Ontology matching tutorial

Jérôme Euzenat  
Pavel Shvaiko

INRIA & LIG  
Montbonnot Saint-Martin, France  
Jerome.Euzenat@inrialpes.fr

Informatica Trentina SPA  
Trento, Italy  
pavel@dit.unitn.it

June 28, 2009

Problem  Applications  Basic techniques  Process  Conclusions

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

1. The ontology matching problem
2. Applications
3. Basic techniques
4. Matching process
5. Conclusions

Semantic webs
Being serious about the semantic web

- It is not one guy’s ontology
- It is not several guys’ common ontology
- It is many guys and girls’ many ontologies
- So it is a mess, but a meaningful mess.

Living with heterogeneity

The semantic web will be:
- huge;
- dynamic;
- heterogeneous.

These are not bugs, they are features.
We must learn to live with them.

Heterogeneity problem

Resources being expressed in different ways must be reconciled before being used.
Mismatch between formalized knowledge can occur when:
- different languages are used;
- different terminologies are used;
- different modelling is used.

I have a plan for you

Reconciliation can be performed in 2 steps
- Match, thereby determines the alignment
- Generate a processor (for merging, transforming, etc.)

Matching can be achieved at run time or at design time.
Matching process

Motivation: two ontologies

Transformation and mediation

SELECT x.doi
WHERE x : Book
AND x.author = "Bertrand Russell"
AND x.topic = "Bertrand Russell"

SELECT x.isbn
WHERE x : Autobiography
AND x.author = "Bertrand Russell"

x.doi=http://dx.doi.org/10.1080/041522862X

x.isbn=041522862X
Definition (Correspondence)
Given two ontologies \( o \) and \( o' \), a correspondence between \( o \) and \( o' \) is a 5-uple: \( \langle id, e, e', r, n \rangle \) such that:
▶ \( id \) is an 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 (\( \sqsubseteq \)), disjointness (\( \perp \))
▶ \( n \) is a confidence measure in some mathematical structure (typically in the [0 1] range)

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 additional metadata (multiplicity: 1-1, 1-*, method, date, properties, etc.)
Application: ontology evolution

Application: Catalog integration

Applications: P2P information sharing

Applications: Peer-to-peer and emergent semantics
Applications: Web service composition

![Diagram: Mapper, Matcher, Generator, mediator, service1, service2, input, output]

Applications: Agent communication

![Diagram: Mapper, Matcher, Generator, Translator, message, axioms, input, output]

On what basis can we match?

- **Content:** relying on what is inside the ontology
  - **Name, comments, alternate names, names of related entities:** NLP, IR, etc.
  - **Internal structure:** constraints on relations, typing
  - **External structure:** relations between entities: Data mining, Discrete mathematics
  - **Extension:** Statistics, data analysis, data mining, machine learning
  - **Semantics** (models): Reasoning techniques
- **Context:** the relations of the ontology with the outside
  - **Annotated resources:**
  - **The web**
  - **External ontologies:** dbpedia, etc.
  - **External resources:** wordnet, etc.
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
- EditDistance(NKN,Nikon) = NKN/5 = 2/5 = 0.4
- EditDistance(editeur,editor) = editeur/7 = 3/7 = 0.43

Element-level techniques: Linguistic resources

Sense-based: WordNet hierarchy distance

Context-based matching
- Using the ontologies on the web as context;
- Composing the relations obtained through these ontologies

1. Harvest ontologies on the web;
2. Select those which are related to the ontologies to match;
3. Match them to the found ontologies;
4. Compose the relations between entities through the intermediate ontologies;
5. Aggregate the obtained results (if desired).

At each step there is some latitude.

Example: Scarlet
Extensional techniques

$\epsilon : C \rightarrow E$

$E$ can be a set of instances, a set of documents which are indexed by concepts, a set of items, e.g., people, which use these concepts.

Two cases:
- $E$ is common to both ontologies;
- $E$ depends on the ontology. This can be reduced to the former case by identification or record linkage techniques.

Techniques:
- statistical and machine learning techniques infer and compare the characteristics of populations;
- set-theoretic techniques compare the extensions;

Structure-level techniques: Model-based

Description logics (DL)-based

- $\text{micro-company} = \text{company}$
- $\text{SME} = \text{firm}$
- $\text{associate} \sqsubseteq \text{employee}$
- $\text{micro-company} \sqsubseteq \text{SME}$
Matching process

Basic matchers provide candidate correspondences, most of the systems use several such matchers and further combine and filter their results.

Sequential composition

Parallel composition

Data integration as sequential composition
Many algorithms are based on similarity or distance computation. A number of operations can be based on similarity/distance matrices. There are many different ways to aggregate matcher results, usually depending on confidence/similarity:

- **Triangular norms** (min, weighted products) useful for selecting only the best results;
- **Multidimentional distances** (Euclidean distance, weighted sum) useful for taking into account all dimensions;
- **Fuzzy aggregation** (min, weighted average) useful for aggregating competing algorithms and averaging their results;
- Other specific measures (e.g., ordered weighted average).

\[
\begin{align*}
\sigma_c(c, c') &= \frac{1}{\max(|A(c)|, |A(c')|)} \sum_{(a, a') \in \text{match}(A(c), A(c'))} \sigma_a(a, a') + w_C \sigma(N(c), N(c')) \\
\sigma_a(a, a') &= w_A \sigma_c(\text{domain}(a), \text{domain}(a')) + w_N \sigma(\text{domain}(a), \text{domain}(a'))
\end{align*}
\]
Dealing with cycles: fix point computation

\[ \sigma_C(c, c') = \min \left\{ \frac{1}{\max(|A(c)|, |A(c')|)} : (a, a') \in \text{match}(A(c), A(c')) \right\} \]

\[ \sigma_C(c, c') = 6 \cdot \sigma_A(a, a') + 4 \cdot \sigma(N(c), N(c')) \]

\[ \sigma_A(a, a') = 5 \cdot \sigma_C(\text{domain}(a), \text{domain}(a')) + \lambda \cdot \sigma(N(a), N(a')) \]

Threshold reached: no \ 1 variation

\[ \sigma_C(c, c') = \min \left\{ \frac{1}{\max(|A(c)|, |A(c')|)} : (a, a') \in \text{match}(A(c), A(c')) \right\} \]

\[ \sigma_C(c, c') = 6 \cdot \sigma_A(a, a') + 4 \cdot \sigma(N(c), N(c')) \]

\[ \sigma_A(a, a') = 5 \cdot \sigma_C(\text{domain}(a), \text{domain}(a')) + \lambda \cdot \sigma(N(a), N(a')) \]
Filtering similarities: thresholding

- **Hard threshold** retains all the correspondence above threshold $n$;
- **Delta threshold** consists of using as a threshold the highest similarity value out of which a particular constant value $d$ is subtracted;
- **Proportional threshold** consists of using as a threshold the percentage of the highest similarity value;
- **Percentage** retains the $n\%$ correspondences above the others.

Filtering similarities: Softening and hardening

Applies a monotonous function $f : [0, 1] \rightarrow [0, 1]$

- **Hardening** all correspondences with non-1 confidence are assigned 0 confidence;
- **Smoothening** (e.g., sigmoid) consists of using as a threshold the highest similarity value out of which a particular constant value $d$ is subtracted;
- **Weakening** consists of using as a threshold the percentage of the highest similarity value;

Extracting alignments

<table>
<thead>
<tr>
<th></th>
<th>Book</th>
<th>Translator</th>
<th>Publisher</th>
<th>Writer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>.84</td>
<td>0.0</td>
<td>.90</td>
<td>.12</td>
</tr>
<tr>
<td>Provider</td>
<td>.12</td>
<td>0.0</td>
<td>.84</td>
<td>.60</td>
</tr>
<tr>
<td>Creator</td>
<td>.60</td>
<td>.05</td>
<td>.12</td>
<td>.84</td>
</tr>
</tbody>
</table>

- **Greedy algorithm**: 1.96;
Excerpts from the document:

**Summary**

- Ontology heterogeneity is the nature of the semantic web;
- Ontology matching is part of the solution;
- It can be based on many different techniques;
- There already are numerous systems there;
- A relatively solid research field has emerged (tools, formats, evaluation, etc.) and is making progress;
- But there remains serious challenges ahead.

**Challenges 2009**

- Large-scale ontology matching evaluation,
- Efficiency of ontology matching techniques,
- Uncertainty in ontology matching,
- Context-based matching,
- Matcher selection and self-configuration,
- User involvement,
- Explanation of matching results,
- Social and collaborative ontology matching,
- Alignment management: infrastructure and support,
- Reasoning with alignments.

and, of course, many others.

**Acknowledgments**

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

In particular, we are grateful to Than-Le Bach, Jesus Barrasa, Paolo Bouquet, Jan De Bo, Jos De Bruijn, Rose Dieng-Kuntz, Enrico Franconi, Raül García Castro, Manfred Hauswirth, Pascal Hitzler, Mustafa Jarrar, Markus Krötzsch, Ruben Lara, Malgorzata Mochol, Amedeo Napoli, Luciano Serafini, François Sharffe, Giorgos Stamou, Heiner Stuckenschmidt, York Sure, Vojtěch Svátek, Valentina Tamma, Sergio Tessaris, Paolo Traverso, Raphaël Troncy, Sven van Acker, Frank van Harmelen, and Ilya Zaihrayeu.

And more specifically to Marc Ehrig, Fausto Giunchiglia, Loredana Laera, Diana Maynard, Deborah McGuinness, petko Valchev, Mikalai Yatskevich, and Antoine Zimmermann for their support and insightful comments.

Part of this work was carried out while Pavel Shvaiko was with the university of Trento.
<table>
<thead>
<tr>
<th>Problem</th>
<th>Applications</th>
<th>Basic techniques</th>
<th>Process</th>
<th>Conclusions</th>
</tr>
</thead>
</table>

**Ontology matching**

Jérôme Euzenat, Pavel Shvaiko

**Ontology matching**

1. Applications
2. Problem definition
3. Classification
4. Basic techniques
5. Strategies
6. Systems
7. Evaluation
8. Representation
9. Explanation
10. Processing

http://book.ontologymatching.org

---

**Questions?**

pavel@dit.unitn.it

Jerome.Euzenat@inria.fr

http://exmo.inrialpes.fr