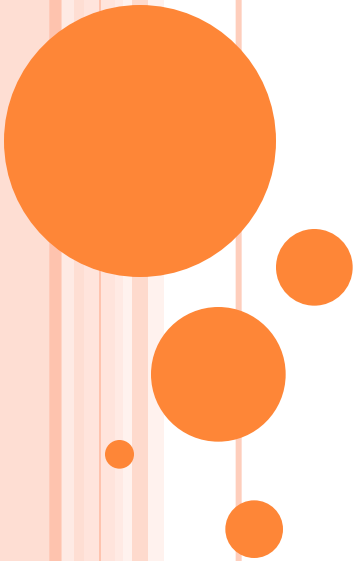
Cours 3 : Ontology Alignment And Ontology Enrichment for Data Integration



Data Integration

Master D&K

pernelle@lri.fr

Ontology

- An ontology provides a vocabulary that describes a domain of interest and a specification of the meaning of terms used in the vocabulary.

- Depending on the precision of this specification, the notion of ontology encompasses several data and conceptual models, including simple taxonomies, or fully axiomatized ontologies.

Semantic web

- The semantic web is an effort for publishing such formal knowledge on the web.
  - RDF allow to expressing data as graphs;
  - OWL, RDFS allow to express ontologies governing such graphs;
  - SPARQL: a query language for such graphs
- There exist many tools for dealing with such languages.
- Tens of billions of RDF triples and thousands of ontologies on the web. Now, governments and their agencies publish their data in RDF.

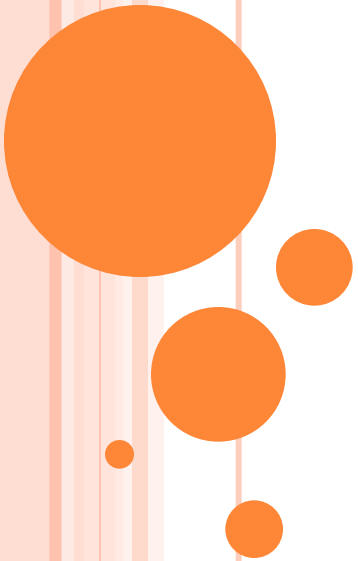
But

It is not one guy's ontology.
It is not several guys' common ontology.

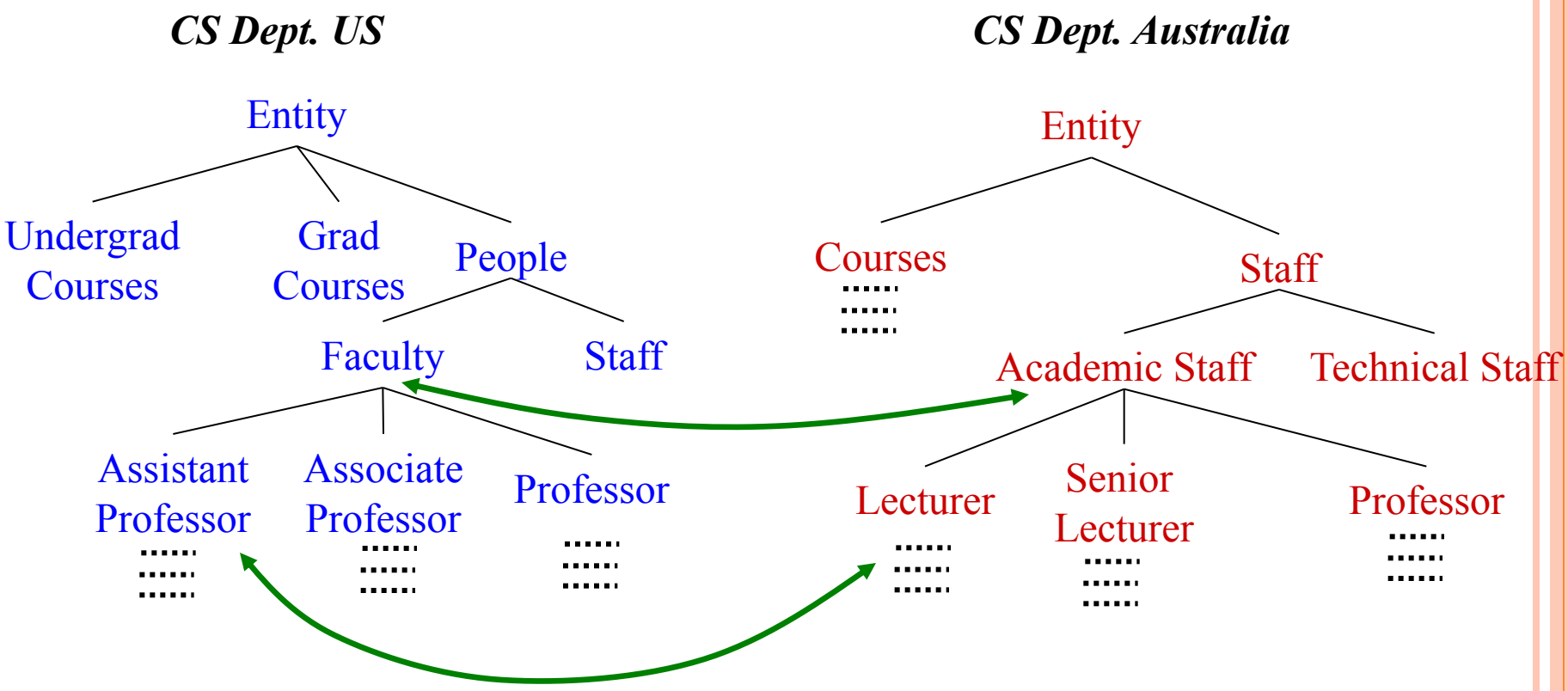
It is many guys' many ontologies.

So it is a mess …

Ontology Alignment



Alignment of a simple taxonomy (Θ=EQUIVALENTCLASS)



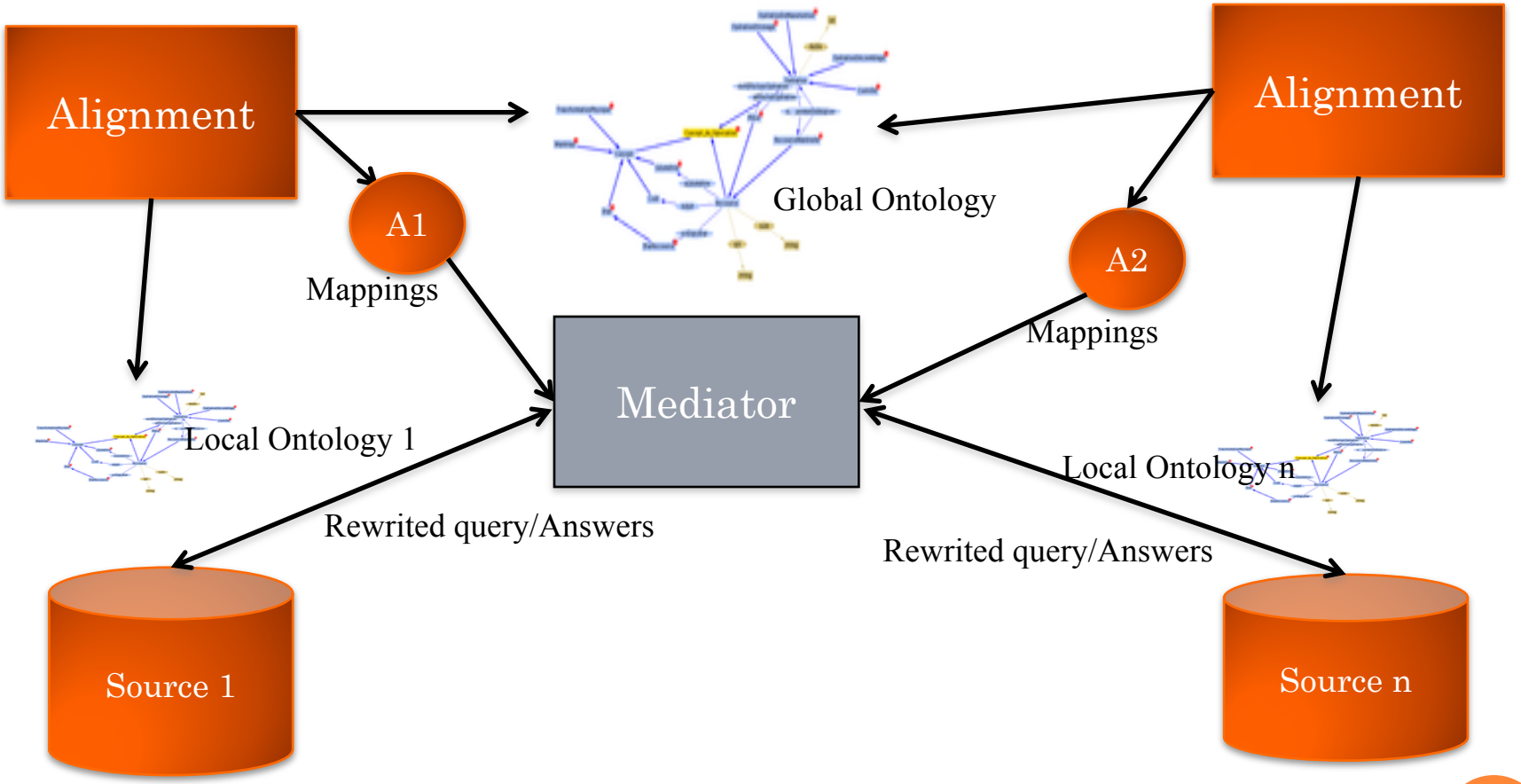
Objective of an Alignment process

- Data integration provided by other organisations (ex: providers) in a Data Warehouse
  (ETL construction).

- Data Integration in a virtualized approach (Wrapper construction)

- Information shared in a P2P architecture
- Composition of Web services.
- Ontology Evolution/Ontology Fusion.

Alignments used in a virtualized approach

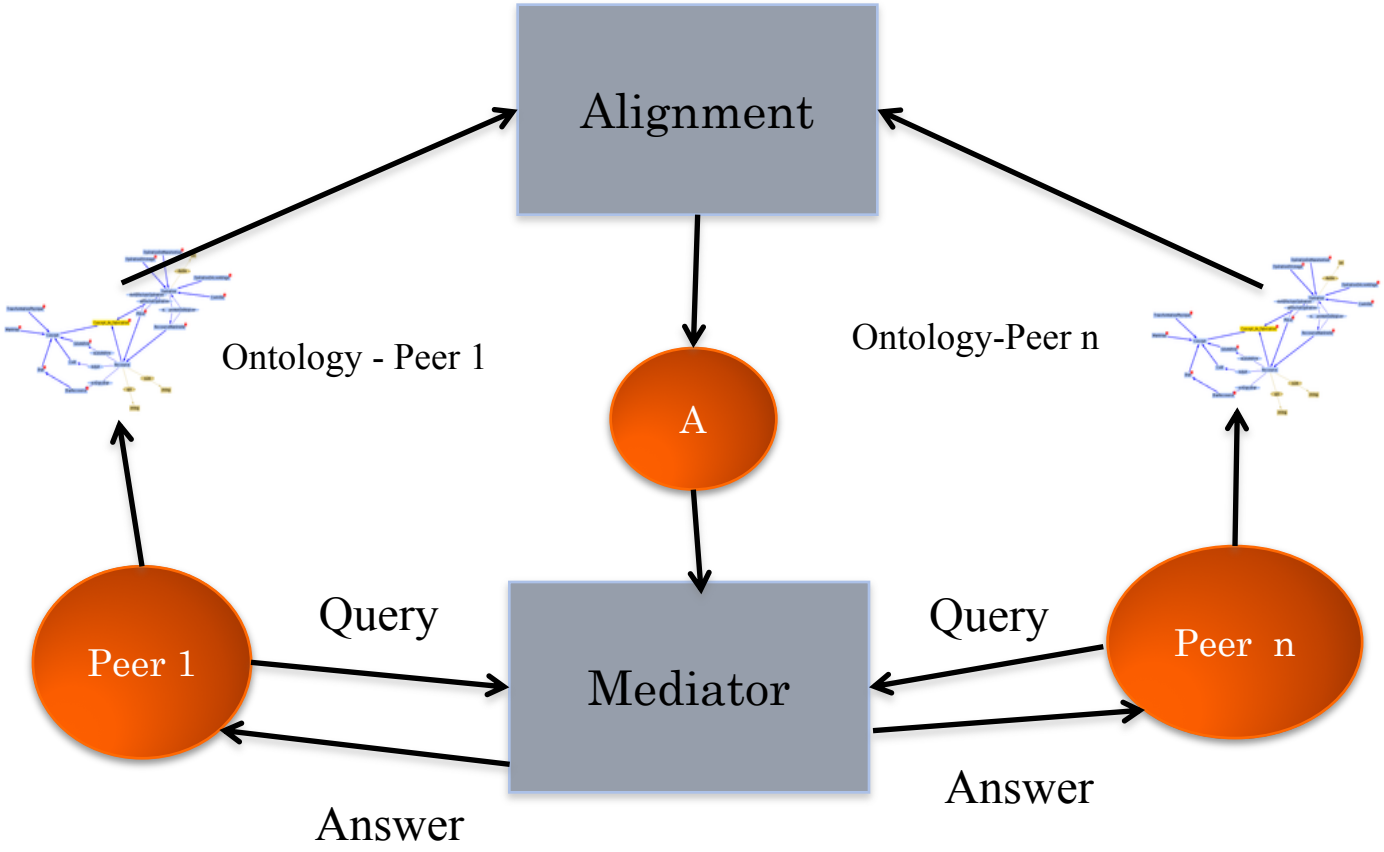


Sources are described by a local ontology.

The alignment allows to discover mappings between a local ontology and a global ontology.

Queries are formulated in the global ontology vocabulary. The mediator reformulates the query to obtain answers.

Alignments in P2P Architectures



What kind of hetereogeneity ?

Syntaxic: 2 ontologies are not described using the same langage (OWL-DL / RDFS): equivalence between language constructs, transformations, abstractions are needed.

Terminological : variations in equivalent concept labels. (Homonymies, Synonymies, various languages)
CD/Compact Disk, paper/publication, paper/papier

Conceptual : differences between two models of the same domain (coverage, granularity, different points of view)

Conceptual heterogeneity

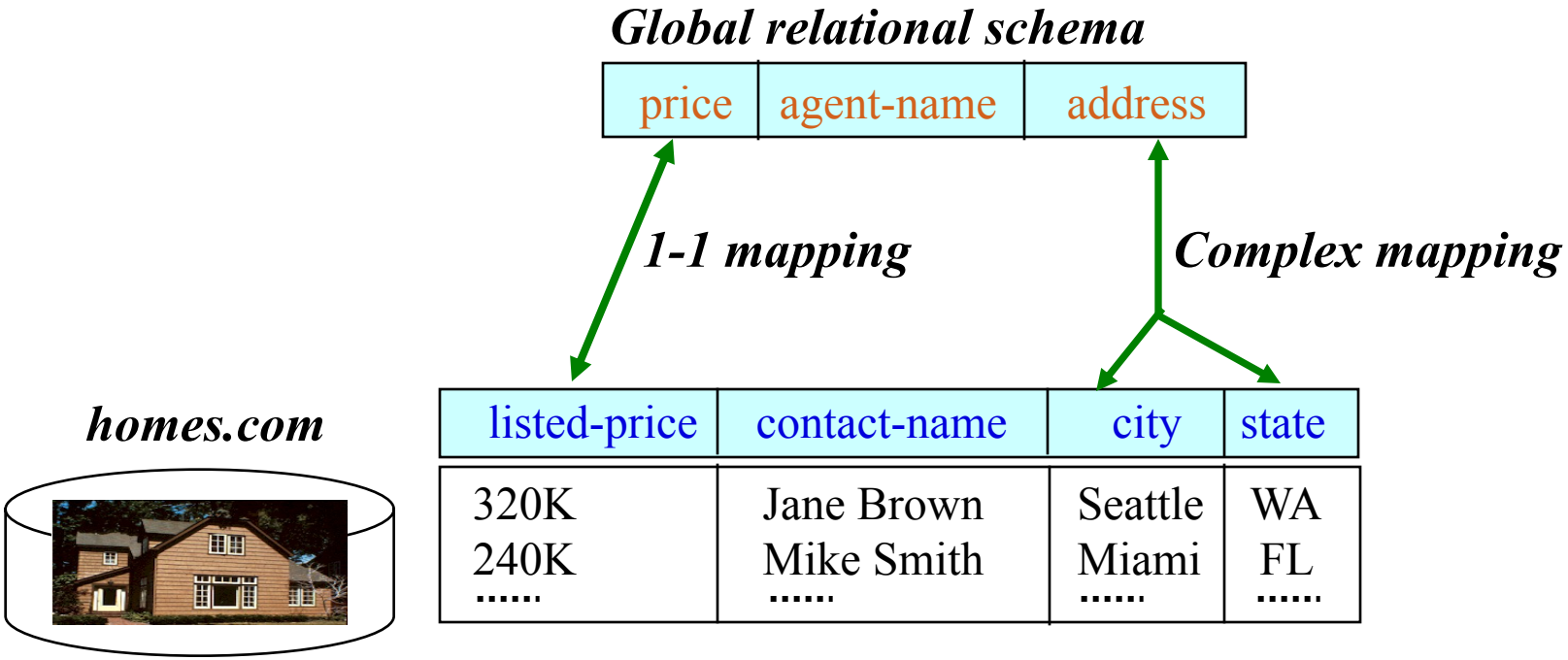
- **Coverage**: the two ontologies describe different domains but share a more or less important set of concepts and properties.

- **Granularity**: the two ontologies describe the same domain, but they are more or less accurate (ex: detailed bio-medical ontology for experts, versus simple ontology used by ordinary patients).

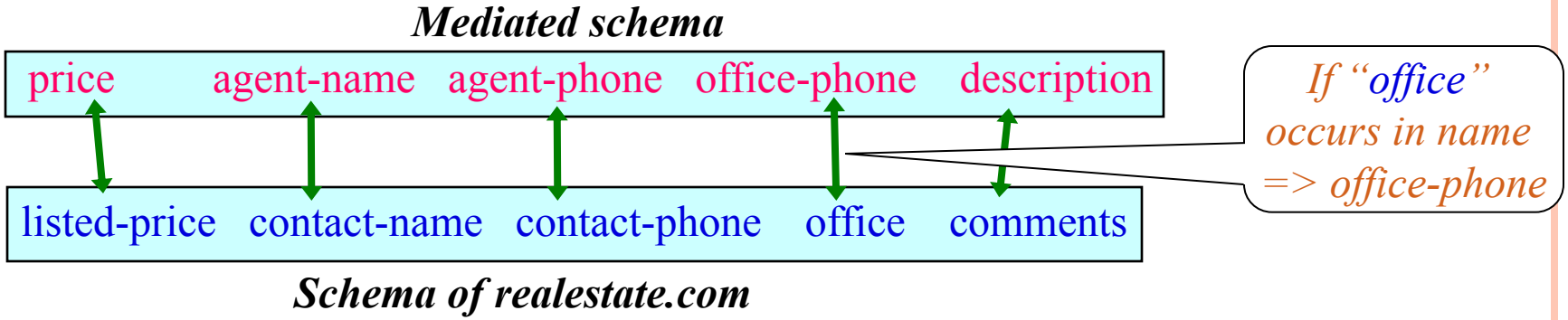
- **Distinct Point of views**

(ex : geographical vs geopolitical)

Similar problem in schema matching (relational databases)



Similar problem in schema matching



realestate.com

listed-price	contact-name	contact-phone	office	comments
$250K	James Smith	(305) 729 0831	(305) 616 1822	Fantastic house
$320K	Mike Doan	(617) 253 1429	(617) 112 2315	Great location
.....	.....	.....	.....	.....

If "fantastic" & "great" occur frequently in data instances => description

homes.com

sold-at	contact-agent	extra-info
$350K	(206) 634 9435	Beautiful yard
$230K	(617) 335 4243	Close to Seattle

What is an Ontology Alignment ?

- Alignment A: set of mappings declared between ontology entities (properties or classes) of two ontologies O1 and O2.

$$f(O1,O2)=A$$

Θ is the set of relations that can be used to express a mapping (chosed by the alignment approach).

Example: Θ={owl:disjointWith, owl:equivalentClass, rdfs:subclassOf, closeTo…}

A = {owl:equivalentClass(http://amaz.com/dvd, http://fnac.com/etol/filmdvd), closeTo(http://amaz.com/road, http://ign.com/tronçon-route) }

What is an Ontology Alignment ?

- Each mapping can be associated with a confidence degree.

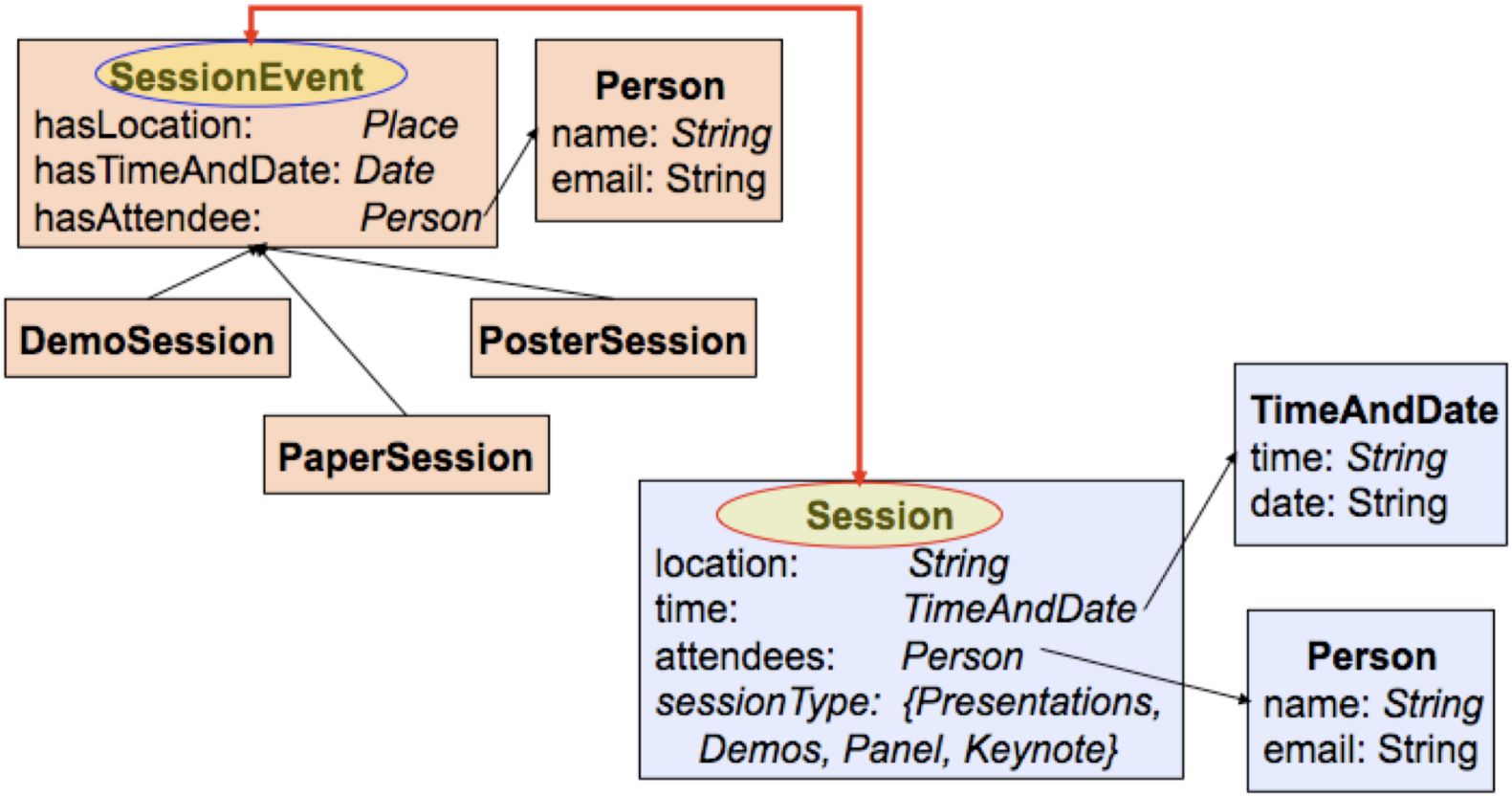
  closeTo(http://amaz.com/road, http://ign.com/tronçon-route), **0.8**

- A mapping can be complex:
  realisation(y,concat(x.firstName,x.Name))
  ← 0.9 film(y) Λ realised(x,y)

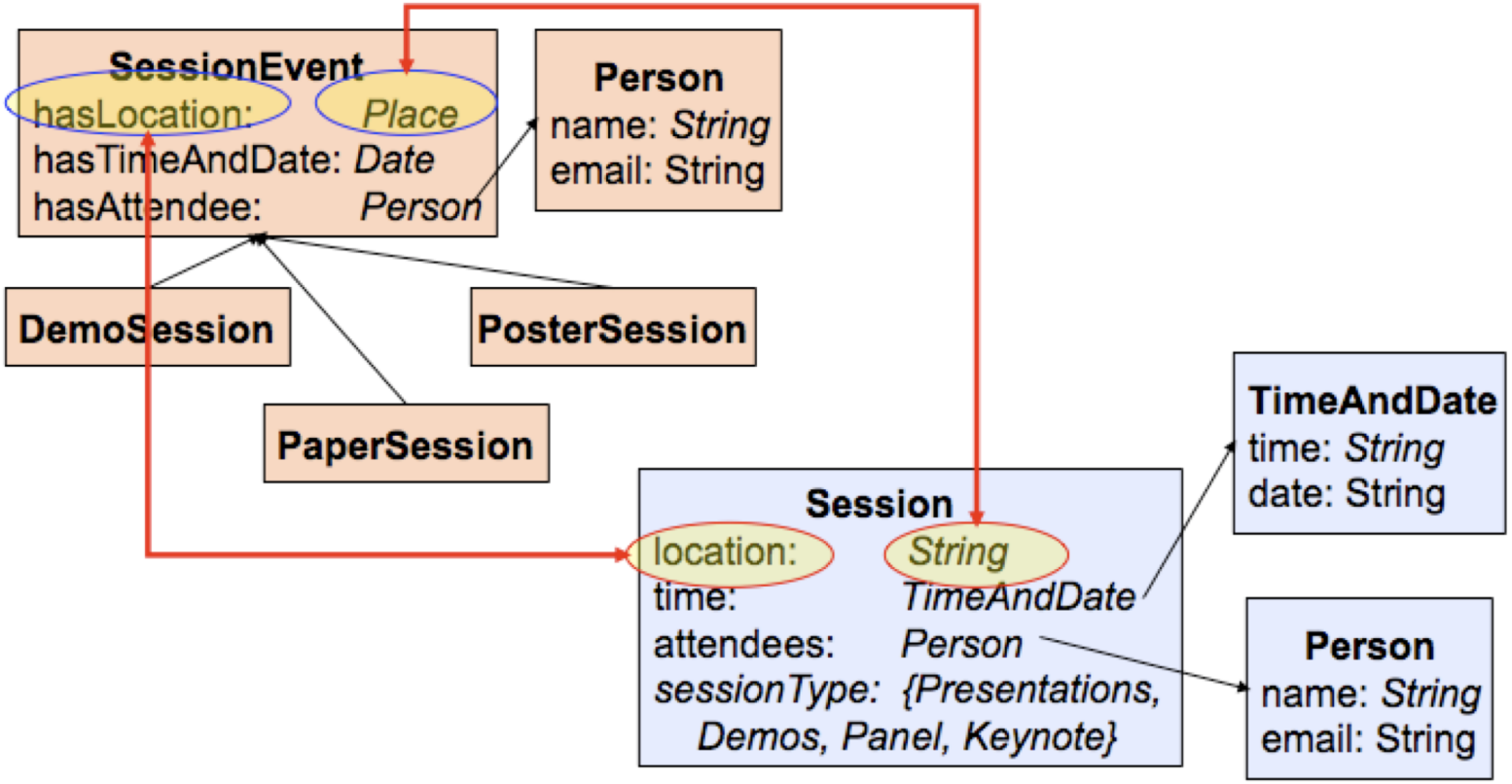
- An alignment can be **1-1** (« **one to one** » mapping) bijective relation.

- Some alignment approaches are not symetric :
  f(O1,O2) <> f(O2,O1)

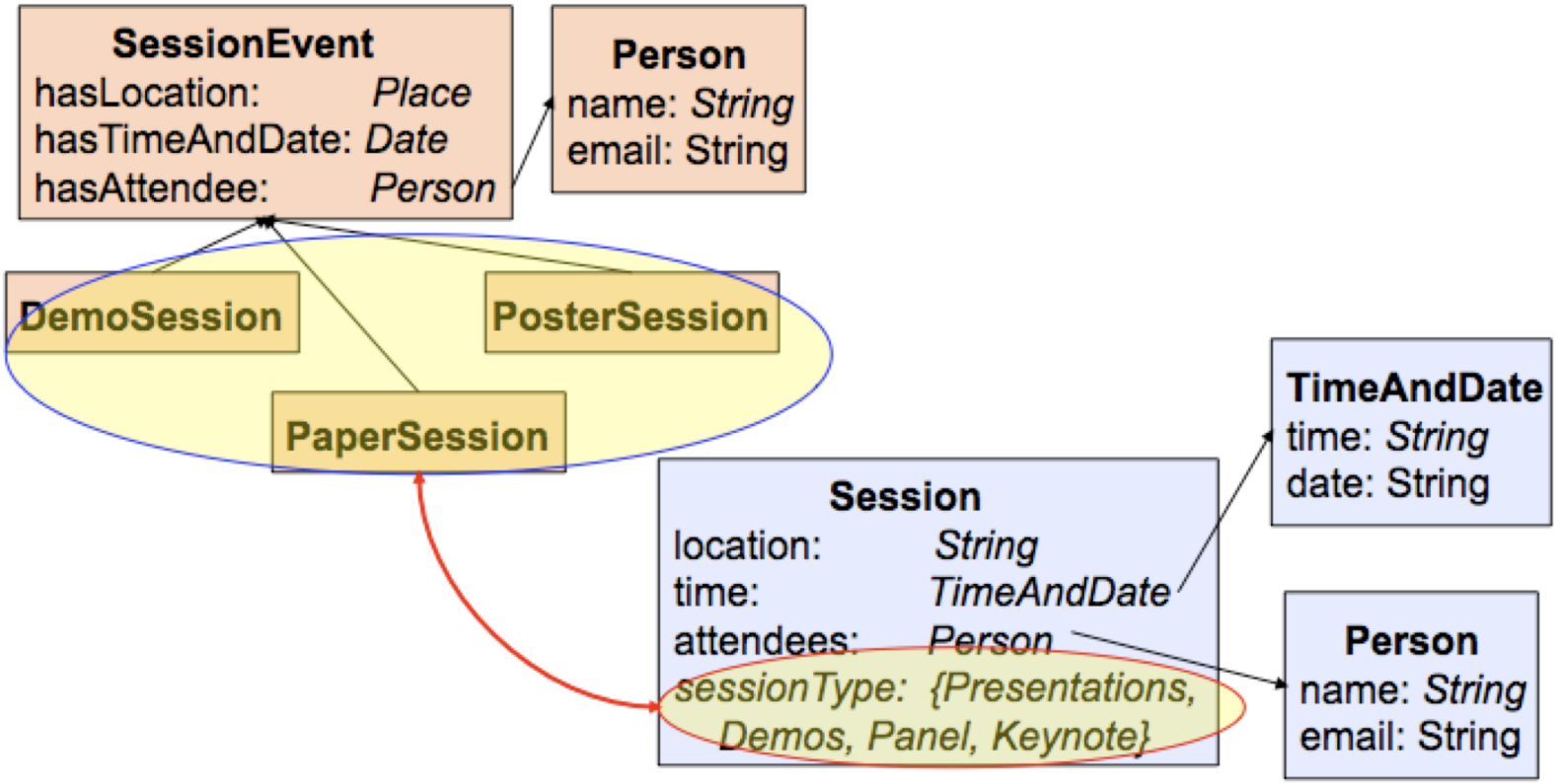
DIFFICULTIES



DIFFICULTIES



DIFFICULTIES



What can be taken into account to decide?

- **Terminological Information** (concept labels, alt-labels, property names), coments, names of other linked entites, …)

  ➔ Usual similarity measures can be used: token-based (n-grams, jaccard, or edit-based (levenstein, jaro-winkler) …

  → Other linguistic informations

  Examples:

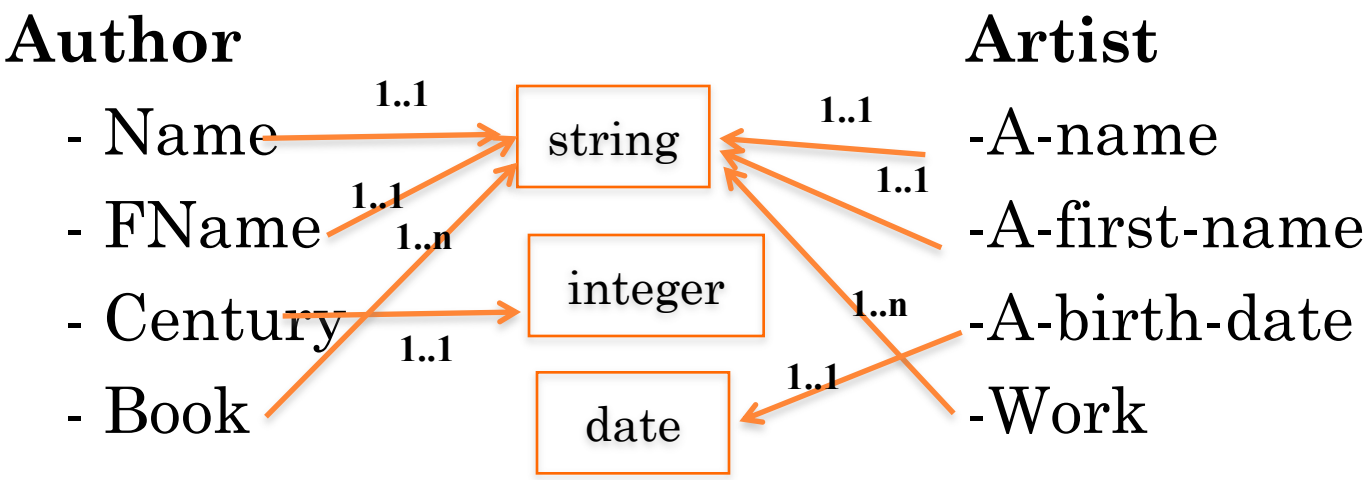
  Stop-words (of, le, for, the,..)

  Word weights can vary depending on its syntactic function (example : an adjective is less important than a nominal … *national road/ national organization*)

What can be taken into account to decide?

- **Ontology Structure** :

  Internal Structure (datatype properties, range, cardinalities)



SIMILARITY OF INTERNAL STRUCTURES

Compatibility measures can be defined [Valchev et Euzenat 97, 04]

→ To compare the range (table)

compatibility (integer,date)=0.5

compatibility(integer, number)=0.9

→ To compare cardinalities

As it can be done for an XML DTD : compatibility (*, +)=0.9

In OWL: mincardinality b /maxcardinality e

Sim([b1 e1], [b2 e2])=0 if b2>e1 ou b1>e2

else = (min(e1,e2)-max(b1,b2))/(max(e1,e2)-min(b1,b2))

SIMILARITY OF EXTERNAL STRUCTURE

External Structure (related concepts) :

Hypothesis: the more 2 concepts are similar, the more their linked concepts are similar.

Given a property r (usually r=subclassOf)

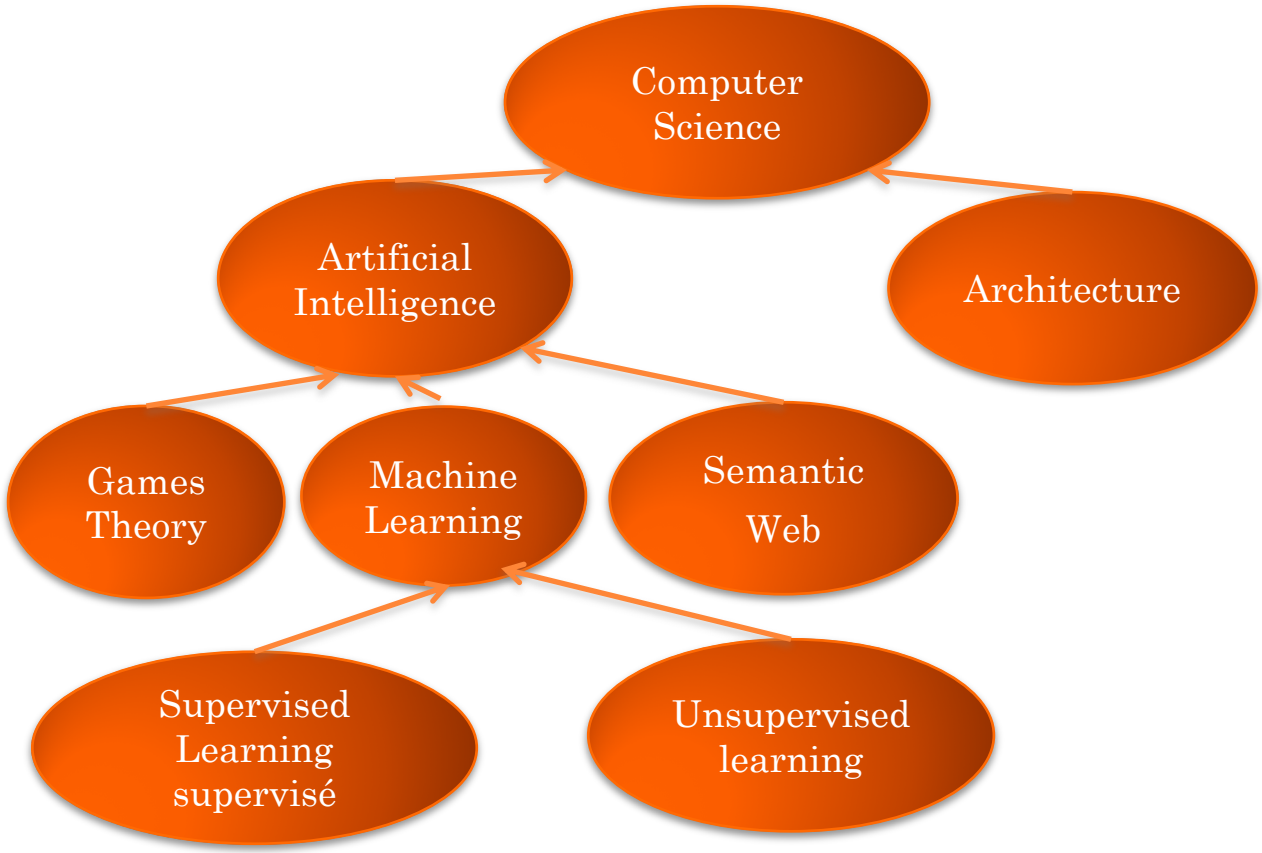
r : concepts that are directly linked using r

r+: concepts that belong to the transitive closure (w.r.t the property r)

r-1; more general concepts

r! : leaves that belong the transitive closure of subclassOf.

EXAMPLE


Computer Science
Artificial Intelligence
Architecture
Games Theory
Machine Learning
Semantic Web
Supervised Learning supervisé
Unsupervised learning

EXAMPLE

- With r= subclassOf

subclassOf(Artificial Intelligence)={Game theory, Machine Learning, Semantic web}

subclassOf+(Artificial Intelligence)={Game theory, Machine Learning, Supervised Learning, Unsupervised learning, Semantic web}

subclassOf-1(Artificial Intelligence)={Computer Science}

subclassOf!(Artificial Intelligence)={Game theory, Supervised Learning, Unsupervised learning, Semantic web}

EXTERNAL RESOURCE

- Two concepts can be compared using an external ontology or taxonomy they also belong to (or they have been found in).

Similarity Measure: Wu et Palmer [Wu &al. 94]

$$Sim(C1, C2) = 2 * depth(C) / (depthc(C1) + depthc(C2))$$

where C is the most specific common ancester (LCS) of C1 and C2.

EXTERNAL RESOURCE

- Examples

Sim(Supervised Learning, Unsupervised Learning)

= 2* Depth(Machine Learning)/Depthc(Supervised Learning)+Depthc(Unsupervised Learning)

= (2*3)/(4+4)=6/8=0.75

Sim(Art Intelligence, Architecture)

=(2*1)/(2+2)=0.5

Sim(Supervised Learning, Architecture)

= (2*1)/(4+2)=0.33

EXTERNAL RESOURCE

Results that can be obtained using Wordnet [Madche et Zacharias 2002]

	illustrator	**author**	**creator**	**person**
illustrator	1	0.5	0.67	0.4
author		1	0.67	0.4
creator			1	0.67
person				1

EXTERNAL RESOURCE

- Similarity measure of [Resnik, 1995]: based on the informational content (CI).

This measure exploits the ontology and its instances.

CI allow to take into account the frequency of the concept in the studied domain.

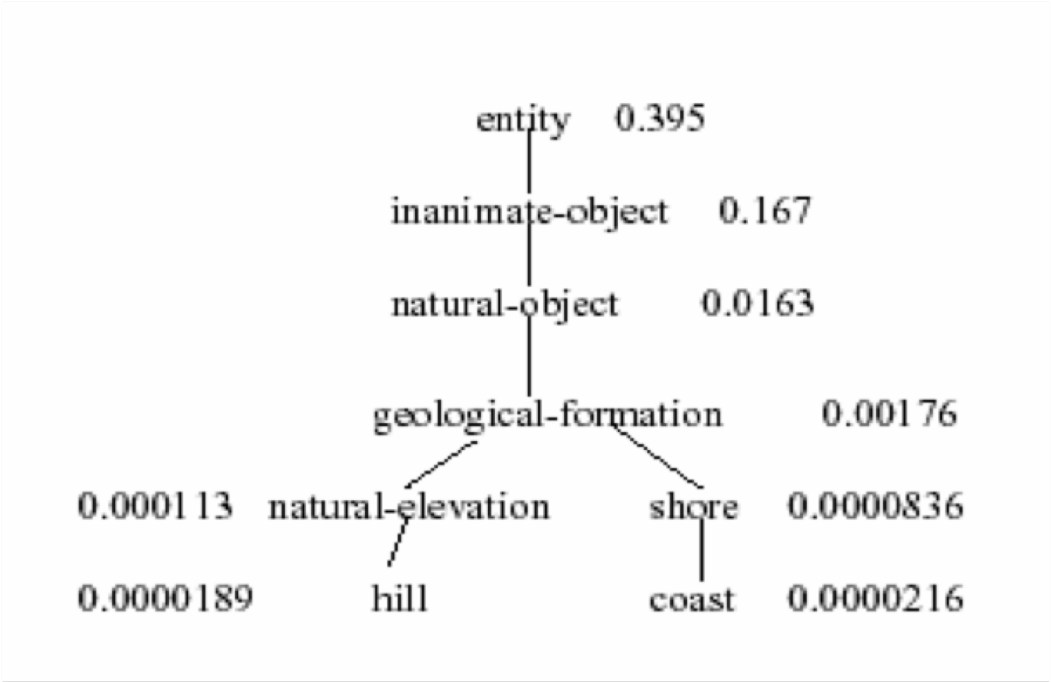
$CI(c) = - log(P(c))$

where P(c) is the probability to find an instance of c (number of instances(c)/N where N is the number of concepts).

EXTERNAL RESOURCE

- The semantic similarity of two concepts c1, c2 is here the informational content they share in their LCS .

P(C) computed on wordnet



HOW SEMANTICS CAN BE USED?

- Infer new alignments thanks to A and a reasoning step

Example: (Description Logic: [Bouquet et al 2006])

Micro-company=Company$\Pi \leq$5 employee

SME = Firm $\Pi \leq$10 associate

If we know in a first alignment result A that:

Company = Firm, associate [employee,

Micro-company [SME is inferred

- Detect inconsistencies when A is used (filtering step) … Example ?

How to use concept instances (extensional information)?

Naive way: compute the intersection of two concept extensions.

Equivalence if $C1 \cap C2 = C1 = C2$

Subsumption if $C1 \cap C2 = C1$

Disjunction if $C1 \cap C2$ is empty

And more generally, define and use a similarity measure based on the size of this intersection.

Example : (jaccard) $(C1 \cap C2)/(C1 \cup C2)$

[RIMOM14] [PARIS12]

Extensional Approaches

- Problem: how do you know that two instances are the same.

Possible if instance URI are the same or if the distinct but identical instances share same document url that describe them (ex: DBPedia/Yago).

……. If not, a linking method is needed (that detect identity links) .

STRATEGIES

- Computation of the similarity scores then selection of the proposed mappings. [AML15]
- Combinaison of different strategies in //

  [Glue 2004]
- Sequential application of different strategies [Taxomap 2008] [PARIS2012]
- Filtering
- Ontology partionning [Falcon 2008]

MAPPING SELECTION

Thresholds are defined ….

- Absolute threshold (sa) : select all mappings such that the similarity score is >sa
- Relative threshold (sd) : select all mappings such that the similarity score >MaxScore – sd
- Rate (n%) : select the n% best scores in A.

MAPPING SELECTION

- Example:

	book	translator	editor	author
product	.84	0	.9	.12
provider	.12	0	.84	.60
artist	.60	.05	.12	.84

Absolute threshold 0.7: (product, book), (product, editor) (provider,editor), (artist, author)

Relative Threshold delta 0.3: (product, book), (product, editor) (providor,editor), (providor,author), (artist, book), (artist, author)

Rate =30% (4): (product, book), (product, editor), (provider, editor), (artist, author)

MAPPING SELECTION

- When a 1-1 alignment is wanted, a choice is necessary

Greedy algorithm (best score at each iteration).

Stable marriage problem (local optimum):

Given 2 entity (concept) sets E et E' and a similarity function sim : E x E' → [0,1], selection of an alignment A s.t. for all (e1,e2) in A, and for all (e3,e4) in A, we have:

$$sim(e1,e3)+sim(e2,e4) \geq sim(e1,e2)+sim(e3,e4)$$

MAPPING SELECTION

Maximum weight graphs (global optimum):

Given 2 entity (concept) sets E et E' and a similarity function sim : E x E' → [0,1], selection of an alignment A s.t.:

$$\Sigma_{(ei,ej) \text{ in } A} \geq \Sigma_{(ei,ej) \text{ in } A'}$$

MAPPING SELECTION

- Example :

	book	translator	editor	author
product	.84	0	.9	.12
provider	.12	0	.84	.60
artist	.60	.05	.12	.84

Greedy : (product,editor), (artist, author), (provider, book) (1.96)

Local: replace with (artist, book), (provider, author) (2.1)

Global : (product,book), (provider, editor), (artist, author) (2.52)

OAEI Competition

- Ontology Alignment Evaluation Initiative (OAEI)
  → every year since 2006

- Different Tracks (ontology size, formalism, domain, instances or not)

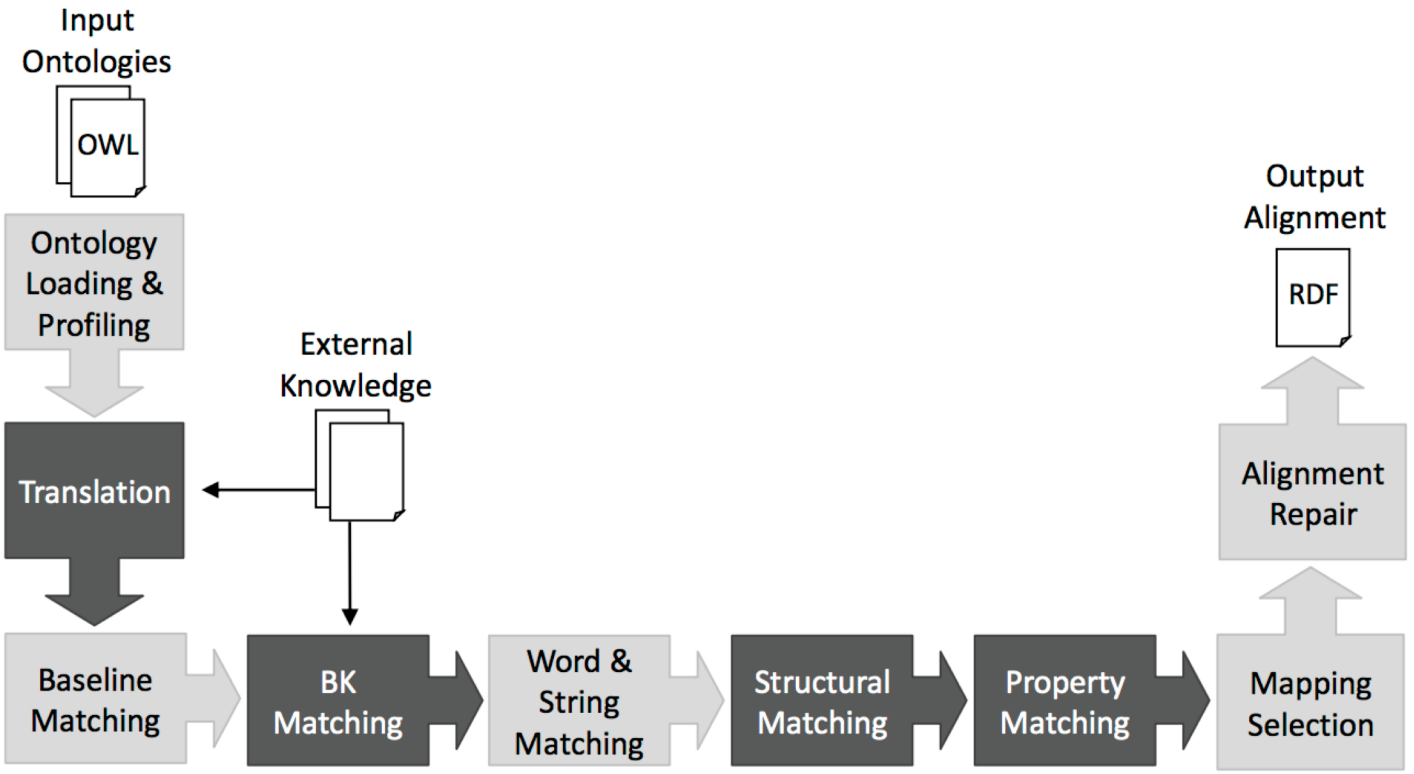
- Various results
  (< 50% of precision/recall,
  to 100% when ontology are highly similar)

Matcher	Runtime	Size	Precision	F-measure	Recall	Recall+	Coherent
AML	47	1493	0.95	0.943	0.936	0.832	√
CroMatcher	573	1442	0.949	0.925	0.902	0.773	-
XMap	45	1413	0.929	0.896	0.865	0.647	√
LogMapBio	758	1531	0.888	0.892	0.896	0.728	√
FCA_Map	117	1361	0.932	0.882	0.837	0.578	-
LogMap	24	1397	0.918	0.88	0.846	0.593	√
LYAM	799	1539	0.863	0.869	0.876	0.682	-
Lily	272	1382	0.87	0.83	0.794	0.515	-
LogMapLite	20	1147	0.962	0.828	0.728	0.288	-
StringEquiv	-	946	0.997	0.766	0.622	0.000	-
LPHOM	1601	1555	0.709	0.718	0.727	0.497	-
Alin	306	510	0.996	0.501	0.335	0.0	√
DKP-AOM-Lite	372	207	0.99	0.238	0.135	0.0	√
DKP-AOM	379	207	0.99	0.238	0.135	0.0	√

Table 6. Comparison, ordered by F-measure, against the reference alignment, runtime is measured in seconds, the "size" column refers to the number of correspondences in the generated alignment.

Anatomy ontologies – Results OAEI 2016

AML


Input
Ontologies
OWL
Ontology
Loading &
Profiling
External
Knowledge
Translation
Baseline
Matching
BK
Matching
Word &
String
Matching
Structural
Matching
Property
Matching
Mapping
Selection
Alignment
Repair
Output
Alignment
RDF

OAEI 2016

Ressource :
http://disi.unitn.it/~pavel/om2016/papers/oaei16_paper0.pdf

- Number of participants?
- Type of tracks? Types of correspondences?
- Are all the approaches tested on all tracks ?
- Table 5 (track anatomy) : size of the ontologies ?

Precision/recall?

Are there many approaches that check the result consistency
Best approach (best F-Mesure)?

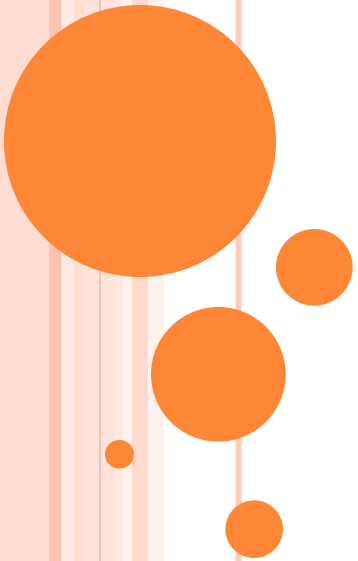
Challenges

- How to interact with user experts during the alignment process (validation, updates, suppression),
- How to explain discovered mappings
- How to scale
- How to mix data linking and ontology alignment processes
- How confidence degrees that can be associated to mappings can be used.
- Ontology evolution → How to update alignments efficiently

REFERENCES

- "Ontology Matching" by Euzenat and Shvaiko , springer, 2007.
- AML (AgreementMakerLight) : D. Faria, C. Martins, A. Nanavaty, D. Oliveira, B. S. Balasubramani, A. Taheri, C. Pesquita, F. M. Couto, and I. F. Cruz. AML results for OAEI 2015. In *Ontology Matching Workshop*. CEUR, 2015.
- Proceedings des conferences ISWC, ESWC, SIGMOD, WWW, EKAW, K-Cap.
- *Journal of web semantics, Journal on data semantics.*
- *http://www.ontologymatching.org*

Ontology Enrichment for Data Integration



Data Linking approaches

Different criteria can be used to distinguish data linking approaches [FNS11]

- **Instance-based** approaches: exploit property values to link 2 instances / **Graph-based** approaches: propagate similarities, decisions
- **Supervised** approaches : exploit labeled training data given by an expert / **Unsupervised** approaches
- **Knowledge based** approaches : exploit ontology axioms (eg. functional properties, disjunctions) or expert rules
- **Logical** or **Numerical** approaches

Data Linking using rules

- Rules
  - Logical Rules
    - SSN(p1, y) ∧ SSN(p2, y) ➔ sameAs(p1, p2)
  - Complex Rules
    - max(jaccard(Name(p1, n); Name(p2, m);jarowinkler(address(p1, x); address(p2, y))) > 0.8 ➔ sameAs(p1, p2)

Data Linking using rules

- Rules use discriminative properties => keys
  - Not easy to be declared by expert
    - {SSN}, {ISBN} easy
    - {Name, dateOfBirth, BornIn} **is it a key?**
  - Erroneous keys can be given by experts
  - As many keys as possible

- **Automatic discovery of keys**

OWL2 Key

- OWL (Web Ontology Language)
  - Declaration of classes, properties, axioms (subsumption, equivalence, etc.)

- **OWL2 Key for a class**: a combination of properties that uniquely identify each instance of a class
  - hasKey(CE (OPE_1 ... OPE_m) (DPE_1 ... DPE_n))

$$\forall X, \forall Y, \forall Z_1, \ldots, Z_n, \forall T_1, \ldots, T_m \wedge ce(X) \wedge ce(Y) \bigwedge_{i=1}^{n} (ope_i(X, Z_i) \wedge ope_i(Y, Z_i))$$

$$\bigwedge_{i=1}^{m} (dpe_i(X, T_i) \wedge dpe_i(Y, T_i)) \Rightarrow X = Y$$

hasKey(Book(Author) (Title)) means:
$Book(x_1) \wedge Book(x_2) \wedge Author(x_1, y) \wedge Author(x_2, y) \wedge Title(x_1, w) \wedge Title(x_2, w)$ ➔ $sameAs(x_1, x_2)$

How to discover keys ? Assumptions

- **Open World Assumption (OWA)**

  what is not in the data is not false

- **How to discover keys in Open World Assumption (OWA)??**

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2,i3
i2	Tompson	Maria	
i3	David	George	i2, i4
i4	Solgar	Michel	

i1 =?= i2 =?=i3 =?=i4
hasFriend(i1,i4) …. ?
hasFriend(i2, i3) …. ?
firstName(i1, Elodie) … ?

Assumptions

- **Unique Name Assumption (UNA)**: different identifiers refer to distinct real world objects
  - i1 <> i2 <> i3 <> i4

Fulfilled in datasets extracted from RDB, YAGO

- **Uniform Vocabulary**: Syntactically different ➔semantically different in one dataset
  - "UK" and "United Kingdom" never in the same dataset

Optimistic keys

- **Optimistic key**: set of properties that has a unique value for every instance described by this property set in the data
  -

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2, i3 ~~**i4**~~
i2	Tompson	Maria	~~**i1, i3, i4**~~
i3	David	George	i2, i4 ~~**i1**~~
i4	Solgar	Michel	~~**i1, i2, i3**~~

- **Keys**: {hasFriend, lastName}, {firstName}

Algorithms

- **Naive automatic way to discover keys**
  - Examine all the possible combinations of properties
  - Scan all instances for each candidate key

  Example: Class described by 15 properties ➔ 2^{15} = 32767 candidate keys

- Discover keys efficiently by:
  - Reducing the combinations
  - Partially scanning the data

SAKEY - ALGORITHM

- Non key discovery first (Partially scan the data)

Key

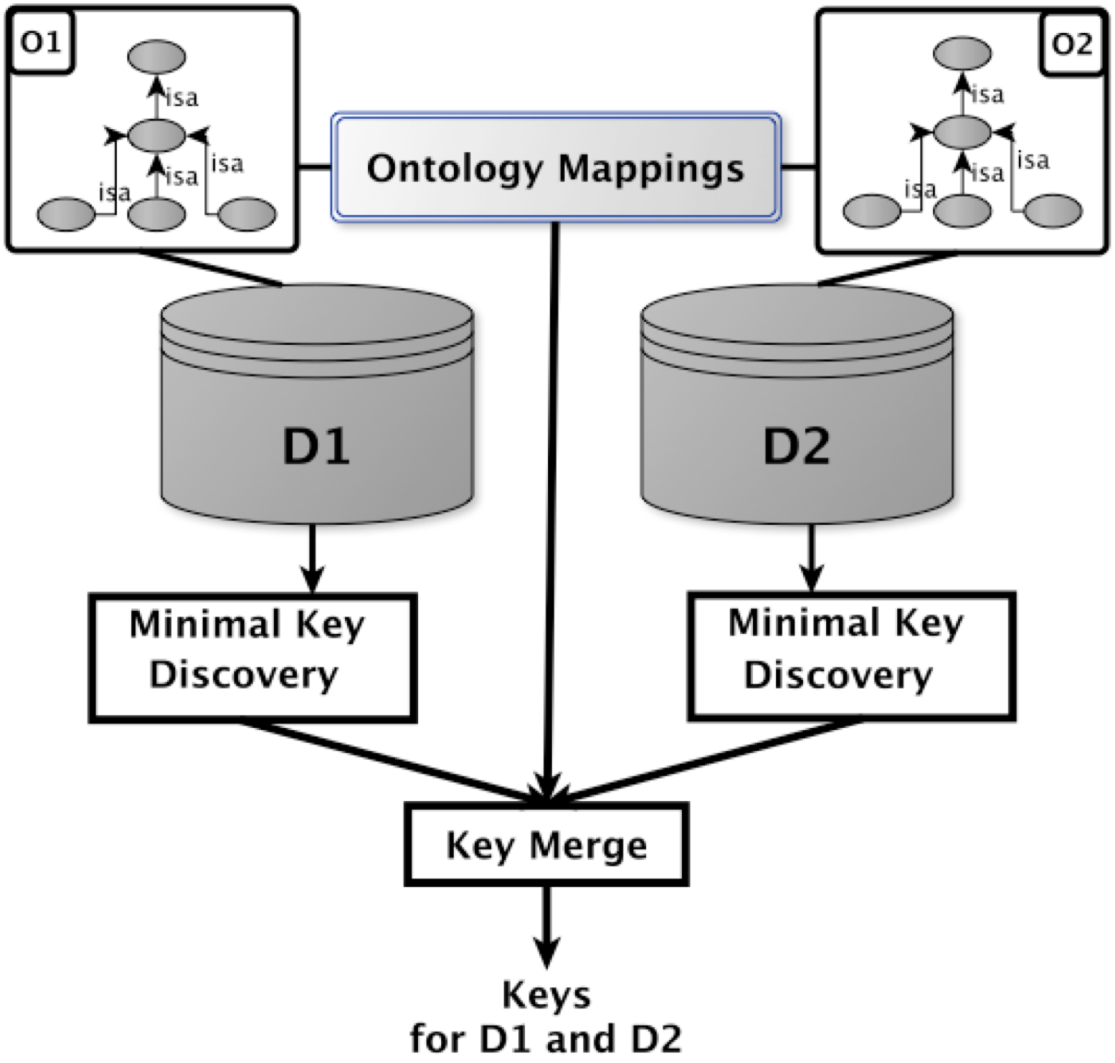
Non key

	museumName	...	museumAddress	inCountry
Museum1	"Archaeological Museum"		"44 Patission Street"	"Greece"
Museum2	"Pompidou"		------	**"France"**
Museum3	"Musée d'Orsay"		"62, rue de Lille"	**"France"**
Museum4	"Madame Tussauds"		"Marylebone Road"	"England"
Museum5	"Vatican Museums"		"Piazza San Giovanni"	"Italy"
Museum6	"Deutsches Museum "		"Museumsinsel 1"	"Germany"
Museum7	"Olympia Museum"		"Archea Olympia"	"Greece"
Museum8	"Dalí museum"		"1, Dali Boulevard"	"Spain"

- All the sets of properties that are not subsets of maximal non keys are keys
  - **Example:** class described by the properties p1, p2, p3, p4

Maximal non key = {{p1, p2}} → keys = {{p3}, {p4}}

Objective


O1
isa
isa
isa
isa
Ontology Mappings
O2
isa
isa
isa
D1
D2
Minimal Key Discovery
Minimal Key Discovery
Key Merge
Keys for D1 and D2

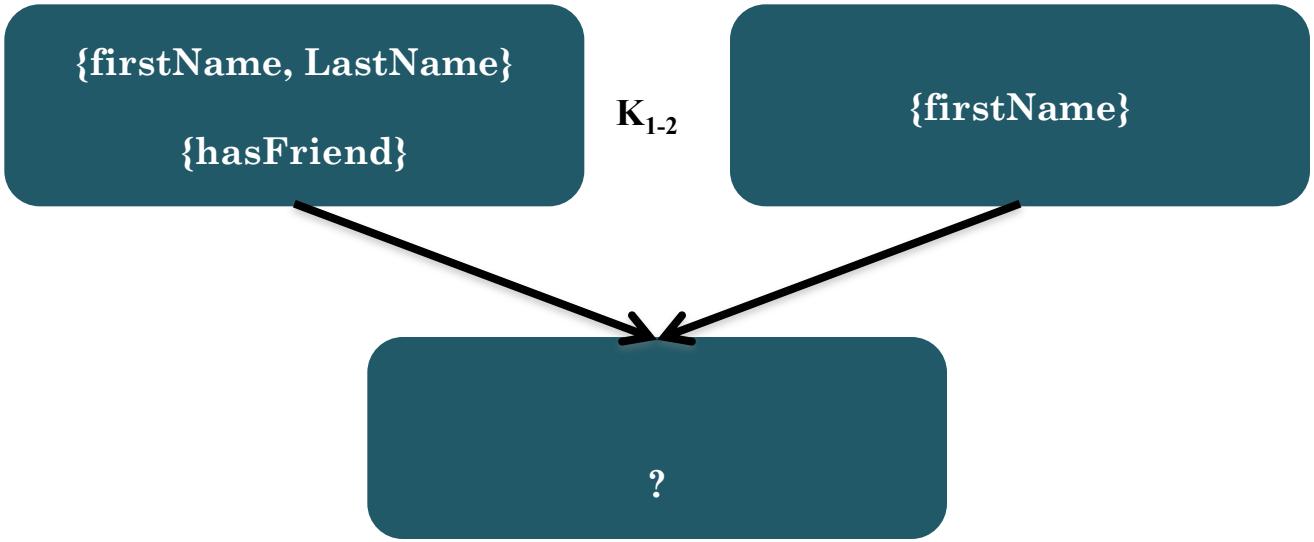
KEY DERIVATION [SBNHR06]

Intuition: All the sets of properties not belonging to maximal non keys are keys

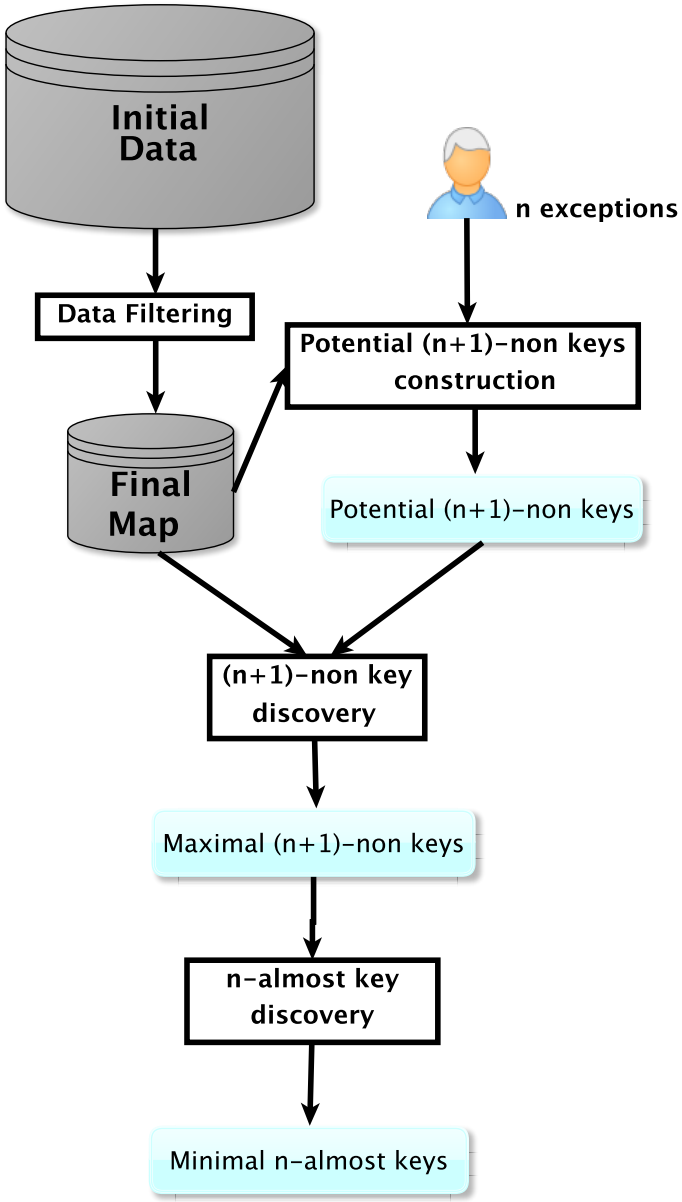
- Given the complete set of maximal non keys
  - Compute complement sets of each non key
  - Compute the Cartesian product of complement sets
  - Simplify the discovered keys

KEY MERGE

- **Goal**: Keys valid in both datasets
  - More sure keys
- **Intuition**: Computation of Cartesian product of K_1 and K_2 sets of keys
  - Keep only minimal keys



SAKey - General Architecture



N-ALMOST KEYS

- **Exception of a key:** an instance that shares values with another instance for a given set of properties *P*
  - f1, f2, f3, f4 are exception for {Actor}

	Name	Actor	Director	ReleaseDate	Website	Language
f1	"Ocean's 11"	**"B. Pitt"** "J. Roberts"	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
f2	"Ocean's 12"	**"B. Pitt"** "G. Clooney" "J. Roberts"	"S. Soderbergh" "R. Howard"	"2/5/04"	www.oceans12.com	---
f3	"Ocean's 13"	**"B. Pitt"** "G. Clooney"	"S. Soderbergh" "R. Howard"	"30/6/07"	www.oceans13.com	---
f4	"The descendants"	"N. Krause" "G. Clooney"	"A. Payne"	"15/9/11"	www.descendants.com	"english"
f5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"
f6	"Ocean's 12"	---	"R. Howard"	"2/5/04"	---	---

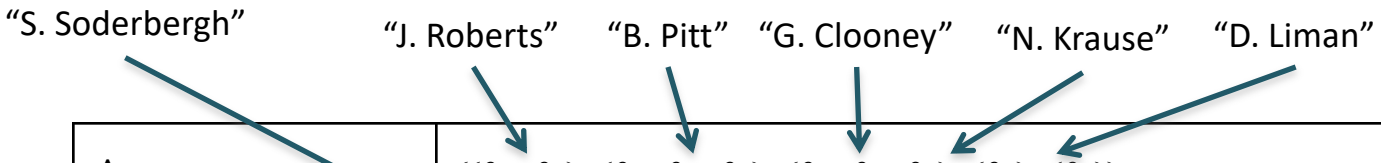
N-ALMOST KEYS

- ***n*-almost key:** a set of properties where $|E_P| \leq n$
  - {Actor} is a 4-almost key

- ***n*-non key:** a set of properties where $|E_P| \geq n$
  - Using all the maximal *n*-non keys we can derive all the minimal (*n*-1)-almost keys

Example: All the sets of properties that are not maximal 5-non keys are 4-almost keys

N-NON KEY DISCOVERY: INITIAL MAP



Actor	{{f1, f2}, {f1, f2, f3}, {f2, f3, f4}, {f4}, {f5}}
Director	{{f1,f2,f3}, {f2, f3, f6}, {f4}}
ReleaseDate	{{f1}, {f2, f6}, {f3}, {f4}, {f5}}
Name	{{f1}, {f2, f6}, {f3}, {f4}, {f5}}
Language	{{f4, f5}}
Website	{{f1}, {f2}, {f3}, {f4}, {f5}, {f6}}

N-NON KEY DISCOVERY: FINAL MAP

Actor	{{f1, f2, f3}, {f2, f3, f4}}
Director	{{f1,f2,f3}, {f2, f3, f6}}
ReleaseDate	{{f2, f6}}
Name	{{f2, f6}}
Language	{{f4, f5}}

N-NON KEY DISCOVERY

- Actor
  - {f1, f2, f3} ∪ {f2, f3, f4} = {f1, f2, f3, f4} => 4-non key
- Composite *n*-non keys
  - Intersections between sets of different properties

Actor	{{f1, f2, f3}, {f2, f3, f4}}
Director	{{f1,f2,f3}, {f2, f3, f6}}
ReleaseDate	{{f2, f6}}
Name	{{f2, f6}}
Language	{{f4, f5}}

Data Linking using almost keys

- **Goal**: Compare linking results using almost keys with different *n*

- Evaluation of linking using
  - Recall
  - Precision
  - F-Measure

- Datasets
  - OAEI 2010
  - OAEI 2013

Data Linking using almost keys

- **Goal**: Compare linking results using almost keys with different *n*

- Evaluation of linking using
  - Recall
  - Precision
  - F-Measure
- Datasets : OAEI 2010 , OAEI 2013
- Conclusion
  - Use keys is almost the same or better than using expert keys, Linking results using *n*-almost keys are the better than using keys

Data Linking using almost keys

- OAEI 2013 - Person

	Almost keys	Recall	Precision	F-Measure
0-almost key	{BirthDate, award}	9.3%	100%	17%
2-almost key	{BirthDate}	32.5%	98.6%	49%

# exceptions	Recall	Precision	F-measure
0, 1	25.6%	100%	41%
2, 3	47.6%	98.1%	64.2%
4, 5	47.9%	96.3%	63.9%
6, ..., 16	48.1%	96.3%	64.1%
17	49.3%	82.8%	61.8%

Scalability of SAKey

- $n > 1$
  - Class DB:NaturalPlace

n	**Runtime**	**# *n*-non keys**
1	61s	298
50	55s	118
100	58s	78
200	56s	53
300	59s	45
400	56s	41

CONDITIONAL KEYS [VICKEY 17]

- **Conditional key:** a key, valid for instances satisfying a specific condition
  - **Condition part:** pairs of property and value

  Eg. {Lab=INRA}, {Gender=Male}, {Gender=Female ^ Lab=INRA} etc.
  - **Key part:** a set of properties

Instances of the class Person

	FirstName	LastName	Gender	Lab	Nationality
instance1	Claude	Dupont	Female	Paris-Sud	France
instance2	Claude	Dupont	Male	Paris-Sud	Belgium
instance3	Juan	Rodríguez	Male	**INRA**	Spain, Italy
instance4	Juan	Salvez	Male	**INRA**	Spain
instance5	Anna	Georgiou	Female	**INRA**	Greece, France
instance6	Pavlos	Markou	Male	Paris-Sud	Greece
instance7	Marie	Legendre	Female	**INRA**	France

{LastName} is a *key* under the *condition* **{Lab=INRA}**

VICKEY: MINING EFFICIENTLY CONDITIONAL KEYS

- **A key is also a conditional key under any condition**
  - {LastName, Gender} is a *key*
  - {LastName, Gender} is a *key* under the *condition* {Lab=INRA}

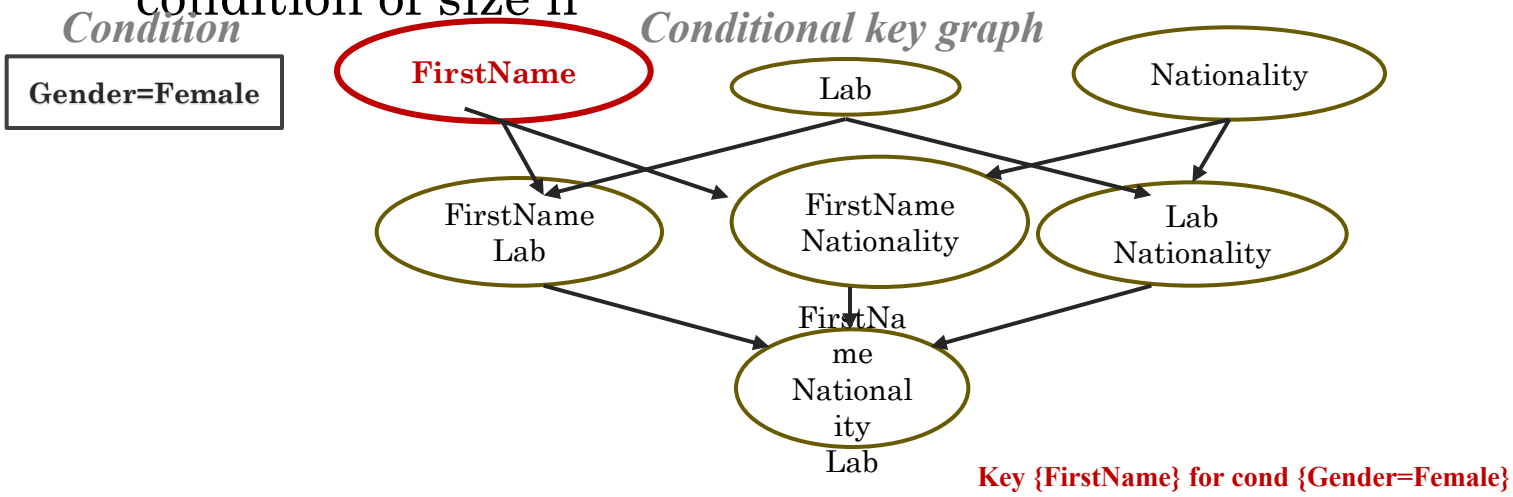
Instances of the class Person

	FirstName	LastName	Gender	Lab	Nationality
instance1	Claude	Dupont	Female	Paris-Sud	France
instance2	Claude	Dupont	Male	Paris-Sud	Belgium
instance3	Juan	Rodríguez	Male	INRA	Spain, Italy
instance4	Juan	Salvez	Male	INRA	Spain
instance5	Anna	Georgiou	Female	INRA	Greece, France
instance6	Pavlos	Markou	Male	Paris-Sud	Greece
instance7	Marie	Legendre	Female	INRA	France

Conditional keys can be found in non-keys

Conditional key graph exploration

- Given a non-key
  - **Step 1: Discover all minimal conditional keys with condition of size 1**
  - Step 2: Discover all minimal conditional keys with condition of size 2
  - …Step n: Discover all minimal conditional keys with condition of size n



Data Linking - VICKEY vs. SAKey[1]

Class		Recall	Precision	F-Measure	
Actor	Keys[1]*	0.27	0.99	**0.43**	
	Conditional keys**	0.57	0.99	0.73	**x 1.75**
	Keys[1]+Conditional keys	0.6	0.99	**0.75**	
Album	Keys[1]	0	1	**0.00**	
	Conditional keys	0.15	0.99	0.26	**x 869**
	Keys[1]+Conditional keys	0.15	0.99	**0.26**	
Film	Keys[1]	0.04	0.99	**0.08**	
	Conditional keys	0.38	0.96	0.54	**x 7.1**
	Keys[1]+Conditional keys	0.39	0.98	**0.55**	
Museum	Keys[1]	0.12	1	**0.21**	
	Conditional keys	0.25	1	0.40	**x 2.19**
	Keys[1]+Conditional keys	0.31	1	**0.47**	

**Keys[1] from SAKey*

***Conditional keys from VICKEY*

RUNTIME RESULTS - VICKEY

Class*	#Triples	#Instances	#Properties	#NonKeys	VICKEY	#Conditional Keys
Actor	57.2k	5.8k	71	137	4.52m	311
Album	786.1k	85.3k	39	68	1.53h	304
Book	258.4k	30.0k	51	95	11.84h	419
Film	832.1k	82.6k	74	132	1.37h	185
Mountain	127.8k	16.4k	58	47	2.86m	257
Organization	1.82M	178.7k	553	3221	26.32h	28
Scientist	258.5k	19.7k	73	309	27.67m	582

**All used classes are obtained from DBpedia*

Conclusion

- **SAKey**: almost key discovery approach in RDF data
  - Erroneous data, duplicates
  - *n*-almost keys: keys with at most *n* exceptions
  - Scalable thanks to:
    - filtering and pruning strategies
  - Experiments show the scalability of SAKey and the relevance of almost keys in data linking
- **VICKEY**: Conditional key discovery approach
- Can improve the results

Challenges

- Can the number of exceptions be set automatically?
- Define different merging strategies
  - How to merge almost keys with different n values?
  - How to exploit mappings, subsumptions etc?
  - Should all the keys participate in the merge?
- Key update when data evolve

References

- **[SBHR06]** Yannis Sismanis, Paul Brown, Peter J. Haas, and Berthold Reinwald. Gordian: efficient and scalable discovery of composite keys. In *Proceedings of the 32nd International conference Very Large Data Bases (VLDB),* VLDB '06, pages 691–702. VLDB Endowment, 2006.
- **[ADS12]** Manuel Atencia, Jérôme David, and François Scharffe. Keys and pseudo- keys detection for web datasets cleansing and interlinking. In *EKAW,* pages 144–153, 2012.
- **[HJAQR+13]** A. Heise, Jorge-Arnulfo, Quiane-Ruiz, Z. Abedjan, A. Jentzsch, and F. Nau- mann. Scalable discovery of unique column combinations. *VLDB,* 7(4):301– 312, 2013.
- **[SAKey14]** Danai Symeonidou, Vincent Armant, Nathalie Pernelle, Fatiha Saïs. SAKey: Scalable Almost Key discovery in RDF data. 13th International Semantic Web Conference (ISWC 2014).
- **[VICKEY17]** Danai Symeonidou, Luis Galarraga, Nathalie Pernelle, Fatiha Saïs**,** Fabian Suchanek, VICKEY, ISWC 2017.