ► takes as input two strings and checks whether the first string starts with the second one
  ► net = network; but also hot = hotel

► Suffix
  ► takes as input two strings and checks whether the first string ends with the second one
  ► ID = PID; but also word = sword

(e.g., COMA, SF, S-Match, OLA)
Element-level techniques: String-based

- **Edit distance**
  - takes as input two strings and calculates the number of edition operations, (e.g., insertions, deletions, substitutions) of characters required to transform one string into another, normalized by length of the maximum string
  - \( \text{EditDistance}(\text{NKN}, \text{Nikon}) = 0.4 \)

(e.g., S-Match, OLA, Anchor-Prompt)
Element-level techniques: Language-based

- **Tokenization**
  - parses names into tokens by recognizing punctuation, cases
  - Hands-Free_Kits → ⟨ hands, free, kits ⟩

(e.g., COMA, Cupid, S-Match, OLA)
Element-level techniques: Language-based

- **Tokenization**
  - parses names into tokens by recognizing punctuation, cases
  - Hands-Free_ Kits $\rightarrow$ ⟨ hands, free, kits ⟩

- **Lemmatization**
  - analyses morphologically tokens in order to find all their possible basic forms
  - Kits $\rightarrow$ Kit

(e.g., COMA, Cupid, S-Match, OLA)
Element-level techniques: Language-based

- **Elimination**
  - discards “empty” tokens that are articles, prepositions, conjunctions . . .
  - a, the, by, type of, their, from

(e.g., Cupid, S-Match)
Element-level techniques: Linguistic resources

- Sense-based: WordNet
  - A ⊑ B if A is a hyponym or meronym of B
    - Brand ⊑ Name
  - A ⊒ B if A is a hypernym or holonym of B
    - Europe ⊒ Greece
  - A = B if they are synonyms
    - Quantity = Amount
  - A ⊥ B if they are antonyms or the siblings in the part of hierarchy
    - Microprocessors ⊥ PC Board

(e.g., Artemis, CtxMatch, S-Match)
Element-level techniques: Linguistic resources

- Gloss-based: WordNet gloss comparison
  - The number of the same words occurring in both input glosses increases the similarity value. The equivalence relation is returned if the resulting similarity value exceeds a given threshold
  - Maltese dog is a breed of toy dogs having a long straight silky white coat
  - Afghan hound is a tall graceful breed of hound with a long silky coat

(e.g., S-Match)
Structure-level techniques: Taxonomy-based

Ontologies are viewed as graph-like structures containing terms and their inter-relationships.

- **Bounded path matching**
  - These take two paths with links between classes defined by the hierarchical relations, compare terms and their positions along these paths, and identify similar terms

(e.g., Anchor-Prompt, NOM, QOM)
Structure-level techniques: Taxonomy-based

Ontologies are viewed as graph-like structures containing terms and their inter-relationships.

- **Bounded path matching**
  - These take two paths with links between classes defined by the hierarchical relations, compare terms and their positions along these paths, and identify similar terms

- **Super(sub)-concepts rules**
  - If super-concepts are the same, the actual concepts are similar to each other

(e.g., Anchor-Prompt, NOM, QOM)
Structure-level techniques: Tree-based

- **Children**
  - Two non-leaf schema elements are structurally similar if their immediate children sets are highly similar

(e.g., Cupid, COMA)
Structure-level techniques: Tree-based

- **Children**
  - Two non-leaf schema elements are structurally similar if their immediate children sets are highly similar

- **Leaves**
  - Two non-leaf schema elements are structurally similar if their leaf sets are highly similar, even if their immediate children are not

(e.g., Cupid, COMA)
Structure-level techniques: Tree-based

(e.g., Cupid, COMA)
Structure-level techniques: Tree-based

(e.g., Cupid, COMA)
Structure-level techniques: Tree-based

(e.g., Cupid, COMA)
Structure-level techniques: Model-based

- Propositional satisfiability (SAT)

\[ Axioms \rightarrow rel(\text{context}_1, \text{context}_2) \]

(e.g., CtxMatch, S-Match)
Structure-level techniques: Model-based

- Propositional satisfiability (SAT)

\[ \text{Axioms} \rightarrow \text{rel}(\text{context}_1, \text{context}_2) \]

(e.g., CtxMatch, S-Match)
Structure-level techniques: Model-based

Propositional satisfiability (SAT)

Axioms $\rightarrow \text{rel}(\text{context}_1, \text{context}_2)$

(e.g., CtxMatch, S-Match)
Structure-level techniques: Model-based

Description logics (DL)-based

\[ \text{micro-company} = \text{company} \quad \sqsubseteq \leq_5 \text{employee} \]

\[ \text{SME} = \text{firm} \quad \sqsubseteq \leq_{10} \text{associate} \]
Structure-level techniques: Model-based

Description logics (DL)-based

- micro-company = company
  - □ ≤₅ employee

- SME = firm
  - □ ≤₁₀ associate
Structure-level techniques: Model-based

Description logics (DL)-based

micro-company = company ⊓ ≤₅ employee

SME = firm ⊓ ≤₁₀ associate

company = firm ; associate ⊑ employee
Structure-level techniques: Model-based

Description logics (DL)-based

\[ \text{micro-company} = \text{company} \quad \sqsubseteq \quad \leq 5 \text{ employee} \]

\[ \text{SME} = \text{firm} \quad \sqsupseteq \quad \leq 10 \text{ associate} \]

\[ \text{company} = \text{firm} ; \text{associate} \sqsubseteq \text{employee} \]

\[ \text{micro-company} \sqsubseteq \text{SME} \]
Outline

Matching problem

Classification

Basic techniques

Matching process

Systems

Conclusions
Sequential composition
Parallel composition

- **parameters**
- **resources**
- **parameters'**
- **resources'**

Matching problem
Classification
Basic techniques
Matching process
Systems
Conclusions
Selecting the final alignment

- Ranking strategies
  - Thresholds
  - MaxDelta
Selecting the final alignment

- Ranking strategies
  - Thresholds
  - MaxDelta
- Cardinalities
  - 1-1, 1-*, *-*
Selecting the final alignment

- Ranking strategies
  - Thresholds
  - MaxDelta

- Cardinalities
  - 1-1, 1-*, *-*

- Optimization
  - stable marriage
  - maximal weight match
Selecting the final alignment

- **Ranking strategies**
  - Thresholds
  - MaxDelta

- **Cardinalities**
  - 1-1, 1-*, *-*

- **Optimization**
  - stable marriage
  - maximal weight match

- **Directionality**
  - \( O \rightarrow O', \ O' \rightarrow O \) (SmallLarge, LargeSmall)
  - \( O \rightarrow O' \) and \( O' \rightarrow O \) (Both)
<table>
<thead>
<tr>
<th>Outline</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching problem</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic techniques</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Matching process</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conclusions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
State of the art systems

~50 matching systems exist, ... we consider some of them

- **Cupid** (U. of Washington, Microsoft Corporation and U. of Leipzig)
- **Falcon-AO** (China Southwest U.)
- **OLA** (INRIA Rhône-Alpes and U. de Montréal)
- **S-Match** (U. of Trento)
- ...
Cupid

- Schema-based
- Computes similarity coefficients in the [0 1] range
- Performs linguistic and structure matching
- Sequential system
Cupid architecture
OLA

- Schema- and Instance-based
- Computes dissimilarities + extracts alignments (equivalences in the [0 1] range)
- Based on terminological (including linguistic) and structural (internal and relational) distances
- Neither sequential nor parallel
OLA architecture
Falcon-OA architecture
S-Match

- Schema-based
- Computes equivalence (=), more general (⊇), less general (⊆), disjointness (⊥)
- Analyzes the meaning (concepts, not labels) which is codified in the elements and the structures of ontologies
- Sequential system with a composition at the element level
S-Match architecture
Outline

Matching problem

Classification

Basic techniques

Matching process

Systems

Conclusions
Summary

- We have discussed the ontology matching problem and its application domains.
- We have provided classificatory elements for approaching ontology matching techniques.
- We have presented a number of basic matching techniques as well as different strategies for building the matching process.
- We have reviewed some existing matching systems.
Uses of classification

- It provides a common conceptual basis, and hence, can be used for comparing (analytically) different existing ontology matching systems.
- It can help in designing a new matching system, or an elementary matcher, taking advantages of state of the art solutions.
- It can help in designing systematic benchmarks, e.g., by discarding features one by one from ontologies, namely, what class of basic techniques deals with what feature.
Challenges

▶ Missing background knowledge
▶ Performance of systems
▶ Interactive approaches
▶ Explanations of matching
▶ Social aspects of ontology matching
▶ Large-scale evaluation
▶ Infrastructures
▶ ...
Acknowledgments

We thank all the participants of the Heterogeneity workpackage of the Knowledge Web network of excellence

Matching problem
Classification
Basic techniques
Matching process
Systems
Conclusions

...coming up soon
Thank you for your attention and interest!
Questions?

pavel@dit.unitn.it
Jerome.Euzenat@inrialpes.fr

http://www.ontologymatching.org