Semi-Automatic Information and Knowledge Systems
:
Ontology Mapping & Alignment

Monika Lanzenberger

- Introduction
- Similarity Layers
- General Alignment Process
- Application Scenarios
- Complexity & Evaluation
- Using Mappings

- Ontology Mapping: for each ontological entity in the first ontology, we try to locate a corresponding entity in the second ontology, with the same or the closest semantics. It constitutes a fragment of more ambitious tasks such as the alignment of ontologies.
- Ontology Alignment: bringing two ontologies into mutual agreement, making them consistent and coherent with one and another. It may often include a transformation of the source ontologies removing the unnecessary information and integrating missing information.
- Whereas alignment merely identifies the relation between ontologies , mappings focus on the representation and the execution of the relations for a certain task.

[Ehrig, et al. 2004, Ehrig 2007]

- Entities are the same, if their features are the same or similar enough.
- Features represent a certain meaning
- Low similarity may not give evidence for alignments
- High similarity may give strong evidence for alignments
- Not every similarity is of equal importance

[Ehrig 2007]

Introduction: Relations Among Concepts

OWL Ontology Construct	Comparison Relationship	Description
Concept	Equal	URI's equal.
		Class member instances equal.
	Syntactically equal	Labels are the same.
	Similar	Superclasses are the same.
		Subclasses are the same.
		Data properties are the same.
		Object properties are the same.
		Similar low/high fraction of instances.
	Broader than	Subclass superclass comparison.
	Narrower than	Superclass subclass comparison.
	Different	Class is different from all classes of the second ontology.

Introduction: Relations Among Instances

Instances	Equal	URI's equal.
	Syntactically Equal	Labels are the same.
	Similar	Instances of the same concept.
		Property members are the same.
		Two instances linked via the same property to another instance.
	Different	Instance is different from all instances of the second ontology.

Introduction: Relations Among Data Properties

OWL Ontology Construct	Comparison Relationship	Description
Data Properties	Equal	URI's equal.
	Syntactically Equal	Labels are the same.
	Similar	Data property domains are the same.
		Data super properties are the same.
		Data sub properties are the same.
		Data properties members are the same.
	Different	Data property is different from all data properties of the second ontology.

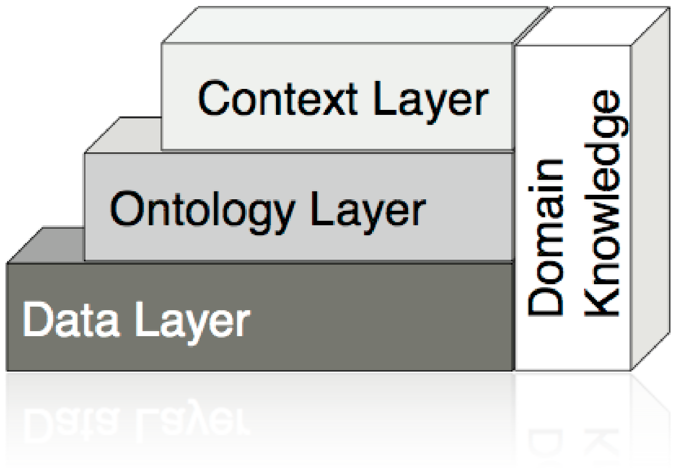
Introduction: Relations Among Object Properties

Object Properties	Equal	URI's equal.
	Syntactically Equal	Labels are the same.
	Similar	Object property domains are the same.
		Object super properties are the same.
		Object sub properties are the same.
		Object properties members are the same.
	Different	Object property is different from all object properties of the second ontology.

Outline

- Introduction
- Similarity Layers
- General Alignment Process
- Application Scenarios
- Complexity & Evaluation
- Using Mappings

Framework for Similarity Computation



[Ehrig 2007]

Ontology Similarity

$$sim : \mathfrak{P}(E) \times \mathfrak{P}(E) \times O \times O \rightarrow [0,1]$$

Positiveness

$$\forall e, f \in \mathfrak{P}(E), O_1, O_2 \in O, sim(e, f, O_1, O_2) \geq 0$$

Maximality

$$\forall e, f, g \in \mathfrak{P}(E), O_1, O_2 \in O, sim(e, e, O_1, O_2) \geq sim(f, g, O_1, O_2)$$

Symmetry

$$\forall e, f \in \mathfrak{P}(E), O_1, O_2 \in O, sim(e, f, O_1, O_2) = sim(f, e, O_2, O_1)$$

[Ehrig 2007]

Ontology Similarity (2)

$$sim : \mathfrak{P}(E) \times \mathfrak{P}(E) \times O \times O \rightarrow [0,1]$$

Two entity sets are identical

$$\forall e, f \in \mathfrak{P}(E), O_1, O_2 \in O, sim(e, f, O_1, O_2) = 1 \Leftrightarrow e = f$$

Two entity sets are similar / different to a certain degree

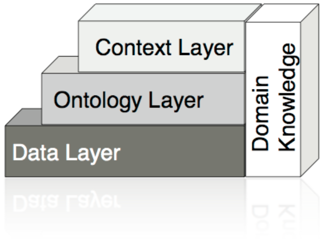
$$\forall e, f \in \mathfrak{P}(E), O_1, O_2 \in O, 0 < sim(e, f, O_1, O_2) < 1$$

Two entity sets are different and have no common characteristics

$$\forall e, f \in \mathfrak{P}(E), O_1, O_2 \in O, sim(e, f, O_1, O_2) = 0 \Leftrightarrow e \neq f$$

[Ehrig 2007]

Data types such as integers, strings etc. are compared by operations such as relative distance and edit distance



Equal values:

$$sim_{equality}(v_1, v_2) := \begin{cases} 1 & \text{if } v_1 = v_2, \\ 0 & \text{otherwise.} \end{cases}$$

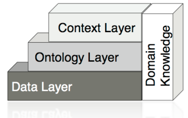
String similarity:

$$sim_{string}(v_1, v_2) := \max(0, \frac{\min(|v_1|, |v_2|) - ed(v_1, v_2)}{\min(|v_1|, |v_2|)})$$

Relative distance:

$$sim_{diff}(v_1, v_2) := 1 - \frac{|v_1 - v_2|}{maxdiff}$$

Object similarity:



- Object Equality

$$sim_{object}(e, f) := \begin{cases} 1 & align(e) = f, \\ 0 & \text{otherwise.} \end{cases}$$

- Explicit Equality

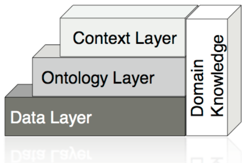
$$sim_{explicit}(e, f) := \begin{cases} 1 & statement(e, \text{"}sameAs\text{"}, f), \\ 0 & \text{otherwise.} \end{cases}$$

Similarity between sets of entities:

- Dice Coefficient $$sim_{dice}(E, F) := \frac{2 \cdot |E \cap F|}{|E| + |F|}$$

- Jacquard Coefficient $$sim_{jacquard}(E, F) := \frac{|E \cap F|}{|E \cup F|}$$

Similarity between sets of entities:



- Single Linkage

$$sim_{single}(E, F) := \max_{(e,f)|e \in E, f \in F} (sim(e, f))$$

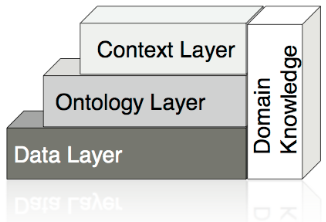
- Average Linkage

$$sim_{complete}(E, F) := \frac{\sum_{\forall (e,f)|e \in E, f \in F} sim(e, f)}{|E| \cdot |F|}$$

- Multi Similarity

$$sim_{set}(E, F) := \frac{\sum_{e \in E} \mathbf{e}}{|\sum_{e \in E} \mathbf{e}|} \cdot \frac{\sum_{f \in F} \mathbf{f}}{|\sum_{f \in F} \mathbf{f}|}$$

with $\mathbf{e} = (sim(e, e_1), sim(e, e_2), ..., sim(e, f_1), sim(e, f_2), ...)$, $\mathbf{f}$ analogously.

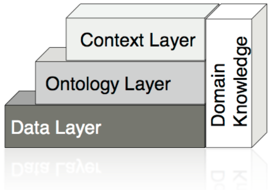


Label similarity:

$$sim_{label}(e, f) := sim_{string}(label(e), label(f))$$

Taxonomic Similarity for Concepts: Extensional

$$sim_{taxonomic}(c_1, c_2) := \begin{cases} e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} & \text{if } c_1 \neq c_2, \\ 1 & \text{otherwise.} \end{cases}$$



Extensional Concept Similarity:

$$sim_{extension}(c_1, c_2) := sim_{set}(\iota_C(c_1), \iota_C(c_2))$$

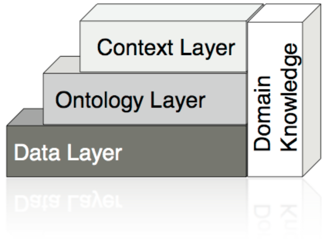
Domain and Range Similarity:

$$sim_{domran}(r_1, r_2) := 0.5 \cdot (\quad sim_{object}(ran(r_1), ran(r_2)) + sim_{object}(dom(r_1), dom(r_2)))$$

Concept Similarity of Instances:

$$sim_{parent}(i_1, i_2) := sim_{object}(c_1, c_2) \quad \text{with } i_1 \in \iota_C(c_1), i_2 \in \iota_C(c_2)$$

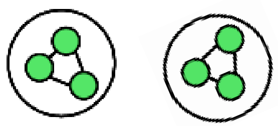
Frequency of usage of an entity or its characteristics in a certain context



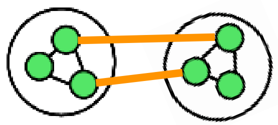
$$sim_{use}(e, f) := sim_{diff}(\text{Usage}(e, con), \text{Usage}(f, con))$$

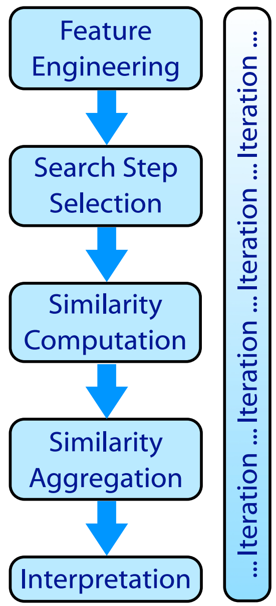
Example: Two books may be similar if their authors have many coauthored publications.

- Introduction
- Similarity Layers
- General Alignment Process
- Application Scenarios
- Complexity & Evaluation
- Using Mappings



$$\begin{aligned} &align : E \times O \times O \rightharpoonup E, \\ \forall e \in E_{O_1} \quad &\exists f \in E_{O_2}, O_1 \in O, O_2 \in O : \\ &align(e, O_1, O_2) = f \\ &\vee\ align(e, O_1, O_2) = \perp \end{aligned}$$





General Alignment Process: Step 1

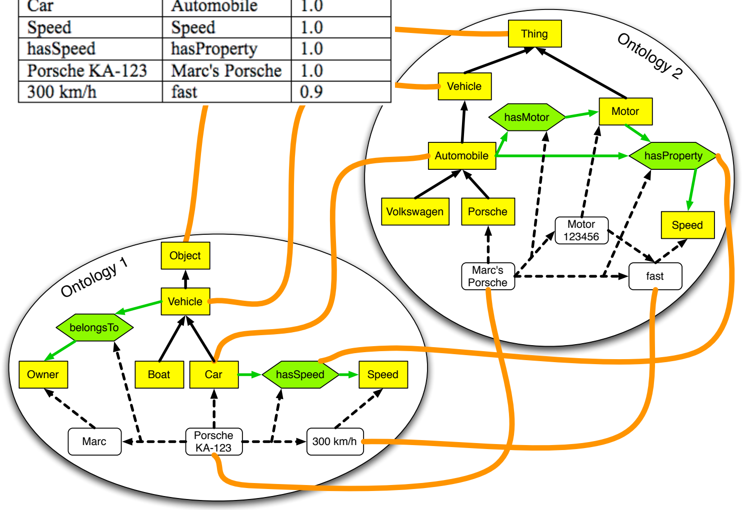
$$feat : O \times O \rightarrow \mathfrak{P}(F)$$

Determine a list of features F:
Extract characteristics of both ontologies, i.e. the features of their ontological entities (concepts C, relations R, and instances I) from intensional and extensional ontology definitions

Feature Engineering
Search Step Selection
Similarity Computation
Similarity Aggregation
Interpretation

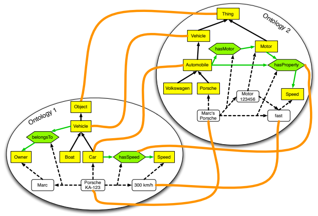
Ontology Alignment Example

Ontology O_1	Ontology O_2	Confidence
Object	Thing	1.0
Vehicle	Vehicle	1.0
Car	Automobile	1.0
Speed	Speed	1.0
hasSpeed	hasProperty	1.0
Porsche KA-123	Marc's Porsche	1.0
300 km/h	fast	0.9



Feature Engineering
Search Step Selection
Similarity Computation
Similarity Aggregation
Interpretation

Ontology Alignment Example: Step 1



The Car concept of ontology 1 is described by its label: Car , its superclass (subclassOf Vehicle), its concept sibling: boat, the direct property: hasSpeed, and its instance Porsche KA-123

The relation hasSpeed is described by the domain: Car and the range: Speed

The instance Porsche KA-123 is characterized by the instantiated property instance: belongsTo: Marc and property instance: hasSpeed: 300 km/h

Feature Engineering
Search Step Selection
Similarity Computation
Similarity Aggregation
Interpretation

General Alignment Process: Step 1

Possible features:

- Identifiers: i.e. strings with dedicated formats, such as unified resource identifiers (URIs) or RDF labels.
- RDF/S Primitives: such as properties or subclass relations
- OWL Primitives: such as an entity being the sameAs another entity
- Derived Features: which constrain or extend simple RDFS primitives (e.g. most-specific-class-of-instance)
- Aggregated Features: i.e. aggregating more than one simple RDFS primitive, e.g. a sibling is every instance-of the parent-concept of an instance
- Domain Specific Features for instance, in an application where files are represented as instances and the relation hashcode-of-file is defined, we use this feature to compare representations of concrete files
- Ontology External Features: Any kind of information not directly encoded in the ontology, such as a bag-of-words from a document describing an instance

Feature Engineering
Search Step Selection
Similarity Computation
Similarity Aggregation
Interpretation

Comparing	No.	Feature	Similarity Measure
Concepts	1	(label,X_1)	string(X_1, X_2)
	2	(identifier,X_1)	explicit(X_1, X_2)
	3	(X_1,sameAs,X_2) relation	object(X_1, X_2)
	4	(direct relations,Y_1)	set(Y_1, Y_2)
	5	all (inherited relations,Y_1)	set(Y_1, Y_2)
	6	all (superconcepts,Y_1)	set(Y_1, Y_2)
	7	all (subconcepts,Y_1)	set(Y_1, Y_2)
	8	(subconc.,Y_1) / (superconc., Y_2)	set(Y_1, Y_2)
	9	(superconc.,Y_1) / (subconc., Y_2)	set(Y_1, Y_2)
	10	(concept siblings,Y_1)	set(Y_1, Y_2)
	11	(instances,Y_1)	set(Y_1, Y_2)
Relations	1	(label,X_1)	string(X_1, X_2)
	2	(identifier,X_1)	explicit(X_1, X_2)
	3	(X_1,sameAs,X_2) relation	object(X_1, X_2)
	4	(domain,X_{d1}) and (range,X_{r1})	object(X_{d1}, X_{d2}), (X_{r1}, X_{r2})
	5	all (superrelations,Y_1)	set(Y_1, Y_2)
	6	all (subrelations,Y_1)	set(Y_1, Y_2)
	7	(relation siblings,Y_1)	set(Y_1, Y_2)
	8	(relation instances,Y_1)	set(Y_1, Y_2)
Instances	1	(label,X_1)	string(X_1, X_2)
	2	(identifier,X_1)	explicit(X_1, X_2)
	3	(X_1,sameAs,X_2) relation	object(X_1, X_2)
	4	all (parent-concepts,Y_1)	set(Y_1, Y_2)
	5	(relation instances,Y_1)	set(Y_1, Y_2)
Relation-Instances	1	(domain,D_1) and (range,R_1)	object(D_1, D_2), (R_1, R_2)
	2	(parent relation,Y_1)	set(Y_1, Y_2)

Feature Engineering
Search Step Selection
Similarity Computation
Similarity Aggregation
Interpretation

$$select : O \times O \rightarrow \mathfrak{P}(E \times E)$$

Most common methods:

- compare all entities of O_1 with all entities of O_2:

$$e, f \in E_1 \times E_2$$

any pair is treated as a candidate mapping

- or only compare entities of the same type

$$e, f \in (C_1 \times C_2) \cup (R_1 \times R_2) \cup (I_1 \times I_2)$$

- or use heuristics to lower the number of candidate mappings (e.g., applied in QOM) using strategies such as random or label, or change propagation

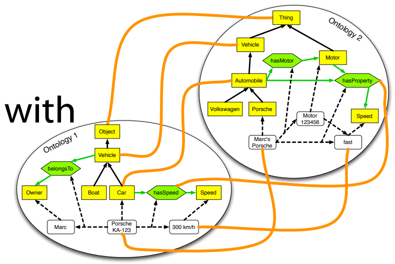
Feature Engineering
Search Step Selection
Similarity Computation
Similarity Aggregation
Interpretation

Compare all entities of the same type:

55 candidate alignments:

- 42 concept pairs (6x7)
- 4 relation pairs (2x2)
- 9 instance pairs (3x3)

e.g., comparing
o_1: belongsTo with
o_2:hasProperty and with
o_2:hasMotor



Feature Engineering
Search Step Selection
Similarity Computation
Similarity Aggregation
Interpretation

$$sim : E \times E \times F \rightarrow [0,1]^k$$

Based on the features F of the ontological entities we do the similarity computation for all pairs of candidates

Additional similarity measures exist

An example is described on the next slide

Feature Engineering
Search Step Selection
Similarity Computation
Similarity Aggregation
Interpretation

Features and Similarity Measures

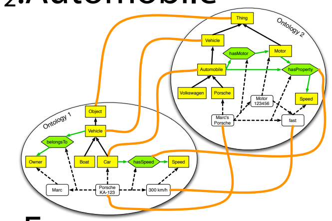
Comparing	No.	Feature	Similarity Measure
Concepts	1	(label,X_1)	string(X_1, X_2)
	2	(identifier,X_1)	explicit(X_1, X_2)
	3	(X_1,sameAs,X_2) relation	object(X_1, X_2)
	4	(direct relations,Y_1)	set(Y_1, Y_2)
	5	all (inherited relations,Y_1)	set(Y_1, Y_2)
	6	all (superconcepts,Y_1)	set(Y_1, Y_2)
	7	all (subconcepts,Y_1)	set(Y_1, Y_2)
	8	(subconc.,Y_1) / (supercon&., Y_2)	set(Y_1, Y_2)
	9	(superconc.,Y_1) / (subconc., Y_2)	set(Y_1, Y_2)
	10	(concept siblings,Y_1)	set(Y_1, Y_2)
	11	(instances,Y_1)	set(Y_1, Y_2)
Relations	1	(label,X_1)	string(X_1, X_2)
	2	(identifier,X_1)	explicit(X_1, X_2)
	3	(X_1,sameAs,X_2) relation	object(X_1, X_2)
	4	(domain,X_{d1}) and (range,X_{r1})	object(X_{d1}, X_{d2}), (X_{r1}, X_{r2})
	5	all (superrelations,Y_1)	set(Y_1, Y_2)
	6	all (subrelations,Y_1)	set(Y_1, Y_2)
	7	(relation siblings,Y_1)	set(Y_1, Y_2)
	8	(relation instances,Y_1)	set(Y_1, Y_2)
Instances	1	(label,X_1)	string(X_1, X_2)
	2	(identifier,X_1)	explicit(X_1, X_2)
	3	(X_1,sameAs,X_2) relation	object(X_1, X_2)
	4	all (parent-concepts,Y_1)	set(Y_1, Y_2)
	5	(relation instances,Y_1)	set(Y_1, Y_2)
Relation-Instances	1	(domain,D_1) and (range,R_1)	object(D_1, D_2), (R_1, R_2)
	2	(parent relation,Y_1)	set(Y_1, Y_2)

Feature Engineering → Search Step Selection → Similarity Computation → Similarity Aggregation → Interpretation

[Ehrig 2007]

Ontology Alignment Example: Step 3

Computing the similarity of the candidate alignment: o_1:Car and o_2:Automobile



For every feature we compute a similarity. E.g.,

sim_{label}(o_1:Car,o_2:Automobile) = sim_{string}('Car', 'Automobile') = **0.0**

$sim_{superconcept}$ (o_1:Car,o_2:Automobile) = sim_{set}({o_1:Vehicle}, {o_2:Vehicle}) = **1.0**

Feature Engineering → Search Step Selection → Similarity Computation → Similarity Aggregation → Interpretation

General Alignment Process: Step 4

$$agg : [0,1]^k \to [0,1]$$

The individual similarity measures are weighted and combined

$$sim_{aggs}(e,f) := \frac{\sum_{k=1..n} w_k \cdot adj_k(sim_k(e,f))}{\sum_{k=1..n} w_k}$$

Some approaches for aggregation:
Averaging:

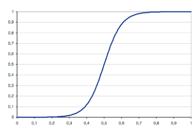
$$w_k = 1, adj_k(x) = id(x)$$

Linear Summation:

$$w_k \text{ learned or manually assigned, } adj_k(x) = id(x)$$

Linear Summation with negative evidence: w_k: can have a negative value (e.g., superconcepts of the first entity have a high similarity with subconcepts of the second entity)
Sigmoid Function: emphasize high similarity and de-emphasize low similarity: $adj_k(x) = sig_k(x - 0.5),$

$$sig_k(x) = \frac{1}{1 + e^{-a_k x}}$$



[Ehrig 2007]

Feature Engineering → Search Step Selection → Similarity Computation → Similarity Aggregation → Interpretation

General Alignment Process: Step 5

$$inter : [0,1] \rightharpoonup \{alignment\}$$

Aggregated similarity is compared to a threshold:
every value above indicates an alignment

Determine the threshold:

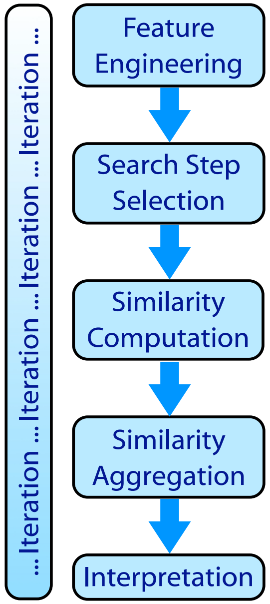
- $constant,$
- $max(sim_{agg}(e,f)) - constant,$
- $max(sim_{agg}(e,f))(1-p)$

Feature Engineering → Search Step Selection → Similarity Computation → Similarity Aggregation → Interpretation

[Ehrig 2007]

General Alignment Process: Step 6

- Entities are similar if their position in the structure is similar
- Structure similarity is expressed through the similarity of the other entities in the structure
- Calculating the similarity for one entity pair, the similarity of the neighboring entity pairs are needed
- In a first round only basic comparison methods (e.g. string similarity) is applied (or pre-given alignments are used)
- In further rounds already computed pairs and use more sophisticated structural similarity measures
- When to stop the iteration:
  1. fixed number of iterations
  2. fixed time constraint
  3. changes of alignments compared to a threshold



[Ehrig 2007]

Outline

- Introduction
- Similarity Layers
- General Alignment Process
- Application Scenarios
- Complexity & Evaluation
- Using Mappings

Ontology Merging

- Two ore more ontologies
  are combined into one target ontology
- By establishing alignments among entities
  we identify equal entities which we can merge

- Time resources are less critical
- Human post-processing is required
- Finally high quality requirements

[Ehrig 2007]

Web Service Composition

- Agents or web service often use different representations of their domains resulting in different expressions on their goals, and their input or output
- Collaborate despite the heterogeneous representations
- Standard upper-level ontologies or ontology alignment

- Alignment needs to be fast, reliable, and correct
- Wrong results can lead to unjustified costs
- Sometimes user checks are possible
- Example: combine a booking service of an air carrier and a hotel reservation network

[Ehrig 2007]

- Users formulate a query in a specific query language based on a specific ontology
- Query is sent to a query engine
- To access heterogeneous information sources the query needs to be re-written for the target ontologie(s)
- For the presentation of the answers the results have to be transformed back again

- Rewriting / Mapping should be fast and fully automatic
- Users may tolerate wrong results as long as the correct results are returned as well

- New information is inferred from distributed and heterogeneous ontologies
- Time constraints are not critical (for both, alignment and inference tools)
- Quality of the alignments is very important
- Alignment needs to be done automatically
- Wrong results may trigger additional wrong results in a cascading manner
- Detection of conflicting inconsistencies is required

- Many unsolved research issues !

- Introduction
- Similarity Layers
- General Alignment Process
- Application Scenarios
- Complexity & Evaluation
- Using Mappings

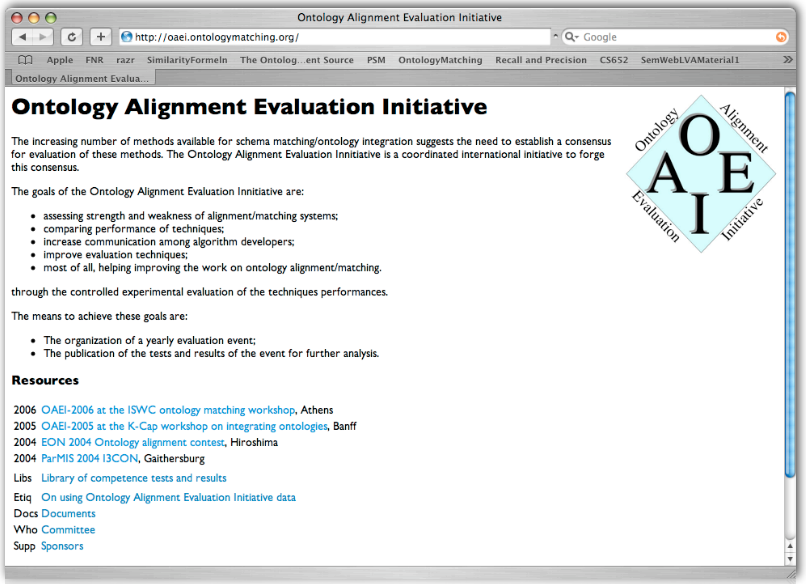
Standard information retrieval metrics

$$\begin{aligned}\text{Precision} \quad p &= \frac{\#correct\_found\_mapping}{\#found\_mappings} \\ \text{Recall} \quad r &= \frac{\#correct\_found\_mappings}{\#existing\_mappings} \\ \text{F-Measure} \quad f_1 &= \frac{2pr}{p+r}\end{aligned}$$

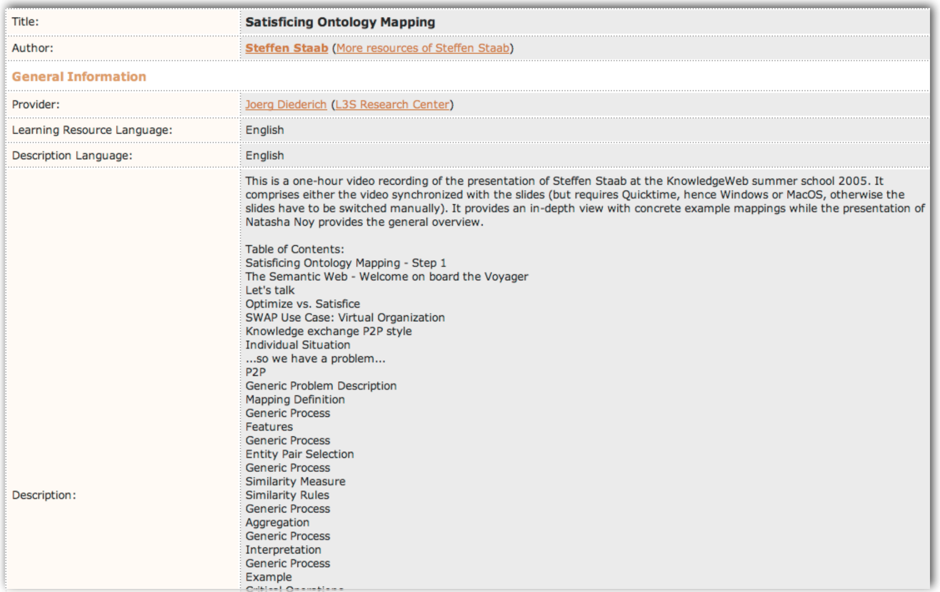
Evaluation

- Compliance measure
  quality of identified alignments
- Performance measure
  quality of algorithm in terms of computational resources
- User-related measure
  overall subjective user satisfaction,
  e.g., how much user effort is needed
- Task-related measure
  quality of alignment for a certain use case or application scenario

Yearly Contest

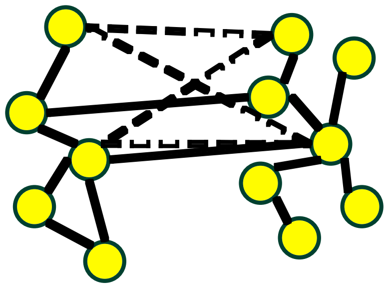


Video Satisficing Ontology Mapping

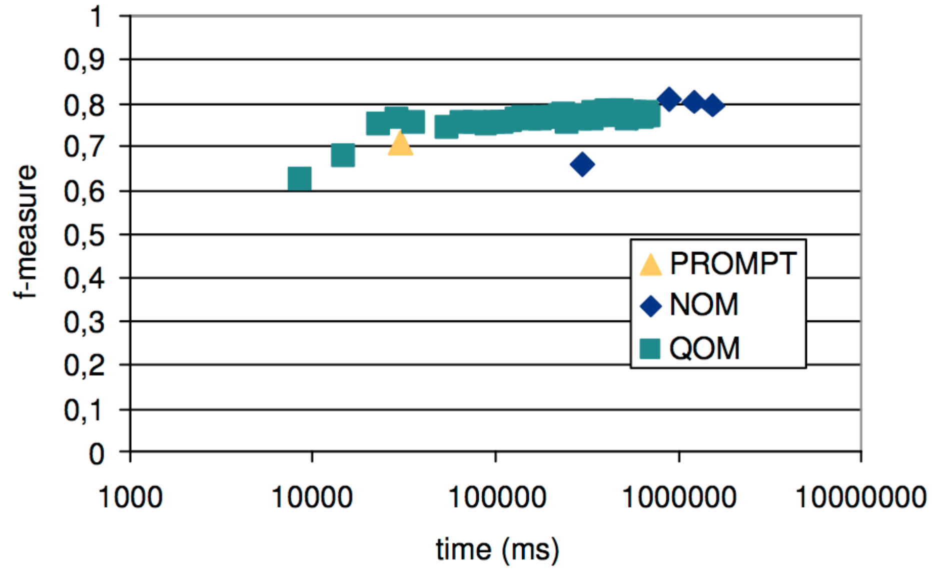


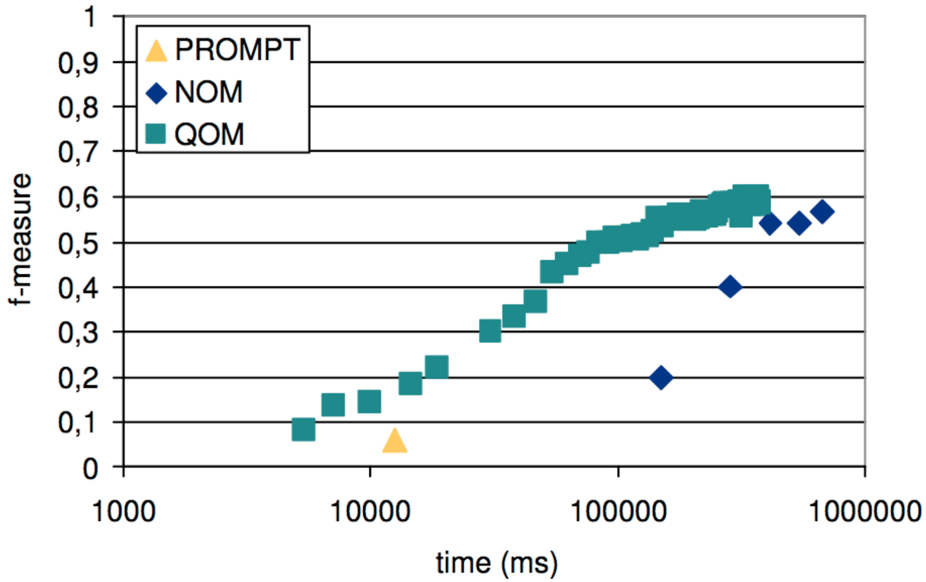
Steffen Staab Video Slide - Reduction of Comparisons

- Random Selection
- Closest Label
- Change Propagation
- Combination

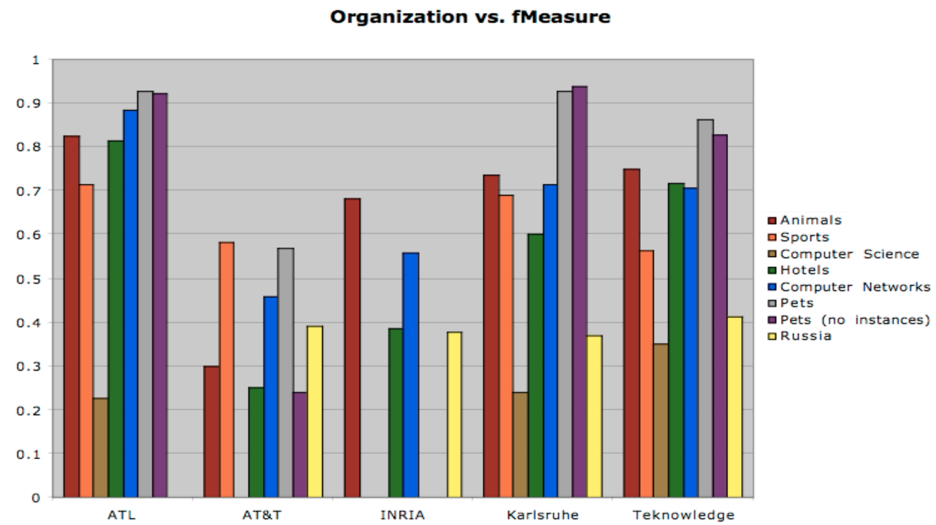


- $c = (\text{feat} + \text{select} + \text{comp} \cdot (\Sigma_k\, \text{sim}_k + \text{agg}) + \text{inter}) \cdot \text{iter}$
- NOM
  $c = O((n + n^2 + n^2 \cdot (\log^2(n) + 1) + n) \cdot 1)$
  $= O(n^2 \cdot \log^2(n))$
- PROMPT
  $c = O((n + n^2 + n^2 \cdot (1 + 0) + n) \cdot 1)$
  $= O(n^2)$
- QOM
  $c = O((n + n \cdot \log(n) + n \cdot (1 + 1) + n) \cdot 1)$
  $= O(n \cdot \log(n))$



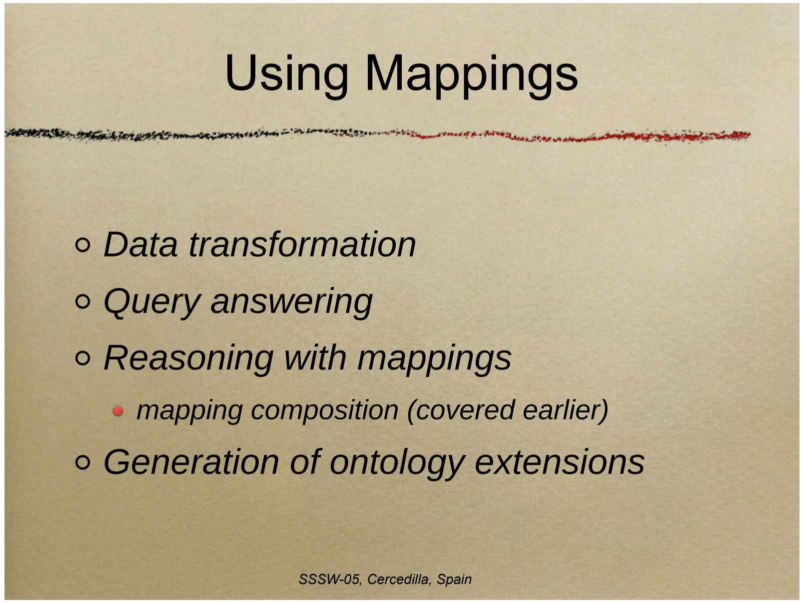


Results for I3CON Ontology Alignment Experiment

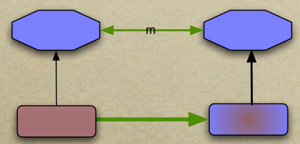


- Introduction
- Similarity Layers
- General Alignment Process
- Application Scenarios
- Complexity & Evaluation
- Using Mappings

Title:	**Ontology Mapping and Alignment**
Author:	Natasha Noy (More resources of Natasha Noy)
General Information	
Provider:	Joerg Diederich (L3S Research Center)
Learning Resource Language:	English
Description Language:	English
Description:	This is a one-hour video recording of the presentation of Natasha Noy at the KnowledgeWeb summer school 2005. It comprises either the video synchronized with the slides (but requires Quicktime, hence Windows or MacOS, otherwise the slides have to be switched manually). It provide a high-level overview on ontology mapping while the presentation of Steffen Staab provides more details and example systems. Table of Contents: Ontology Mapping and Alignment Lots of Overlapping Ontologies on the Semantic Web Example Definitions of School Creating Correspondences Between Ontologies Semantic Integration Tasks Reasons for Mismatches Types of Mismatches Language-level Mismatches Ontology-level Mismatches Ontology-level Mismatches: Examples Some of the Differences Categories of Mappings Mapping Discovery: Information Sources Using a Common Reference Ontology Solve the problem before it arises Using reference ontologies: Problems Using Lexical Information Using Ontology Structure Using External Sources User Input Using Prior Matches Mapping Composition Using Corpus of Matches Mapping Discovery: Information Sources



Data Transformation



- *Mapping Interpreter (Stanford SMI), OntoMerge*

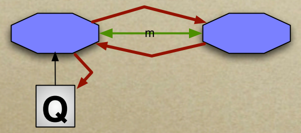
SSSW-05, Cercedilla, Spain

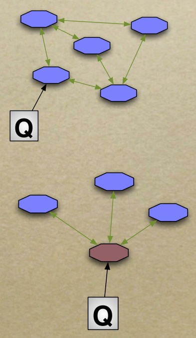
Data Transformation (II)

- *Mapping interpreter (Stanford SMI)*
  - *Uses an instantiated ontology of mappings*
    - *mapping structure*
    - *Python rules*
- *OntoMerge*
  - *Treats source ontologies with data and mapping axioms as a single ontology*
  - *Uses a theorem prover to create new data*

SSSW-05, Cercedilla, Spain

Query Answering

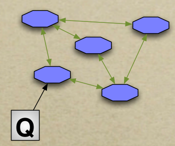




- *Two settings*
  - *one-to-one mappings*
  - *global ontology*

SSSW-05, Cercedilla, Spain

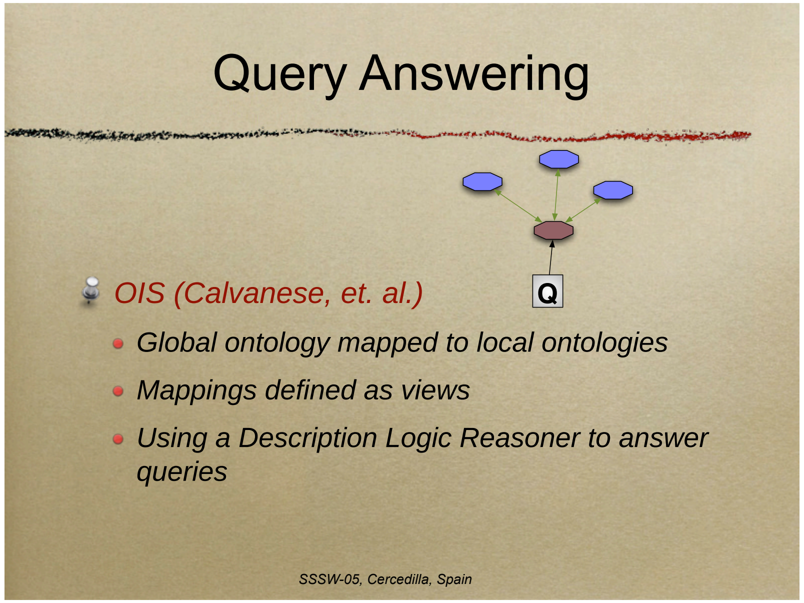
Query Answering (II)



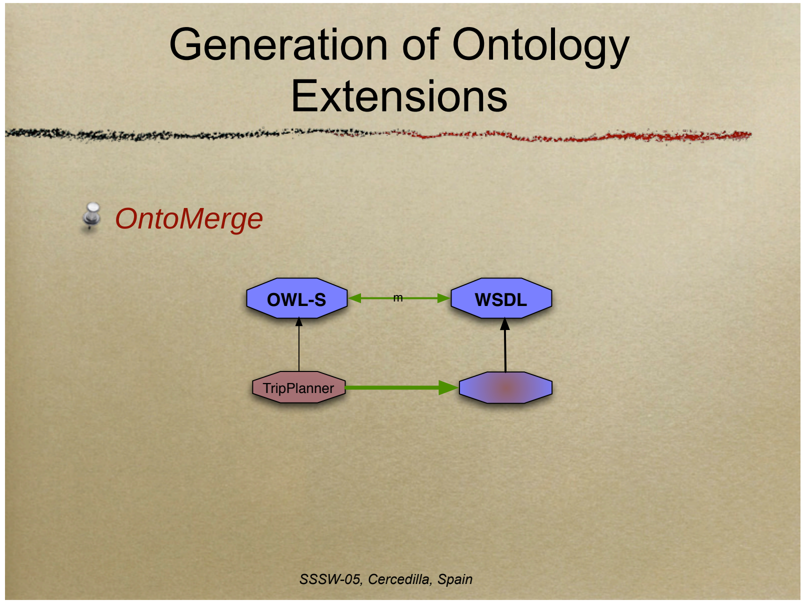
- *Piazza (UW)*
  - *Peer-to-peer architecture for query answering*
  - *Query refomulation using mappings between adjacent peers*

SSSW-05, Cercedilla, Spain

Natasha Noy Video Slide - Using Mappings 57



Natasha Noy Video Slide - Using Mappings 58



References 60

[Ehrig 2007] Marc Ehrig: Ontology Alignment - Bridging the Semantic Gap, Dissertation, Springer, New York, 2007.

[Ehrig, et al. 2004] Marc Ehrig and York Sure: Ontology Mapping - An Integrated Approach, In: Proceedings of the First European Semantic Web Symposium. C. Bussler, J. Davis, D. Fensel, R. Studer, eds, 2004.

[REASE] REASE - Repository of EASE for Learning Units, http://rease.semanticweb.org, checked online 7.Jan.2007.

[Slides Noy Video] Natasha Noy: Ontology Mapping and Alignment, SSSW, 2005, pp. 39- 45, http://dit.unitn.it/~accord/RelatedWork/Matching/Noy-MappingAlignment-SSSW-05.pdf, checked online 7.Jan.2007.

[Slides Staab Video] Steffen Staab: Satisficing Ontology Mapping, SSSW, 2005, http://rease.semanticweb.org/ubp/PUSH/search@srchDetailsLR?lrID=lr-lear-diederich-1165406019525, checked online 7.Jan.2007.