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

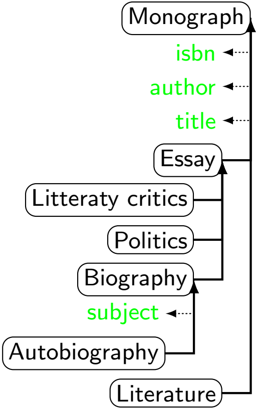
Outline

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

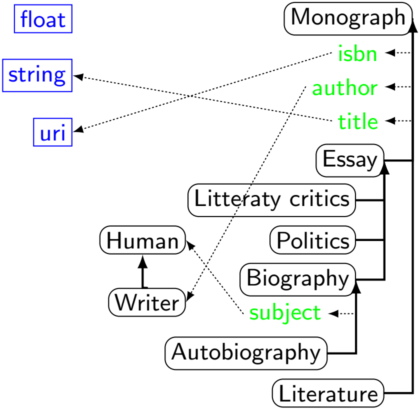
Ontologies

- 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

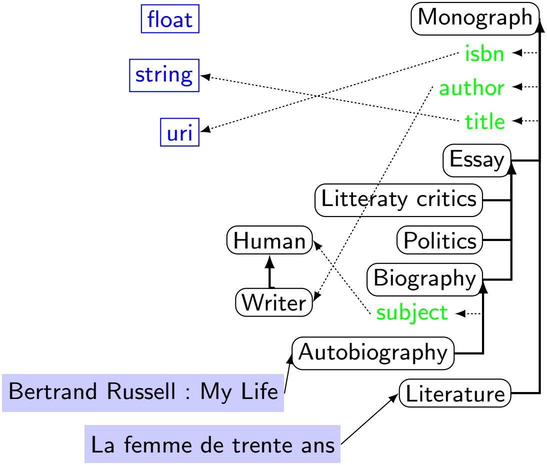
Ontologies


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

Ontologies


float
string
uri
Monograph
isbn
author
title
Essay
Litteraty critics
Politics
Biography
subject
Autobiography
Literature
Human
Writer

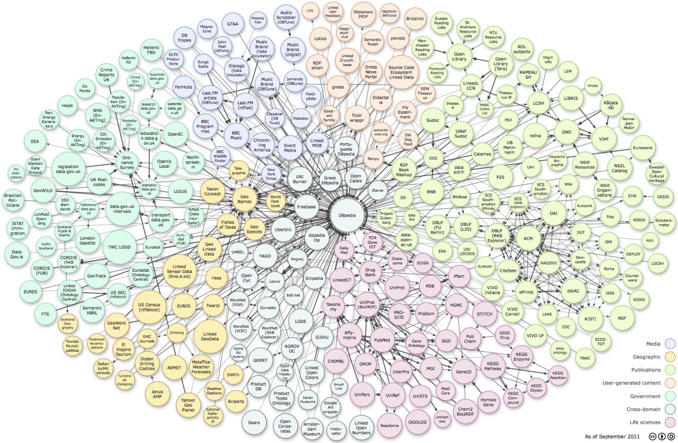
Ontologies


float
string
uri
Monograph
isbn
author
title
Essay
Litteraty critics
Human
Politics
Biography
Writer
subject
Autobiography
Bertrand Russell : My Life
Literature
La femme de trente ans

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

Ontologies in the Linked Open Data age



As of September 2011

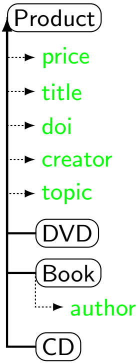
.. but la vie n'est pas rose

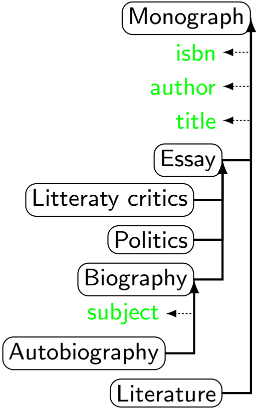
- Many ontology designers, many different views on how to model the world
- Just using ontologies, does not reduce heterogeneity : it raises heterogeneity problems at a higher level
  - different ontologies covering different aspects of a domain
  - different ontologies covering same aspects of a domain
  - different schemes for web site annotation
  - different vocabularies for describing data sets
- For many tasks, ontologies have to be conciliated

Kinds of heterogeneity

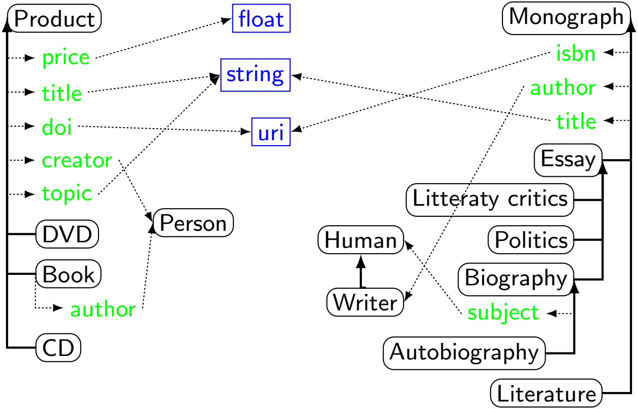
- Different languages are used (XML, SKOS, OWL)
- Different terminologies are used
  - English vs. Chinese
  - Book vs. Monograph
- Different models are used
  - different classes : Autobiography vs. Paperback
  - classes vs. property : Essay vs. literarygenre
  - classes vs. instances : one physical 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

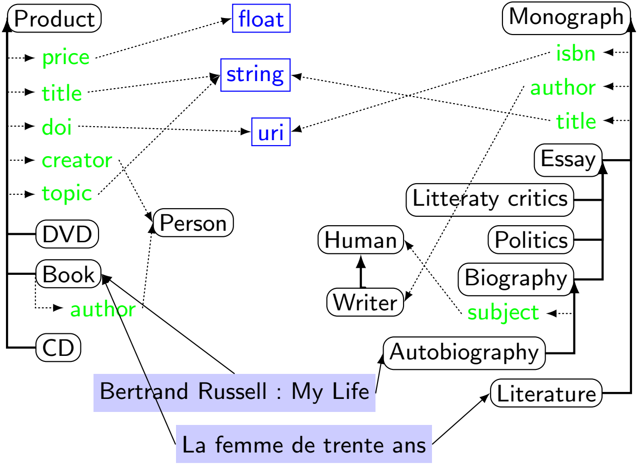




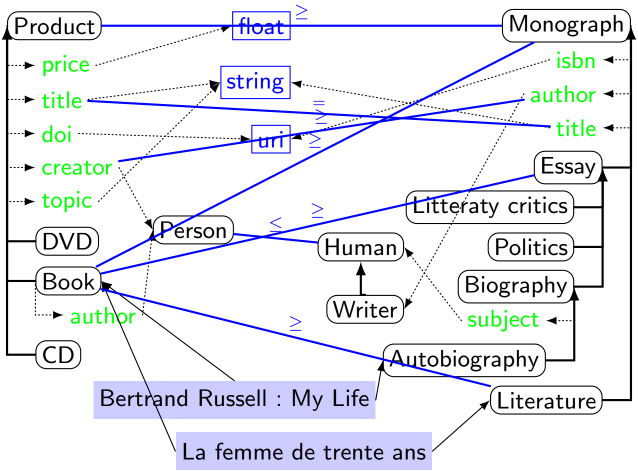
Two ontologies



Two ontologies



Two ontologies



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 :

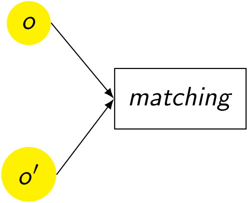
- id is a URI identifying the given correspondence ;
- e and e' are named ontology entities (i.e., named classes, properties, or instances) ;
- r is a relation among equivalence ($\equiv$), more general ($\sqsupseteq$), more specific ($\sqsubseteq$), and disjointness ($\perp$) ;
- n is a number in the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

Alignment

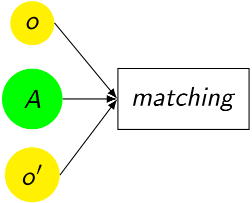
Definition (Alignment)

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

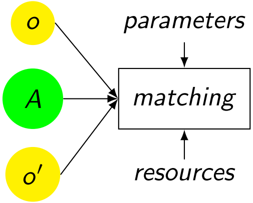
Matching process



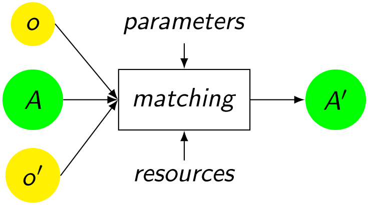
Matching process



Matching process



Matching process



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

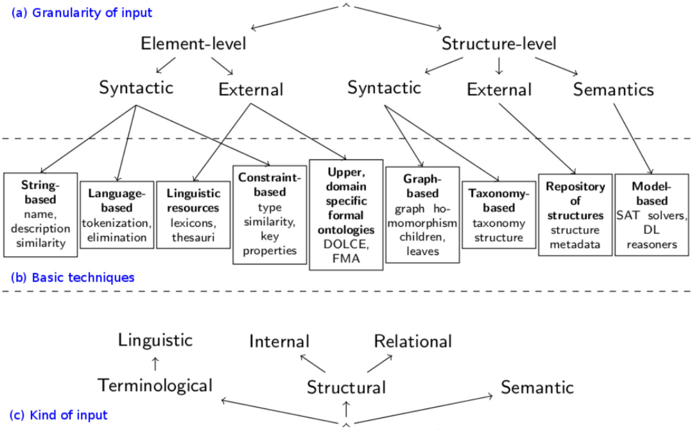
Matching dimensions

- 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
Structure-level
Syntactic
External
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
(b) Basic techniques
Linguistic
Internal
Relational
Terminological
Structural
Semantic
(c) Kind of input

[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
  - EditDistance(NKN,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

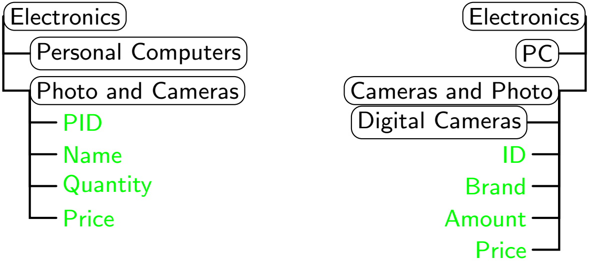
Structure-level techniques : Taxonomy-based

- 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

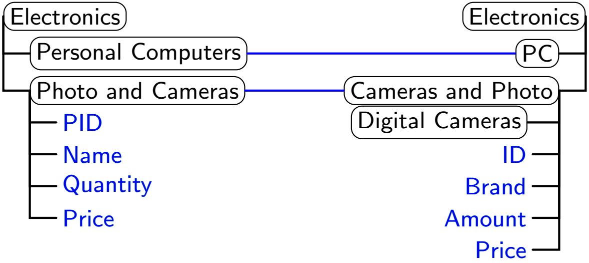
Structure-level techniques : Tree-based



- Electronics
  - Personal Computers
  - Photo and Cameras
    - PID
    - Name
    - Quantity
    - Price

- Electronics
  - PC
  - Cameras and Photo
    - Digital Cameras
    - ID
    - Brand
    - Amount
    - Price

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

```
cd material
export CWD=‘pwd‘
echo $CWD
```

- Generating a very simple alignment (equal)

```
#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

java -jar lib/procalign.jar file://$CWD/ontos/Conference.owl file://$CWD/
    ontos/confOf.owl -r fr.inrialpes.exmo.align.impl.renderer.
    HTMLRendererVisitor -o alignments/equal.html
```

Hands-on 1 : getting started with the Alignment API

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

```
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
- `http://alignapi.gforge.inria.fr/tutorial/tutorial3/index.html`

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

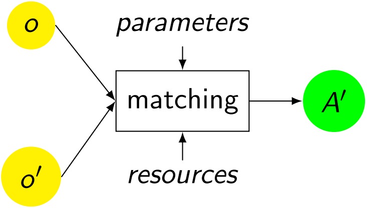
What are best matching approaches ?

- 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)

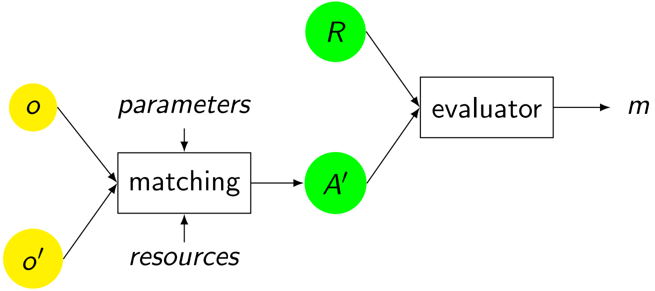
Ontology matching evaluation : why ?

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



Ontology matching evaluation : why ?

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

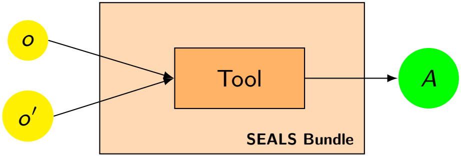


Ontology matching evaluation

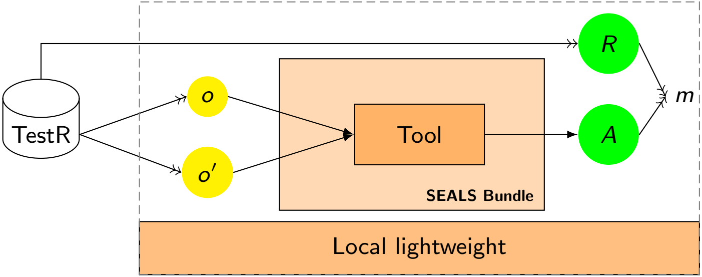
Initiatives Ontology Alignment Evaluation Initiative (OAEI)
Annual campaigns since 2004
Different tracks from different domains
`http://oaei.ontologymatching.org/`

SEALS Semantic Evaluation at Large Scale (SEALS)
Automatisation of the evaluation process
`http://www.seals-project.eu/`

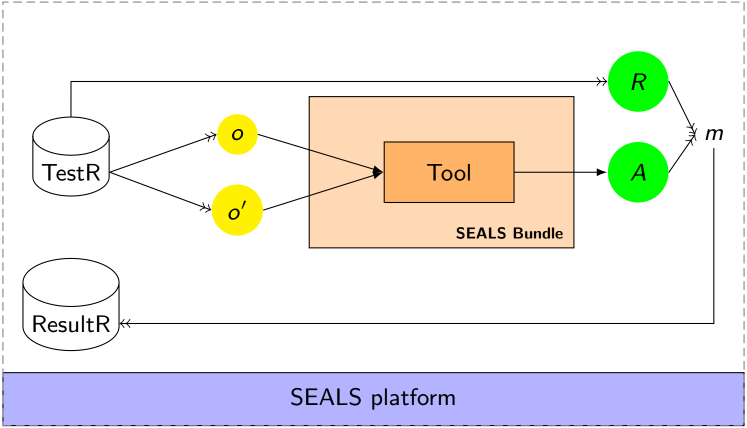
Automatic evaluation in SEALS



Automatic evaluation in SEALS



Automatic evaluation in SEALS



OAEI 2013

test	formalism	relations	confidence	modalities	language	SEALS
benchmark	OWL	=	[0 1]	blind+open	EN	√
anatomy	OWL	=	[0 1]	open	EN	√
conference	OWL-DL	=, <=	[0 1]	blind+open	EN	√
large bio	OWL	=	[0 1]	open	EN	√
multifarm	OWL	=	[0 1]	open	CZ, CN, DE, EN, ES, FR, NL, RU, PT	√
library	OWL	=	[0 1]	open	EN, DE	√
interactive	OWL-DL	=, <=	[0 1]	open	EN	√
im-rdft	RDF	=	[0 1]	blind	EN	

OAEI 2013

System	AML	CIDER-CL	CroMatcher	HerTUDA	HotMatch	IAMA	LilyIOM	LogMap	LogMapLite	MaasMtch	MapSSS	ODGOMS	OntoK	RiMOM2013	ServOMap	SLINT++	SPHeRe	StringsAuto	Synthesis	WeSeE	WikiMatch	XMap	YAM++	Total=23
Confidence	√	√	√				√	√		√		√	√	√	√	√				√		√	√	14
benchmarks	√	√	√	√	√	√		√	√	√	√	√	√	√	√			√	√	√	√	√	√	20
anatomy	√	√		√	√	√		√	√	√	√	√			√			√		√	√	√	√	17
conference	√	√	√	√	√	√		√	√	√	√	√	√	√	√			√	√	√	√	√	√	20
multifarm	√	√		√	√	√		√	√	√	√	√	√	√				√	√	√	√		√	17
library	√			√	√	√		√	√			√			√			√				√	√	11
interactive	√			√				√												√				4
large bio	√			√	√	√		√	√	√		√			√		√	√				√	√	13
im-rdft							√	√						√		√								4
total	7	4	2	7	6	6	1	8	6	5	4	6	3	4	5	1	1	6	3	5	4	5	6	106

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
- `http://oaei.ontologymatching.org/2013/seals-eval.html`

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/
java -cp lib/procalign.jar fr.inrialpes.exmo.align.cli.EvalAlign -i
     fr.inrialpes.exmo.align.impl.eval.PRecEvaluator file://$CWD/
    references/conference-confOf.rdf file://$CWD//alignments/equal
    .rdf
```

Hands-on 3 : evaluating alignments

- 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 -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

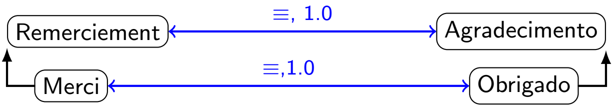
Credits

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

The ontologies used in this tutorial come from
`http://nb.vse.cz/~svatek/ontofarm.html` and
`http://oaei.ontologymatching.org/2013/conference/index.html`

References

- J. Euzenat and P. Shvaiko. Ontology Matching. Springer-Verlag, Heidelberg (DE), 2007.



Questions ?