Introduction to Ontology Matching and Evaluation

Cássia Trojahn dos Santos
cassia.trojahn@irit.fr

Tutorial – Cameleon Project

17 décembre 2013
Outline

1. The ontology matching problem
2. Overview on matching techniques
3. Hands-on 1: getting started with the Alignment API
4. Ontology matching evaluation
5. Hands-on 2: using real matchers
6. Hands-on 3: evaluating alignments
Outline

1. The ontology matching problem
2. Overview on matching techniques
3. Hands-on 1: getting started with the Alignment API
4. Ontology matching evaluation
5. Hands-on 2: using real matchers
6. Hands-on 3: evaluating alignments
Ontologies

- Language-independent representations of knowledge of a domain of interest
  - define the concepts and relationships used to describe and represent an area of concern
  - define possible constraints on using those concepts and relationships
- Comprise a layer of terminology expressed in natural language
- Taxonomies vs. thesaurus vs. lightweight ontologies vs. full ontologies
Ontologies

- Play a key role in a range of applications:
  - information retrieval
    - query expansion using the domain ontologies
  - web sites annotation
    - using ontologies to add a semantic layer to html-based sites
  - machine translation
    - translation disambiguation using the domain ontologies
but la vie n’est pas rose

- Many ontology designers, many different views on how to model the world

- Just using ontologies, does not reduce heterogeneity: it raises heterogeneity problems at a higher level
  - different ontologies covering different aspects of a domain
  - different ontologies covering same aspects of a domain
  - different schemes for web site annotation
  - different vocabularies for describing data sets

- For many tasks, ontologies have to be conciliated
Kinds of heterogeneity

- Different languages are used (XML, SKOS, OWL)

- Different terminologies are used
  - English vs. Chinese
  - Book vs. Monograph

- Different models are used
  - different classes: Autobiography vs. Paperback
  - classes vs. property: Essay vs. literarygenre
  - classes vs. instances: one physical book as an instance vs. one work as an instance

- Different scopes and granularity are used
  - Only books vs. cultural items vs. any product
Two ontologies

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

Monograph
- isbn
- author
- title

Essay
- Litteraty critics
- Politics
- Biography
- subject
- Autobiography
- Literature
Two ontologies

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

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

Person
- float
- string
- uri

Human
- Writer
Two ontologies

Product
- price
- title
- doi
- creator
- topic

Monograph
- isbn
- author
- title

Essay

Litteraty critics
- Politics
- Biography

Person
- Human
- Writer

CD
- author

Book
- author

DVD

CD
- author

Essay

Literature

Bertrand Russell : My Life

La femme de trente ans
Two ontologies

- Product
  - price
  - title
  - doi
  - creator
  - topic
  - DVD
  - Book
  - CD
- Monograph
  - isbn
  - author
  - title
- Essay
- Person
- Human
- Writer
- Autobiography
- Literature
- Bertrand Russell: My Life
- La femme de trente ans
Correspondence

Definition (Simple correspondence)

Given two ontologies, $o$ and $o'$, a simple correspondence is a quintuple:

$$\langle id, e, e', r, n \rangle,$$

such that:

- $id$ is a URI identifying the given correspondence;
- $e$ and $e'$ are named ontology entities (i.e., named classes, properties, or instances);
- $r$ is a relation among equivalence ($\equiv$), more general ($\supseteq$), more specific ($\subseteq$), and disjointness ($\bot$);
- $n$ is a number in the $[0, 1]$ range.
Definition (Alignment)

An **alignment** $A'$ is a set of correspondences between $o$ and $o'$. $A'$ has some cardinality: 1-1, 1-n, or n-n.
Matching process

\[
\begin{align*}
\text{o} & \rightarrow \text{matching} \\
\text{o'} & \rightarrow \text{matching}
\end{align*}
\]
Matching process

\[ o \rightarrow A \rightarrow \text{matching} \rightarrow o' \]
Matching process
Matching process

\[ A \xrightarrow{\text{parameters}} \text{matching} \xrightarrow{\text{resources}} A' \]
Applications

- **Traditional applications**
  - Ontology evolution
  - Schema integration
  - Catalog integration
  - Data integration

- **Emergent applications**
  - Agent communication
  - Query (translation) answering on the web
  - Web service composition
  - P2P information sharing
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
Outline

1. The ontology matching problem
2. Overview on matching techniques
3. Hands-on 1: getting started with the Alignment API
4. Ontology matching evaluation
5. Hands-on 2: using real matchers
6. Hands-on 3: evaluating alignments
Classification on ontology matching techniques

(a) Granularity of input

Element-level
- Syntactic
- External

Structure-level
- Syntactic
- External
- Semantics

(b) Basic techniques

<table>
<thead>
<tr>
<th>String-based</th>
<th>Language-based</th>
<th>Linguistic resources</th>
<th>Constraint-based</th>
<th>Upper, domain specific formal ontologies</th>
<th>Graph-based</th>
<th>Repository of structures</th>
<th>Model-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>name, description similarity</td>
<td>tokenization, elimination</td>
<td>lexicons, thesauri</td>
<td>type similarity, key properties</td>
<td>DOLCE, FMA</td>
<td>graph homomorphism children, leaves</td>
<td>structure metadata</td>
<td>SAT solvers, DL reasoners</td>
</tr>
</tbody>
</table>

(c) Kind of input

Linguistic
- Terminological

Internal
- Structural

Relational
- Semantic

[Euzenat & Shvaiko, 2007]
Element-level techniques: String-based

- Equality
  - ID = ID

- Prefix
  - net = network but also hot = hotel

- Suffix
  - ID = PID but also word = sword
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
  - $\text{EditDistance}(\text{NKN}, \text{Nikon}) = 0.4$
Element-level techniques : Language-based

- **Tokenization**
  - parses names into tokens by recognising punctuation, cases
  - Personal-Computers → (personal,computers)

- **Lemmatization**
  - analyses morphologically tokens in order to find all their possible basic forms
  - Personal-Computers → (personal,computer)

- **Elimination of stop-words**
  - stop-words : articles, prepositions, conjunctions
  - Writer-of-Paper → (writer,paper)
Element-level techniques : Linguistic resources

- Sense-based (synset) : WordNet (lexical resource)
  - $A \sqsubseteq B \rightarrow A$ is hyponym or meronym of $B$
    - Brand $\sqsubseteq$ Name
  - $A \sqsupseteq B \rightarrow A$ is hypernym or holonym of $B$
    - Europe $\sqsupseteq$ Greece
  - $A \equiv B \rightarrow A$ is synonym of $B$
    - Quantity $\equiv$ Amount
  - $A \perp B \rightarrow A$ is antonym of $B$ or their are siblings in the part of hierarchy
    - Microprocessor $\perp$ PC Board
Element-level techniques : Linguistic resources

- **WordNet gloss comparison (definitions and/or example sentences)**
  - 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
Structure-level techniques: Taxonomy-based

- Ontologies are viewed as graph-like structures containing terms and their inter-relationships
  - **Bounded path matching**
    - These methods 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
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
Structure-level techniques: Tree-based
Structure-level techniques: Tree-based
Ontology matching techniques

- Matching systems rely not on a single technique
  - Sequential composition of alignments
  - Parallel composition of alignments

- ... and more a variety of approaches
  - Negotiation (dialogue)
  - Variations of argumentation frameworks
  - Consensus, voting
  - Intersection (maximising confidence)
  - Merge with filtering out the logical inconsistencies
Outline

1. The ontology matching problem
2. Overview on matching techniques
3. Hands-on 1: getting started with the Alignment API
4. Ontology matching evaluation
5. Hands-on 2: using real matchers
6. Hands-on 3: evaluating alignments
Hands-on 1: getting started with the Alignment API

- Apply simple matching techniques provided in the Alignment API (http://alignapi.gforge.inria.fr/)
- Manipulate the generated alignments

Configuring the environment

```bash
cd material
export CWD='pwd'
echo $CWD
```

Generating a very simple alignment (equal)

```bash
#java -jar lib/procalign.jar <URI onto1> <URI onto2>
java -jar lib/procalign.jar file://$CWD/ontos/Conference.owl file://$CWD/ontos/confOf.owl
```
Hands-on 1: getting started with the Alignment API

- Alignment output: options -o (output) and rendering (-r)

```
# java -jar lib/procalign.jar <URI onto1> <URI onto2> -[options]<parameters>
java -jar lib/procalign.jar file://$CWD/ontos/Conference.owl file://$CWD/ontos/confOf.owl -o alignments/equal.rdf
```
Hands-on 1: getting started with the Alignment API

- Basic alignment methods: -i (implementation)
- Applying a threshold on an alignment: -t (threshold)

```java
java -jar lib/procalign.jar -i fr.inrialpes.exmo.align.impl.method.StringDistAlignment
    -DstringFunction=levenshteinDistance file://$CWD/ontos/Conference.owl file://$CWD/ontos/confOf.owl -o alignments/edit.rdf -t 0.5

.. -i fr.inrialpes.exmo.align.impl.method.StringDistAlignment -
   DstringFunction=smoaDistance

.. Dwndict=$WNDIR -i fr.inrialpes.exmo.align.ling.JWNLAlignment #requires WordNet installation
```
Hands-on 1: further information

- Using the API as basis for creating a matcher
Outline

1. The ontology matching problem
2. Overview on matching techniques
3. Hands-on 1: getting started with the Alignment API
4. Ontology matching evaluation
5. Hands-on 2: using real matchers
6. Hands-on 3: evaluating alignments
What are best matching approaches?

- It depends on how the matching strategies fit the features of the ontologies.

- We need to evaluate the systems:
  - using different ontologies (in size, in formalism, in content)
  - using different metrics (precision, recall, runtime, task-oriented)
Ontology matching evaluation: why?

- Help **developers** of matching systems to improve their systems
- Help **users** evaluating the suitability of proposed systems to their needs

```
 o
  ^
  |  parameters
  v
 matching
  ^
  |  resources
  v
 o'  A'
```
Ontology matching evaluation: why?

- Help developers of matching systems to improve their systems
- Help users evaluating the suitability of proposed systems to their needs
Ontology matching evaluation

**Initiatives**
- Ontology Alignment Evaluation Initiative (OAEI)
  - Annual campaigns since 2004
  - Different tracks from different domains

**SEALS**
- Semantic Evaluation at Large Scale (SEALS)
  - Automatisation of the evaluation process
  - [http://www.seals-project.eu/](http://www.seals-project.eu/)
Automatic evaluation in SEALS

\[ \text{SEALS Bundle} \]

\[ \text{Tool} \]

\[ o \]

\[ o' \]

\[ A \]
Automatic evaluation in SEALS

TestR \rightarrow o \rightarrow Tool \rightarrow A \rightarrow m

○

○'

SEALS Bundle

Local lightweight

R
Automatic evaluation in SEALS
<table>
<thead>
<tr>
<th>test</th>
<th>formalism</th>
<th>relations</th>
<th>confidence</th>
<th>modalities</th>
<th>language</th>
<th>SEALs</th>
</tr>
</thead>
<tbody>
<tr>
<td>benchmark</td>
<td>OWL</td>
<td>=</td>
<td>[0 1]</td>
<td>blind + open</td>
<td>EN</td>
<td>✓</td>
</tr>
<tr>
<td>anatomy</td>
<td>OWL</td>
<td>=</td>
<td>[0 1]</td>
<td>open</td>
<td>EN</td>
<td>✓</td>
</tr>
<tr>
<td>conference</td>
<td>OWL-DL</td>
<td>=, &lt;=</td>
<td>[0 1]</td>
<td>blind + open</td>
<td>EN</td>
<td>✓</td>
</tr>
<tr>
<td>large bio</td>
<td>OWL</td>
<td>=</td>
<td>[0 1]</td>
<td>open</td>
<td>EN</td>
<td>✓</td>
</tr>
<tr>
<td>multifarm</td>
<td>OWL</td>
<td>=</td>
<td>[0 1]</td>
<td>open</td>
<td>CZ, CN, DE, EN, ES, FR, NL, RU, PT</td>
<td>✓</td>
</tr>
<tr>
<td>library</td>
<td>OWL</td>
<td>=</td>
<td>[0 1]</td>
<td>open</td>
<td>EN, DE</td>
<td>✓</td>
</tr>
<tr>
<td>interactive</td>
<td>OWL-DL</td>
<td>=, &lt;=</td>
<td>[0 1]</td>
<td>open</td>
<td>EN</td>
<td>✓</td>
</tr>
<tr>
<td>im-rdft</td>
<td>RDF</td>
<td>=</td>
<td>[0 1]</td>
<td>blind</td>
<td>EN</td>
<td></td>
</tr>
<tr>
<td>System</td>
<td>AML</td>
<td>CIDER-CL</td>
<td>CroMatcher</td>
<td>HerTUDA</td>
<td>HotMatch</td>
<td>IAMA</td>
</tr>
<tr>
<td>--------------</td>
<td>-----</td>
<td>----------</td>
<td>------------</td>
<td>---------</td>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>Confidence</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>benchmarks</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>anatomy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>conference</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>multifarm</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>library</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>interactive</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>large bio</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>im-rdft</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>total</td>
<td>7</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>
Outline

1. The ontology matching problem
2. Overview on matching techniques
3. Hands-on 1: getting started with the Alignment API
4. Ontology matching evaluation
5. Hands-on 2: using real matchers
6. Hands-on 3: evaluating alignments
Hands-on 2: using real matchers

1. Aroma
   - http://exmo.inrialpes.fr/software/aroma/
   - uses of the association rule paradigm
   - “An entity A will be more specific than or equivalent to an entity B if the vocabulary (i.e. terms and also data) used to describe A, its descendants, and its instances tends to be included in that of”

2. LogMap
   - http://www.cs.ox.ac.uk/isg/tools/LogMap/
   - applies reasoning and diagnosis capabilities (semantically rich ontologies)
   - exploits the lexicon of ontologies: part of the entity URIs or entity annotations
Hands-on 2: using real matchers

- Use ‘real’ matcher implementations with the help of the SEALS client

- Configuring the environment

  ```
  export SEALS_HOME=$CWD/seals_home/
  ```

- Generating an alignment with Aroma

  ```
  #java -jar ../lib/seals-client-norep-cameleon.jar <folder matcher> <URI onto1> <URI onto2> <file output>
  
  cd seals_home
  java -jar ../lib/seals-client-norep-cameleon.jar $CWD/matchers/ 
  aroma file://$CWD/ontos/Conference.owl file://$CWD/ontos/ 
  confOf.owl $CWD/alignments/aroma.rdf
  ```
Hands-on 2 : using real matchers

- Generating an alignment with LogMap

```
  cd seals_home
  java -jar ../lib/seals-client-norep-cameleon.jar $CWD/matchers/
    logmap file://$CWD/ontos/Conference.owl file://$CWD/ontos/
    confOf.owl $CWD/alignments/logmap.rdf
```
Hands-on 2: further information

- Packaging your matcher
Outline

1. The ontology matching problem
2. Overview on matching techniques
3. Hands-on 1: getting started with the Alignment API
4. Ontology matching evaluation
5. Hands-on 2: using real matchers
6. Hands-on 3: evaluating alignments
Hands-on 3: evaluating alignments

- Evaluating a single alignment
- Comparison with a reference alignment
- Measuring Precision, Recall and F-measure

Evaluating single alignments

```
#java -cp lib/procalign.jar fr.inrialpes.exmo.align.cli.EvalAlign -i <evaluator> <URI refalign> <URI align>
cd .. % go to material/
```
Evaluating multiple alignments

```
#java -cp ../lib/procalign.jar fr.inrialpes.exmo.align.cli.
  GroupEval -r <file refalign> -l <list matchers> -f <prft> -o
  <file output>

cd .. % go to material/
cp references/conference-confOf.rdf alignments/refalign.rdf
cd alignments/
mkdir conference-confOf
mv *.rdf conference-confOf/
java -cp ../lib/procalign.jar fr.inrialpes.exmo.align.cli.
  GroupEval -r refalign.rdf -l "refalign,equal,edit,aroma,
  logmap" -f prf -o ../results/eval.html
```
Hands-on 3 : evaluating alignments

- Generating precision/recall plots

```java
java -cp lib/procalign.jar fr.inrialpes.exmo.align.cli.GenPlot -l "refalign,equal,edit,aroma,logmap" -t tex -o ../results/prgraph.tex
mv *.table ../results
cd ../results
pdflatex prgraph.tex
```
Credits and references

Credits
Parts of this material are freely inspired by the tutorial of Euzenat at ONTOBRAS 2012 and tutorials on the Alignment API.


References

Questions?