From data to knowledge: some approaches for data linking, data fusion and key discovery

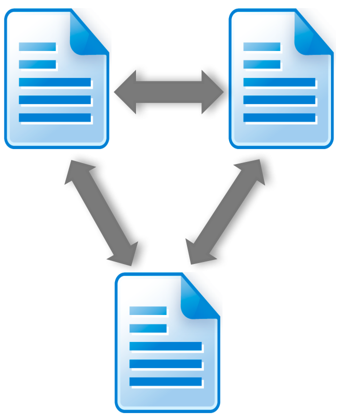
FATIHA SAÏS

LRI, CNRS & PARIS SUD UNIVERSITY

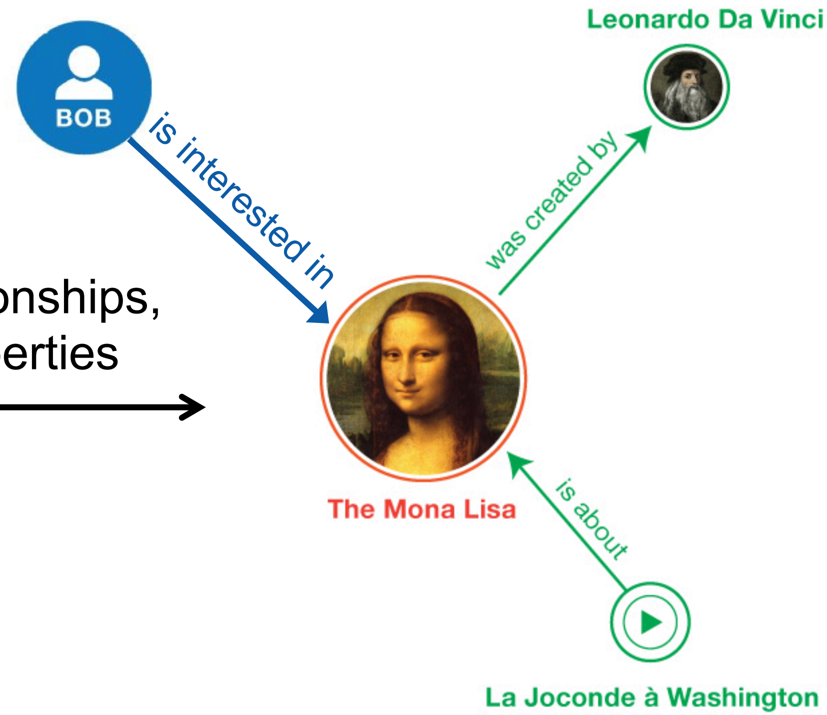
TETIS SEMINAR- 21 FEBRUARY 2017

FROM THE WWW TO THE WEB OF DATA

- applying the principles of the WWW to data



data is relationships,
not only properties



LINKED DATA PRINCIPLES

① **Use HTTP URIs as identifiers for resources**

→ so people can look up the data

② **Provide data at the location of URIs**

→ to provide data for interested parties

③ **Include links to other resources**

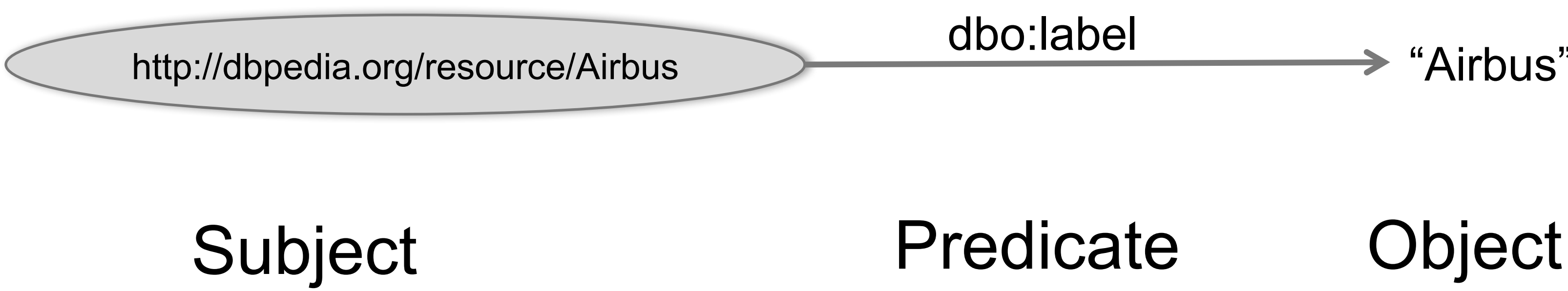
→so people can discover more things

→bridging disciplines and domains

→the more linked resources, the more one can find out

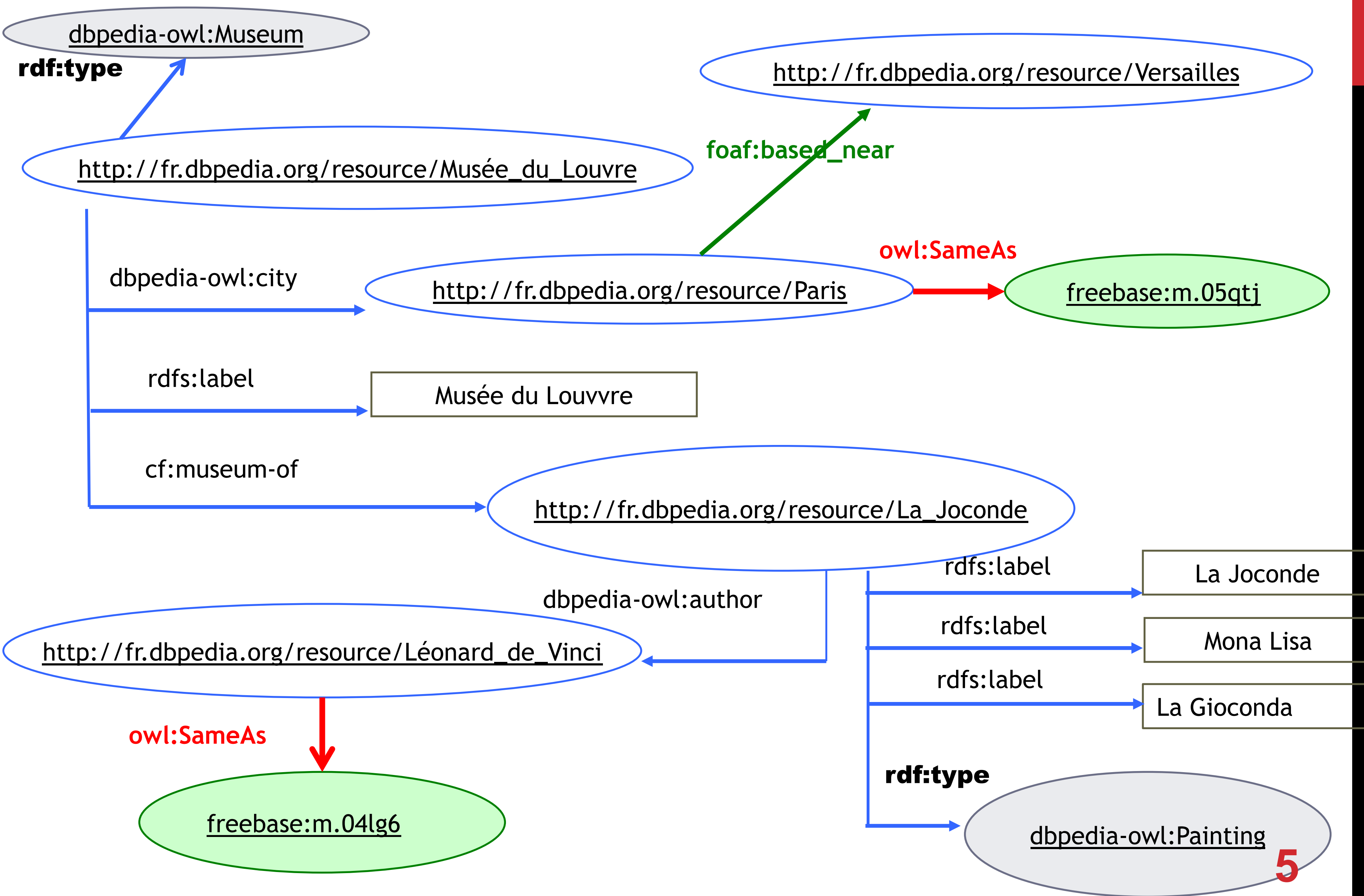
RDF – RESOURCE DESCRIPTION FRAMEWORK

- **Statements of < subject predicate object >**



… is called a triple

Data linking: example



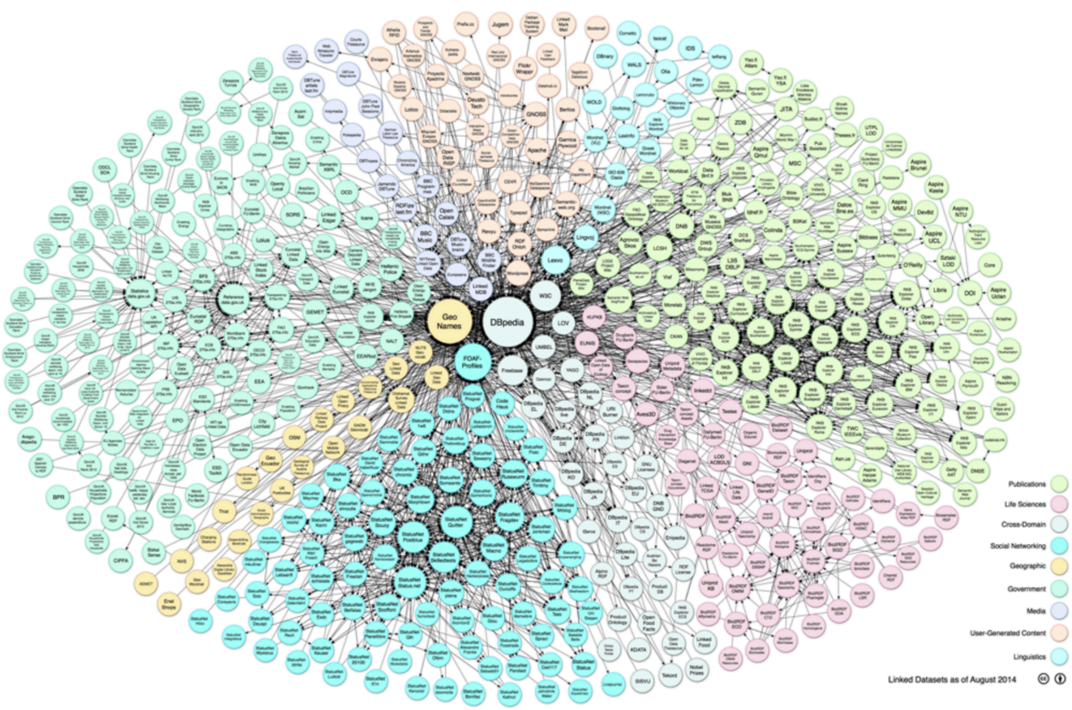
OUTLINE

- ❑ Introduction
- ❑ **Part 1: Data linking**
- ❑ **Part 2: Key discovery**
  - ❑ SAKey: almost key discovery
  - ❑ VICKEY: conditional key discovery
- ❑ **Part 3: SameAs link invalidation**
- ❑ **Part 4: Data fusion**
- ❑ **Conclusion and some future chanllenges**

PART 1:
DATA LINKING

LOD CLOUD IN 2016

- Linked Open Data cloud (LOD)
  - 130+ billion triples and ≈ 0.5 billion links (mostly owl:sameAs)



SAMEAS LINK DISCOVERY PROBLEM

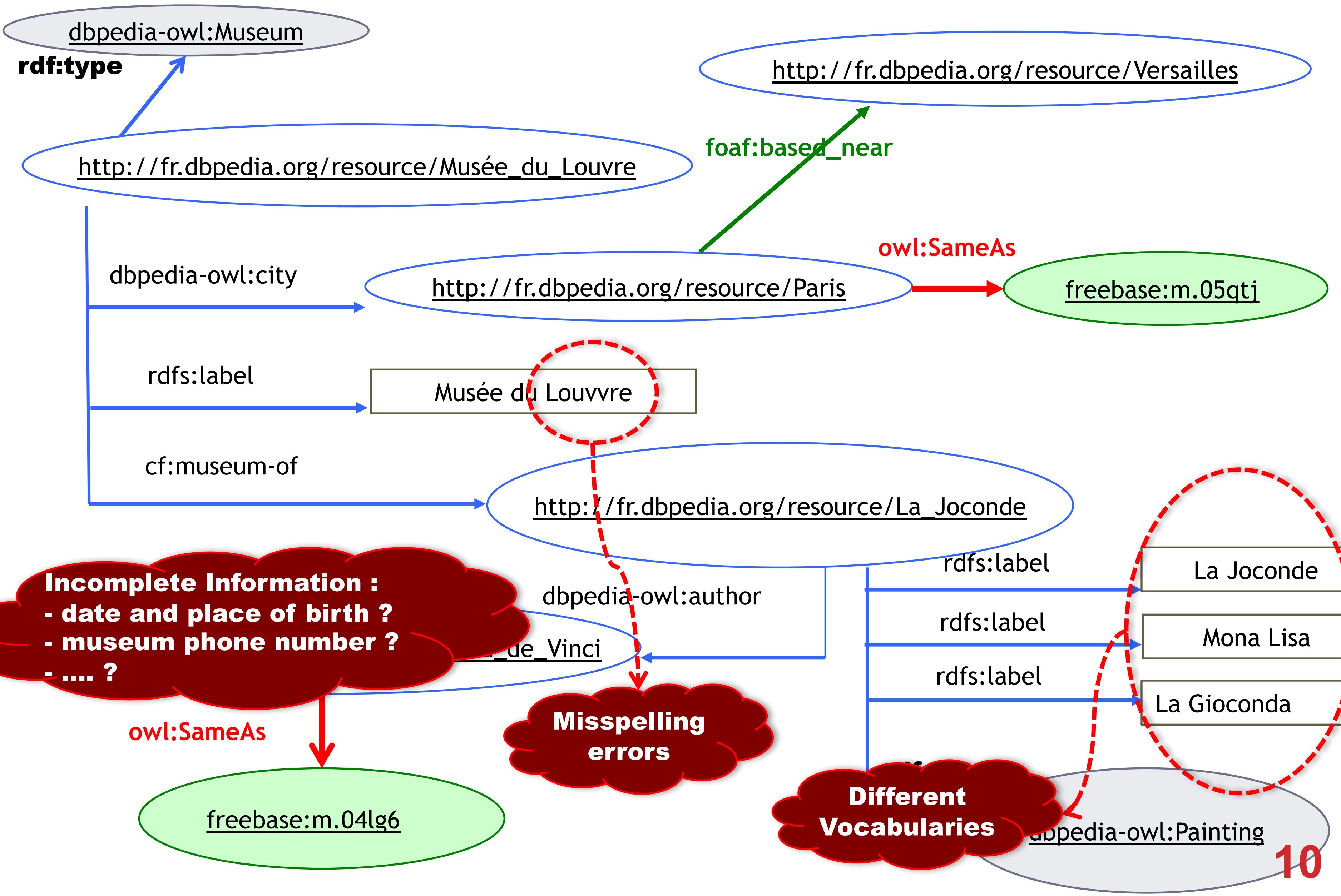
- **SameAs Link discovery** consists in detecting whether two descriptions of resources refer to the same real world entity (e.g. same person, same article, same gene).

- **Definition (Link Discovery)**
  - Given two sets U_1 and U_2 of resources
  - Find a partition of U_1 x U_2 such that :
    - S = {(u1,u2) ∈ u1 × u2: owl:sameAs(s,t)} and
    - D = {(u1,u2) ∈ u1 × u2: owl:differentFrom(s,t)}

- **Naïve complexity** $\in O(U_1 \times U_2)$, i.e. $O(n^2)$

  Example: ≈ 70 days for linking cities in DBpedia and LinkedGeoData

Data linking: difficulties



DATA LINKING: STATE OF THE ART

SOME OF HISTORY …

Problem which exists since the data exists … and under different terminologies: *record linkage, entity resolution, data cleaning, object coreference, duplicate detection, ….*

Automatic Linkage of Vital Records*

[NKAJ, Science 1959]

Computers can be used to extract "follow-up" statistics of families from files of routine records.

H. B. Newcombe, J. M. Kennedy, S. J. Axford, A. P. James

The term *record linkage* has been used to indicate the bringing together of two or more separately recorded pieces of information concerning a particular individual or family (*1*). Defined in this broad manner, it includes almost any use of a file of records to determine what has subsequently happened to people about whom one has some prior information.

***Record linkage**: used to indicate the bringing together of two or more separately recorded pieces of information concerning a particular individual or family.*

portance of repeated natural mutations on the one hand, and of fertility dif-

occurred with frequencies of about 10 percent of all record linkages involving live births and 25 percent of all link-

DATA LINKING IN RELATIONAL DATABASES VS SEMANTIC WEB

	Databases	Semantic Web
Multivaluation	**NO**	**YES** P1 hasAuthor "Michel Chein" P1 hasAuthor "Marie-Christine Rousset"
Open World Assumption	**NO**	**YES**
Ontologies	**NO**	**YES** Use of class hierarchy and ontology axioms

DATA LINKING APPROACHES

- **Instance-based approaches**: consider only data type properties (attributes)

- **Graph-based approaches**: consider data type properties (attributes)as well as object properties (relations) to propagate similarity scores/linking decisions (collective data linking)

- **Supervised approaches**: need an expert to build samples of linked data to train models (manual and interactive approaches)

- **Informed approaches**: need knowledge to be declared in the ontology or in other format given by an expert

DATA LINKING APPROACHES: DIFFERENT CONTEXTS

- Datasets conforming to the same ontology
- Datasets conforming to different ontologies
- Datasets without ontologies

DATA LINKING: OPEN CHALLENGES

- ….

OUTLINE

- Introduction
- Part 1: Data linking
- Part 2: Key discovery
- Part 3: SameAs link invalidation
- Part 4: Data fusion
- Conclusion and some future challenges

PART 2:
KEY DISCOVERY

RULE-BASED DATA LINKING

Some data linking approaches use rules to link data

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)

Rules use discriminative properties => keys

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

Goal: Discover keys automatically

OWL2 KEY

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:

$Book(x_1) \wedge Book(x_2) \wedge Author(x_1, y) \wedge Author(x_2, y) \wedge Title(x_1, w) \wedge Title(x_2, w) \rightarrow sameAs(x_1, x_2)$

KEY DISCOVERY - RELATED WORK

Semantic Web					
Approach	**Composite keys**	**Complete set of keys**	**OWL2 keys**	**Approximate keys**	**Incomplete data heuristics**
[SAS11]			✓	✓	
[SH11]	✓		✓	✓	
[ADS12]	✓	✓		✓	
[KD2R13]	✓	✓	✓		✓

KEY DISCOVERY - RELATED WORK

Semantic Web					
Approach	Composite keys	Complete set of keys	OWL2 keys	Approximate keys	Incomplete data heuristics
[SAS11]			✓	✓	
[SH11]	✓		✓	✓	
[ADS12]	✓	✓		✓	
[KD2R13]	✓	✓	✓		✓

+

Scalability

SAKey

PROBLEM STATEMENT

RDF data might contain errors and/or duplicates

	Name	Actor	Director	ReleaseDate
Film1	"Intouchables"	"F.Cluzet" "O.Sy"	"O.Nakache" "E.Toledano"	"2/11/11"
Film2	"Intouchables"	"F.Cluzet" "O.Sy"	"O.Nakache" "E.Toledano"	"2/11/11"
Film3	"Her"	"J.Phoenix" "S.Johansson"	"S.Jonze"	"10/1/14"
…				

PROBLEM STATEMENT

RDF data might contain errors and/or duplicates

	Name	Actor	Director	ReleaseDate
Film1	“Intouchables”	“F.Cluzet” “O.Sy”	“O.Nakache” “E.Toledano”	“2/11/11”
Film2	“Intouchables”	“F.Cluzet” “O.Sy”	“O.Nakache” “E.Toledano”	“2/11/11”
Film3	“Her”	“J.Phoenix” “S.Johansson”	“S.Jonze”	“10/1/14”
...				

Goal: Discover keys even under the presence of errors and/or duplicates

SAKEY: SCALABLE ALMOST KEY DISCOVERY

Incomplete data

Errors

Duplicates

Large datasets

Discovers almost keys

- Sets of properties that are not keys due to few exceptions

N-ALMOST KEYS

Exception of a key: an instance that shares values with another instance for a given set of properties *P*

	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

Exception of a key: an instance that shares values with another instance for a given set of properties *P*

- f2 is an exception for {Name}

	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

Exception of a key: an instance that shares values with another instance for a given set of properties *P*

- f2 is an exception for {Name}

Exception Set E_P: set of exceptions for *P*

- E_P = {f2, f6} for {Name}

	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 $|E_P| \leq n$**

	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 $|E_P| \leq n$**

- {Name} is a 2-almost key

	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"	---	---

ALMOST KEY DISCOVERY STRATEGY

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

ALMOST KEY DISCOVERY STRATEGY

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

ALMOST KEY DISCOVERY STRATEGY

Non key discovery first

- Partially scan the data

	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”

ALMOST KEY DISCOVERY STRATEGY

Non key discovery first

- Partially scan the data

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”

ALMOST KEY DISCOVERY STRATEGY

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"

ALMOST KEY DISCOVERY STRATEGY

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"

Interested only in maximal non keys

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

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

ALMOST KEY DISCOVERY STRATEGY

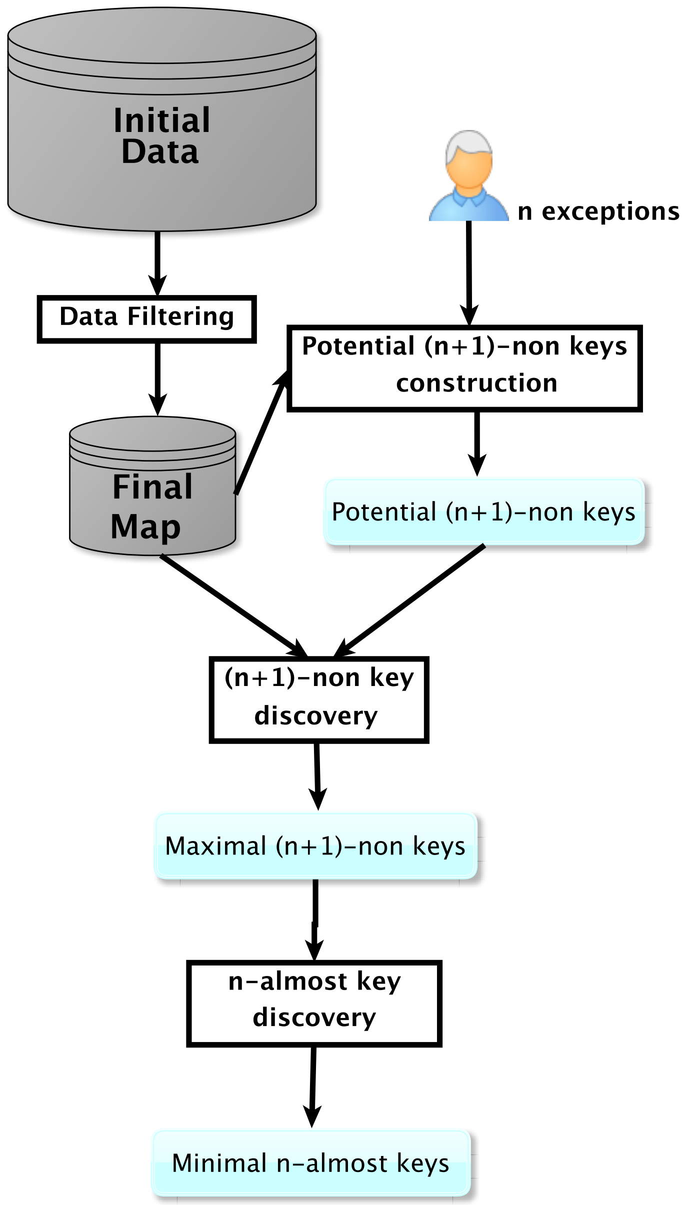
Discover sets of properties that are not *n*-almost keys first

- ***n*-non key:** a set of properties where $|E_P| \geq n$

Derive *n*-almost keys using (n+1)-non keys

Example: **All the sets of properties that are not maximal 3-non keys are 2-almost keys**

SAKEY - GENERAL ARCHITECTURE



N-NON KEY DISCOVERY: PRUNING STRATEGIES

Inclusion pruning

- Discovery of dependencies between data

Seen intersection pruning

- Avoiding already explored sets of instances

Antimonotonic pruning

- All the subsets of a *n*-non key are at least *n*-non keys

EXPERIMENTS

Evaluation of SAKey

- Data Linking using almost keys
- KD2R vs. SAKey
- Scalability of SAKey

Selected datasets

- DBpedia (top classes)
- YAGO
- OAEI 2010, OAEI 2013

DATA LINKING USING ALMOST KEYS

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

DATA LINKING USING ALMOST KEYS

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

Evaluation of linking using

- Recall
- Precision
- F-Measure

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

- Linking results using *n*-almost keys are the better than using keys

EXAMPLE: DATA LINKING USING ALMOST KEYS

OAEI 2013 - Person

- BirthName, BirthDate, award, comment, label, BirthPlace, almaMater, doctoralAdvisor

	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%

KD2R VS. SAKEY

Goal: Compare the runtime of the two approaches

- Non key discovery (SAKey n=0)
- Key derivation

Datasets

- DBpedia (5 classes)
- YAGO (2 classes)

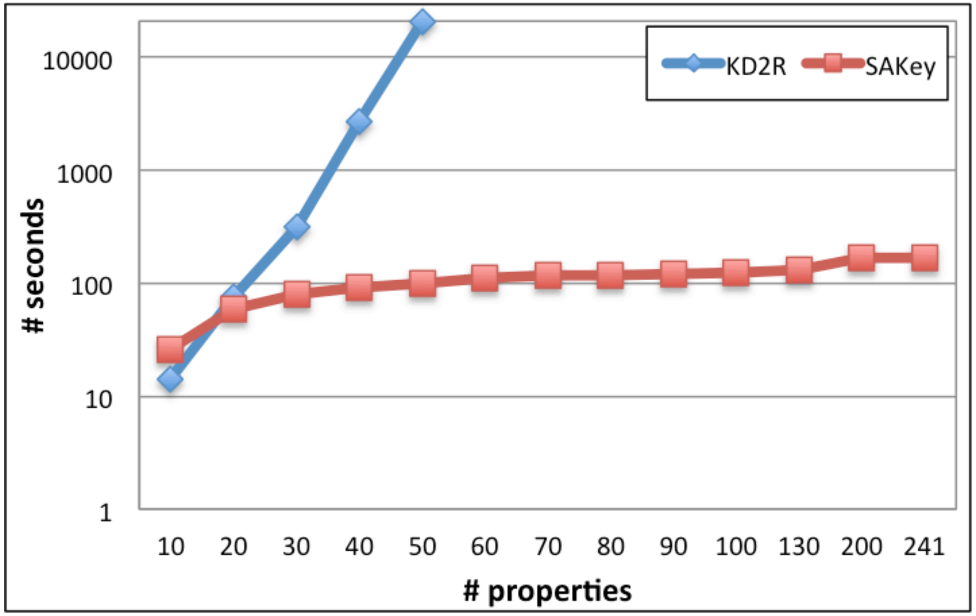
Conclusion

- SAKey non key discovery is orders of magnitude faster than KD2R
- SAKey key derivation is orders of magnitude faster than KD2R

KD2R VS. SAKEY - NON KEY DISCOVERY

Class	# triples	# Instances	#Properties	KD2R Runtime	SAKey Runtime (*n*=0)
DB:Website	8506	2870	66	13min	1s
YA:Building	114783	54384	17	26s	9s
DB:BodyOfWater	1068428	34000	200	outOfMem.	37s
DB:NaturalPlace	1604348	49913	243	outOfMem.	1min10s

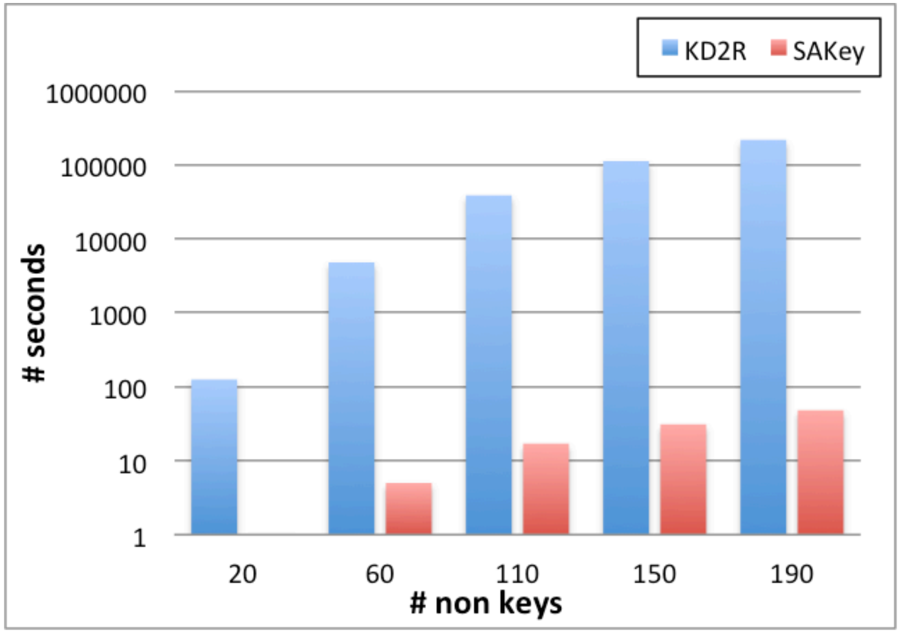
Dbpedia class= DB:NaturalPlace



KD2R VS. SAKEY - KEY DERIVATION

Class	# non keys	# keys	KD2R	SAKey (*n*=0)
DB:Lake	50	480	1min10s	1s
DB:Mountain	49	821	8min	1s
DB:BodyOfWater	220	3846	> 1 day	66s
DB:NaturalPlace	302	7011	> 2 days	5min

Dbpedia class= DB:BodyOfWater



VICKEY: CONDITIONAL KEY DISCOVERY

CONDITIONAL KEY DISCOVERY

- A conditional key is a key constraint that is valid in only a part of the data.
- Definition. (**Conditional key**) A conditional key for a dataset *D* is a non-empty set of conditions $\{cd_1, ..., cd_n\}$ and a non-empty set of properties $\{p_1 , ..., p_m \}$ of *D* (disjoint from the properties in the conditions), such that:

$$\forall x, y,\ u_1, \ldots, u_m \bigwedge_{i=1..n} (cd_i(x) \wedge cd_i(y)) \ \wedge$$
$$\bigwedge_{i=1..m} (p_i(x, u_i) \wedge p_i(y, u_i)) \Rightarrow x = y$$

- Example :

$\forall X\ \forall Y\ \forall Z\ (city(X) \wedge city(Y) \wedge cityName(X,Z) \wedge cityName(Y,Z) \wedge$

$inRegion(X, \text{“Hauts de France”}) \wedge inRegion(Y, \text{“Hauts de France”})) \Rightarrow sameAs(X,Y)$

CONDITIONAL KEY DISCOVERY

- Useful when no or only few keys that are valid in the entire knowledge base (KB).
- May be used in all the applications (data linking, KB enrichment and KB fusion) where classic keys are used.
- Carry knowledge in them selves (e.g. …)
- Conditional key discovery is more complex than key discovery
  - Key discovery problem: $2^{|P|}$, with P is the set of properties
  - Conditional key discovery problem: $|V|^{|P|}$, with V is the set of objects in KB.

VICKEY APPROACH

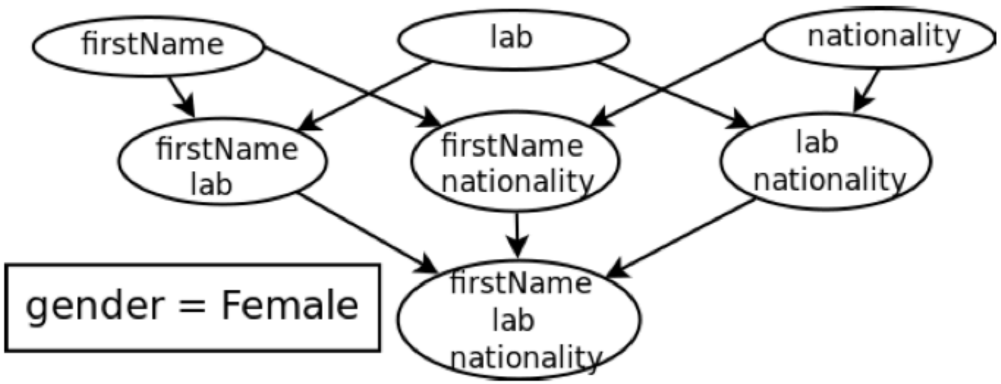
- Discovers minimal conditional keys from a set of maximal non-keys (computed by SAKey).

  - Observation 1: Given a minimal conditional key for a dataset *D* with properties *P* and conditions $\{pc_1 = o_1, ..., pc_n = o_n\}$, the set of properties $P \cup \{pc_1, ..., pc_n\}$ must be a non-key for *D*.

VICKEY APPROACH: ALGORITHM

- **Data structure**: conditional key graph which is a tuple ⟨*Pk, Pc, cond, G*⟩ with the following components:
- P^k and P^c are disjoint sets of properties, called key properties and condition properties, respectively.
- *cond* is a set of conditions on *Pc*.
- G is a directed graph. Each node *v* is associated to a set $v.p \subseteq P^k$ and to a boolean flag *v.explore*, initially set to true. There is a directed edge from *u* to *v* if $u.p \subset v.p$ and $|u.p| = |v.p| - 1$.
- Algorithm:
  - Build all the conditional graphs for which the property condition has a minimum support θ
  - From this, build all conditional key graphs that have a condition set of a given size and respecting θ condition.

VICKEY APPROACH: ALGORITHM



Example of a conditional key graph with P^k = {firstName, lab, nationality}, P^c ={gender}, cond = {gender = Female}.

EXPERIMENTS: SCALABILITY

- Use of nine classes from DBpedia
- Evaluation of VICKEY performance by comparing it with a generic rule mining approach AMIE [Galarraga et al.'13]

Class	Triples	Inst.	#Prop.	#NKs	VICKEY	AMIE	#CKs
Actor	57.2k	5.8k	71	137	4.52m	12.58h	311
Album	786.1k	85.3k	39	68	1.53h	3.90h	304
Book	258.4k	30.0k	51	95	11.84h	> 1d	419
Film	832.1k	82.6k	74	132	1.37h	3.64h	185
Mountain	127.8k	16.4k	58	47	2.86m	23.57m	257
Museum	12.9k	1.9k	65	17	1.46s	6.45s	58
Organisation	1.82M	178.7k	553	3221	26.32h	> 36h	28
Scientist	258.5k	19.7k	73	309	27.67m	> 1d	582
University	85.8k	8.7k	89	140	14.45h	> 1d	941

Table 2: VICKEY vs AMIE on DBpedia classes

EXPERIMENTS: QUALITATIVE EVALUATION

- Use of Dbpedia and YAGO
- There is a gold standard available for the entity links
- Use of simple linking tool with strict string equality

➔ The precision is always over 98%

➔ The use of conditional keys improves significantly the results, e.g., the F1 for film is increased of 47%.

Class		Recall %	Precision %	F1 %	
Actor	Ks	27.43	99.93	**43.05**	
	CKs	57.49	99.63	72.91	× 1.75
	Ks+CKs	60.42	99.81	**75.27**	
Album	Ks	0.01	100.00	**0.03**	
	CKs	15.00	99.39	26.07	× 869
	Ks+CKs	15.01	99.39	**26.08**	
Book	Ks	3.49	100.00	**6.75**	
	CKs	11.31	99.31	20.31	× 3.84
	Ks+CKs	13.33	99.75	**23.51**	
Film	Ks	4.06	99.96	**7.80**	
	CKs	38.17	96.57	54.72	× 7.1
	Ks+CKs	38.62	97.69	**55.35**	
Mountain	Ks	0.22	100.00	**0.44**	
	CKs	28.59	99.39	44.41	× 101
	K+CKs	28.70	99.39	**44.54**	
Museum	Ks	12.05	100.00	**21.51**	
	CKs	24.86	100.00	39.82	× 2.19
	Ks+CKs	30.85	100.00	**47.16**	
Organisation	Ks	1.10	100.00	**2.17**	
	CKs	13.97	98.42	24.46	× 11
	K+CKs	14.24	98.63	**24.88**	
Scientist	Ks	5.78	98.04	**10.92**	
	CKs	16.69	99.93	28.60	× 2.96
	Ks+CKs	19.34	99.41	**32.37**	
University	Ks	8.93	99.87	**16.39**	
	CKs	22.03	99.14	36.04	× 2.44
	Ks+CKs	25.03	99.55	**40.00**	

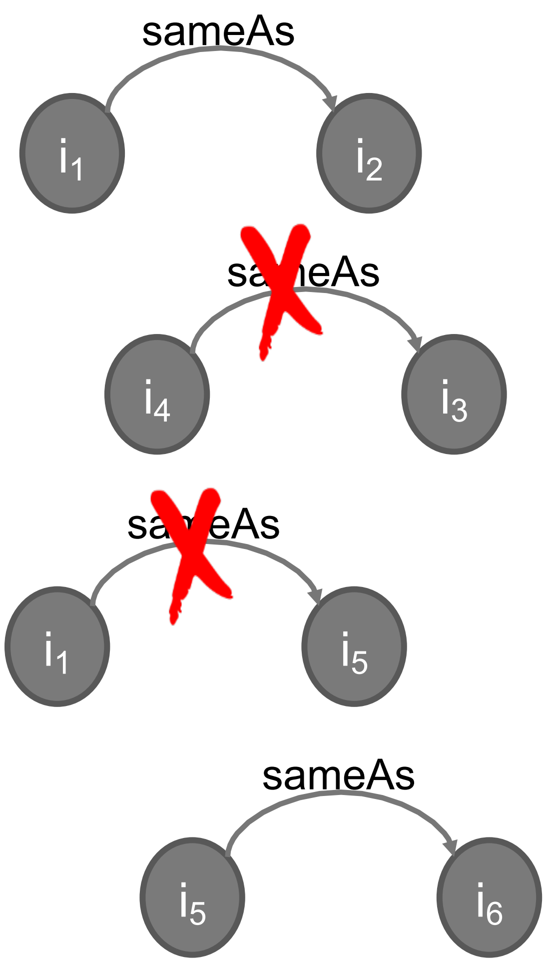
Table 4: Linking results with classical keys (Ks), conditional keys (CKs), and both (Ks+CKs)

OUTLINE

- Introduction
- Part 1: Data linking
- Part 2: Key discovery
  - SAKey: almost key discovery
  - VICKEY: conditional key discovery
- Part 3: SameAs link invalidation
- Part 4: Data fusion
- Conclusion and some future challenges

PART 3:
LINK INVALIDATION

SAMEAS LINK INVALIDATION



- Identity (owl:sameAs) links are often detected automatically.
- Linking tools do not guarantee 100% precision.
- Depending on the data quality and on the linking tool efficiency, some identity links may be incorrect.
- Identity relation is sometimes too strict.

  Example: let b1 and b2, two books.

  SameAs(b1, b2) represents that b1 and b2 are the same book. But why not different editions of the same work?
- Need to develop methods that validate or invalidate identity links.

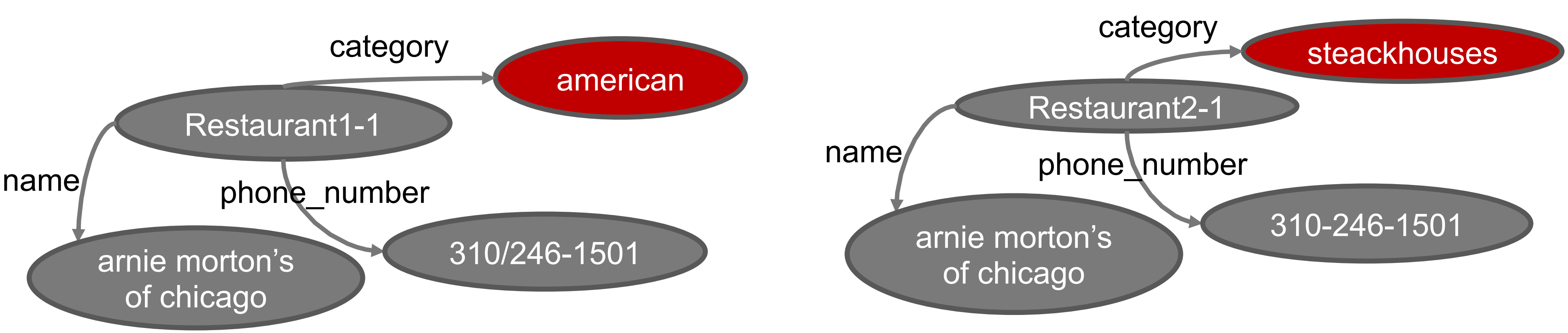
LOGICAL INVALIDATION

- A first logical approach developed in [*Papaleo et al. 2014].*
- *Exploits ontology axioms (functionality of properties and local completeness) to infer invalid links.*
- It suffices to have one property with different values to infer that the sameAs link is invalidated.

- Some results:
  - Exploit data linking results of three tools on OAEI benchmarks.
  - Increase the precision of 4% to 25%.

LOGICAL INVALIDATION

- Limit: if there is an error or literals that are syntactically different but semantically equal, then the tool will provide a false negative.



NUMERICAL INVALIDATION

- A numerical method based on a similarity score computation.
- It exploits ontology axioms (functionality of properties, local completeness) to build a context.

Steps:

- Given a depth m, for each of the two resources, extract a context, i.e., a sub-graph from the RDF description that corresponds to functional and local complete properties.
- Explore the two contexts to compute a similarity score for the sameAs link to be invalidated.
- Given a threshold *T*, infer the invalidation for all the links having a similarity score that is lower than *T*.

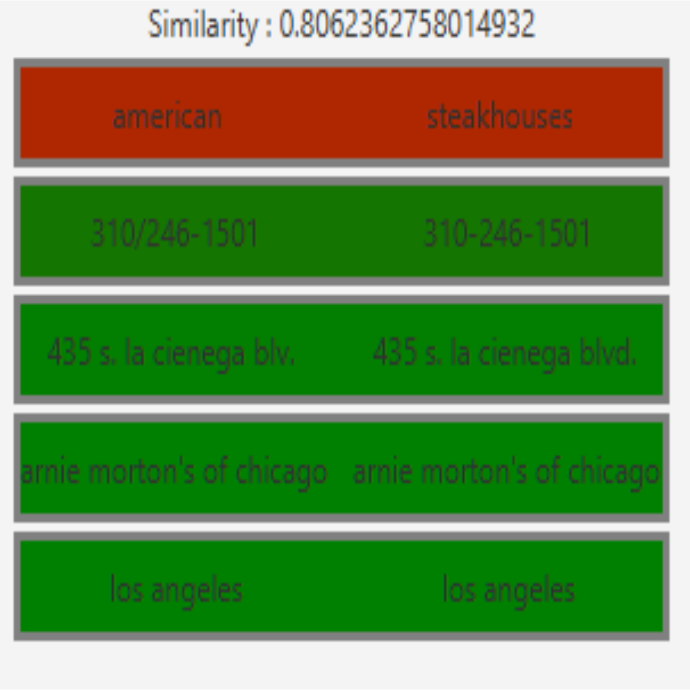
NUMERICAL INVALIDATION

- **Different aggregation functions of the adjacent nodes similarities:**
  - Average
  - Minimum (analogous to the logical method)
  - Weighted average (different weights for the properties).

EXAMPLE OF RESULTS

- It is important to explain why the link is invalid

➔ Show the pairs of literal values and their corresponding similarity scores with respect to the threshold (green or red).



OUTLINE

- Introduction
- Part 1: Data linking
- **Part 2: Key discovery**
  - SAKey: almost key discovery
  - VICKEY: conditional key discovery
- Part 3: SameAs link invalidation
- **Part 4: Data fusion**
- **Conclusion and some future challenges**

PART 4:
DATA FUSION

DATA FUSION

“fusing multiple records representing the same real world object into a single, consistent, and clean representation”

[Bleiholder & Naumann, 2008]

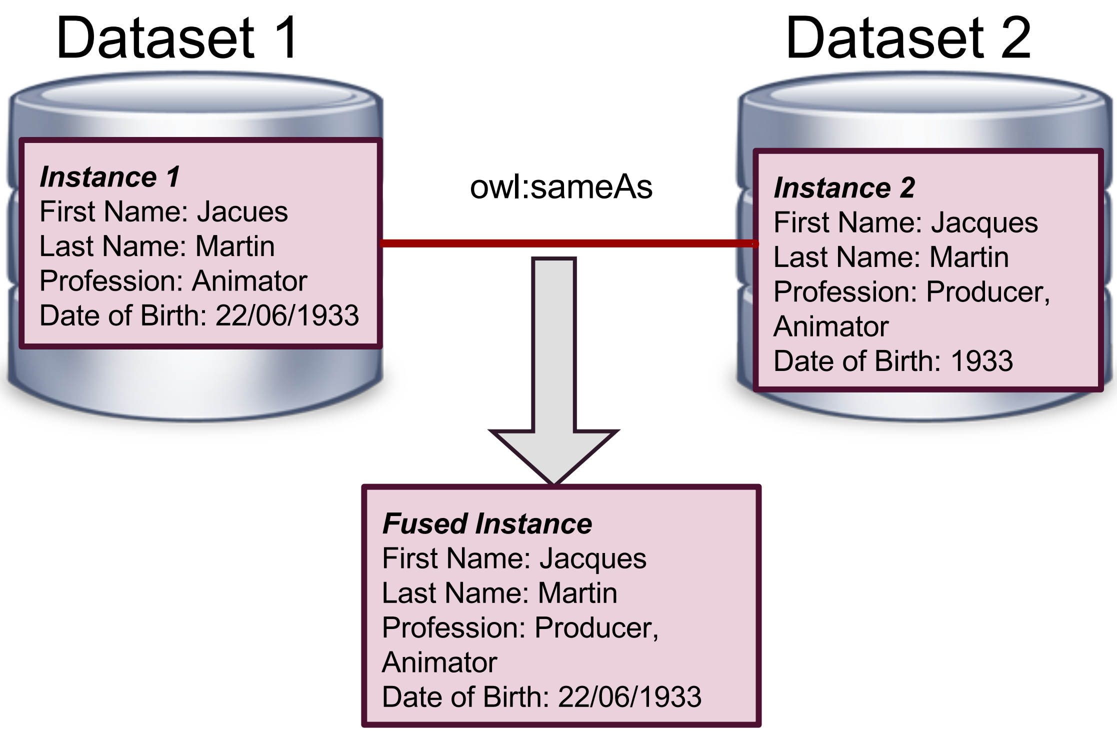
DATA FUSION

- Merge information from objects marked with *sameAs*
- Obtain a single homogenized object

Why fusion?

- Avoid redundancy
- Group together best quality information
- Ensure knowledge consistency

DATA FUSION


Dataset 1
Dataset 2
Instance 1
First Name: Jacues
Last Name: Martin
Profession: Animator
Date of Birth: 22/06/1933
owl:sameAs
Instance 2
First Name: Jacques
Last Name: Martin
Profession: Producer, Animator
Date of Birth: 1933
Fused Instance
First Name: Jacques
Last Name: Martin
Profession: Producer, Animator
Date of Birth: 22/06/1933

DATA FUSION

Challenge: Properties with conflicting values!

- <Great Britain>, <UK>
- <Prime Minister>, <Politician>
- <Louvre>, <Lovre>

→ Which one to choose?

DATA FUSION: CONFLICT RESOLUTION STRATEGIES
[P.N. MENDES ET AL'12, BLEIHOLDER & NAUMANN, 2008]

Independent from data quality

- Keep the most frequent value
- Average, max, min, concatenation, intervals

Data quality-driven

- Keep the value with the best confidence degree (or / threshold)
- Be confident with a data source
- Apply a vote weighted by data source reliability degree

METHODOLOGY: STEP 1

Categorize values

- Allows to apply specified controls and measures

1933 → *numeric*

Prime Minister, Politician → *hierarchical*

"Jacques" → *symbolic*

METHODOLOGY: STEP 2

Detect *implausible* values

Example 1: Misspell

- <hasName>Louvre</hasName>
- <hasName>Lovre</hasName>

→ "Lovre" is implausible: very low frequency in the data sources

Example 2: Expert Rules violation

- <hasAge>25</hasAge>
- <hasAge>-35</hasAge>

→ "-35" is implausible: only accept positive values for age

METHODOLOGY: STEP 3

- **Calculate quality score**

For plausible values, use criteria:

  - Frequency
  - Homogeneity
  - Source freshness
  - Source reliability

→Quality score: (weighted) average

METHODOLOGY: STEP 4

Discover relations

For plausible values, find if they are related to other values:

- More Precise: Paris, Ile-de-France
- Synonym: Great Britain, UK
- Incompatible: birth date < death date

→Relations can affect the quality score

METHODOLOGY: STEP 5

- **Values selection**

→ Sort values by quality score

→ Mono-valued: select best value

→ Multi-valued: all plausible values

KEEP TRACK OF FUSION DECISIONS

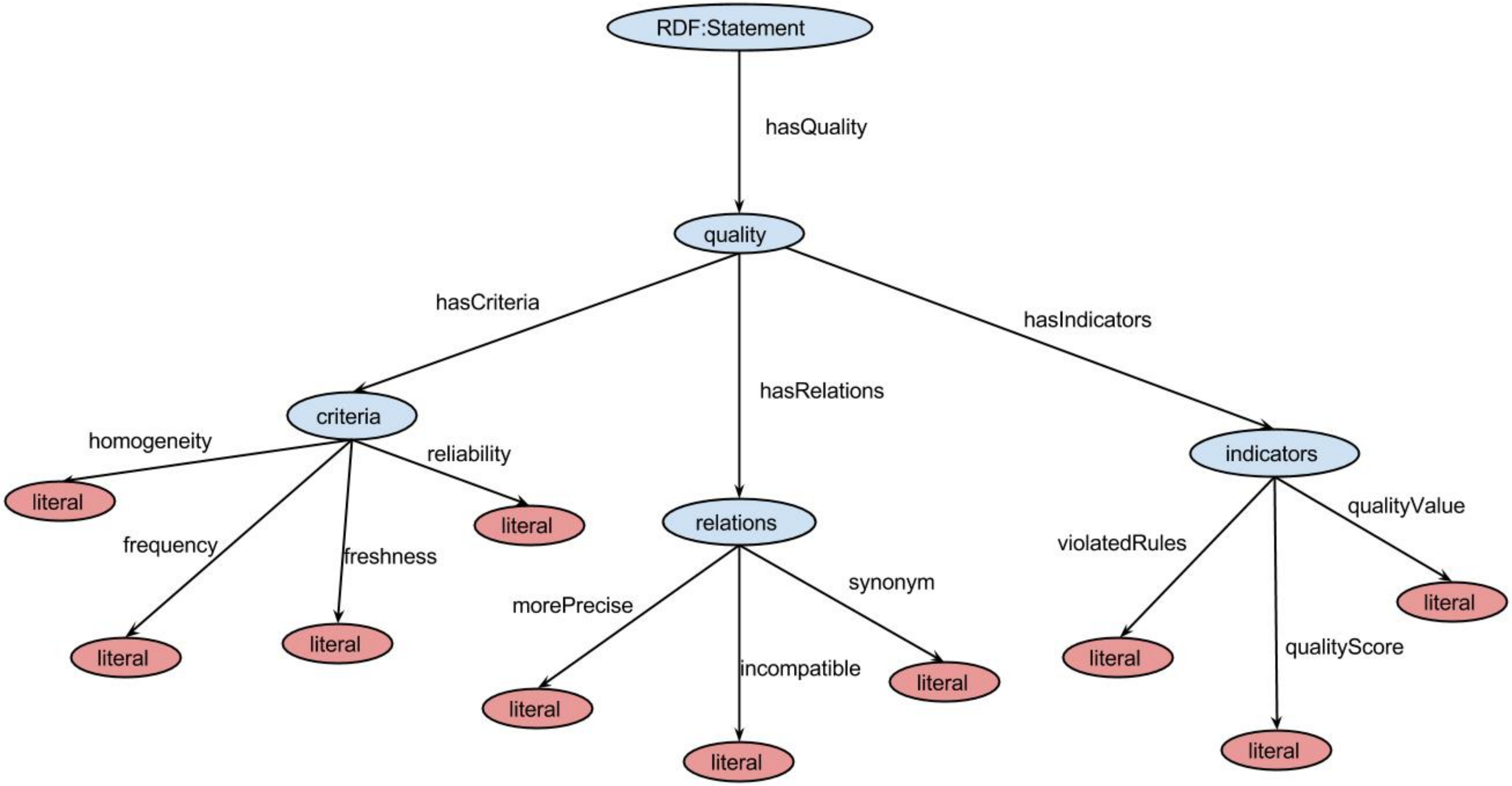
Why is a value selected?

How was the fusion decision taken?

The system stores all the quality aspects

Annotate values with quality information

THE ANNOTATION ONTOLOGY



THE ANNOTATION ONTOLOGY

PERSON_1723 rdf:type ina:PhysicalPerson
v1 rdf:type dfa:Value
q1 rdf:type dfa:Quality
c1 rdf:type dfa:Criteria

PERSON_1723 ina:first_name v1
v1 dfa:hasValue "Jacques"
v1 dfa:isImplausible false
v1 dfa:hasQuality q1
q1 dfa:hasCriteria c1
c1 dfa:hasHomogeneity 0.98
c1 dfa:hasOccurenceFrequency 0.02
c1 dfa:hasReliability 0.8
c1 dfa:hasFreshness 0.7
q1 dfa:hasQualityScore 0.67
q1 dfa:hasQualityValue "excellent"

v2 rdf:type dfa:Value
q2 rdf:type dfa:Quality
c2 rdf:type dfa:Criteria

PERSON_1723 ina:first_name v2
v2 dfa:hasValue "Jacues"
v2 dfa:isImplausible true
v2 dfa:hasQuality q2
q2 dfa:hasCriteria c2
c2 dfa:hasHomogeneity 0.015

FUSION RESULTS ON INA DATA

Results on a corpus of 10819 French celebrities

Table (a)

Person	# instances
Jacques Martin	10288
Philippe Bouvard	264
Daniel Prevost	214
Frederic Martin	26
Emmanuel Petit	12
Luis Fernandez	7
Michel Leclerc	6
Virginie Lemoine	2

Table (b)

	#values	%
#distinct values	14588	–
#isImplausible = “true”	9370	64.23 %
#isImplausible = “false”	5218	34.76 %
#qualityValue = “excellent”	2	0.04 %
#qualityValue = “medium”	3233	61.95 %
#qualityValue = “poor”	1983	38 %

Thresholds for the choice of quality values

- *if qualityScore* $\geq$ *0.67 then qualityValue = “excellent”.*
- *if 0.33 <qualityScore > 0.67 then qualityValue = “medium”.*
- *if qualityScore* $\leq$ *0.33 then qualityValue = “poor”.*

DATA FUSION: EVALUATION

Evaluation of data integration techniques:

- Completeness (recall)
- Conciseness (precision)
- Consistency (conformity to constraints)

OUTLINE

- Introduction
- Part 1: Data linking
- Part 2: Key discovery
  - SAKey: almost key discovery
  - VICKEY: conditional key discovery
- Part 3: SameAs link invalidation
- Part 4: Data fusion
- **Conclusion and some future challenges**

FUTURE CHALLENGES

- **Data linking**
  - **sameAs semantics:** reasoning on LOD, e.g. transitivity?
  - **Link validation**: incorrect link detection
  - **Link provenance**: representation, use
  - **Data evolution** → Link evolution
  - **Data privacy**: how link data in such contexts [Vatsalan13]?

FUTURE CHALLENGES

- Key discovery
  - Scalability for the conditional key discovery
  - Key selection problem
  - Irrelevant property filtering
  - Data evolution ➔ incremental approaches

FUTURE CHALLENGES

- Link invalidation
  - Combined approaches of data linking and link invalidation
  - Requalification of links, e.g. sameBook vs sameWork?

- Data fusion
  - Qualitative evaluation, lack of gold standard
  - Data quality evaluation under open world assumption: completeness, correctness and conciseness

REFERENCES (1)

[Wu, Z., Palmer, M.'94] Verb semantics and lexical selection.

[Volz et al'09] Silk – A Link Discovery Framework for the Web of Data.

Julius Volz, Christian Bizer et al.

[Nikolov et al'08] Handling instance coreferencing in the KnoFuss architecture.

Andriy Nikolov, Victoria Uren, Enrico Motta and Anne de Roeck

[Nikolov et al'12] ***Unsupervised Learning of Link Discovery Conguration***

Andriy Nikolov, Mathieu d'Aquin, Enrico Motta

[Saïs et al.07] L2R: a Logical method for Reference Reconciliation.

Fatiha Saïs, Nathalie Pernelle and Marie-Christine Rousset.

[Saïs et al.09] Combining a Logical and a Numerical Method for Data Reconciliation.

Fatiha Saïs., Nathalie Pernelle and Marie-Christine Rousset.

[Bleiholder & Naumann, 2008] Data fusion (ACM Computing Surveys)

Jens Bleiholder , Felix Naumann,

[P.N. Mendes et al'12] Sieve Linked Data Quality Assessment and Fusion

Pablo N. Mendes, Hannes Mühleisen, Christian Bizer

[Saïs et Thomopoulos'08] Reference Fusion and Flexible Querying.

Fatiha Saïs and Rallou Thomopoulos.

REFERENCES

[Shvaiko,Euzenat13] Ontology Matching: State of the Art and Future Challenges,

Pavel Shvaiko, Jérôme Euzenat.

[Suchanek11] PARIS: Probabilistic Alignment of Relations, Instances, and Schema

Fabian Suchanek, Serge Abiteboul, Pierre Senellart

[Ferrara13] Evaluation of instance matching tools: The experience of OAEI.

Alfio Ferrara, Andriy Nikolov, Jan Noessner, François Scharffe.

[RiMOM2013] Results for OAEI 2013

Qian Zheng, Chao Shao, Juanzi Li, Zhichun Wang and Linmei Hu

[Atencia et al.'12] Keys and Pseudo-Keys Detection for Web Datasets Cleansing and Interlinking.

Manuel Atencia, Jérôme David, François Scharffe

[Hu'11] A Self-Training Approach for Resolving Object Coreference on the Semantic Web

Wei Hu, Jianfeng Chen, Yuzhong Qu

[Pernelle et al.'13] An Automatic Key Discovery Approach for Data Linking.

Nathalie Pernelle, Fatiha Saïs. and Danai Symeounidou.


merci
thank you
gracias
danke
obrigado
go raibh maith agat
grazie
bedankt
sukriya
arigatô
ngiyabonga
dakujem
mochchakkeram
teşekkür ederim
dank je
terima kasih
dziękuję
謝謝
спасибо
sagolun
chokrane
tapadh
takk
maururu
hvala
taiku
vinaka
kiitos
ευχαριστώ
Баярлалаа
kia ora
ありがとう
мерси
grazzi
barka
tau
tack
sulpáy
akun
ačiū
bayarlalaa
děkuji
rahmat
tanemirt
mersi
diolch
djiere dieuf
dhanyavad
shukriya
spas
chnorakaloutioun
teşekkü ederim
gràcies
welalin
enkosi
gratias ago
misaotra
matondo
dhanyavadagalu
তোমাকে ধন্যবাদ
mamnun
kop khun
najis tuke
paldies
faafetai lava
kam sah hamnida
nanni
trugarez
köszönöm
rahmet
kop khun krap
mercé
leat
nandri
krap
didi madloba
mahalo
gràçie
mèsi
blagodaram
хвала
asante
xièxie
dankie
sobodi
dankon
manana
tenki
murakoze

DATA LINKING

SIMILARITY MEASURES

SIMILARITY MEASURES

Need of normalization and similarity measures when comparing entities

- Use normalization methods for data property (attribute) values:
  - Stop words elimination (e.g. the, this, and, at, …),
  - Stemming (e.g. fishing → fish, fisher → fish),
  - Enforce common abbreviations (e.g. D&K →Data and Knowledge),
  - Part of ETL tools, commonly using field segmentation and dictionaries.

- Use similarity measures between two values
  - Basic problem: given two property values S and T quantify their 'similarity' in [0..1].
  - Problem challenging for strings

SIMILARITY MEASURES

- **Token based (e.g. Jaccard, TF/IDF cosinus) :**

  The similarty depends on the set of tokens that appear in both S and T.

- **Edit based (e.g. Levenstein, Jaro, Jaro-Winkler) :**

  The similarty depends on the smallest sequence of edit operations which transform S into T.

- **Hybrid (e.g. N-Grams, Jaro-Winkler/TF-IDF, Soundex)**

LN2R: A LOGICAL AND NUMERICAL METHOD FOR REFERENCE RECONCILIATON

LN2R
(GRAPH BASED, UNSUPERVISED AND INFORMED)

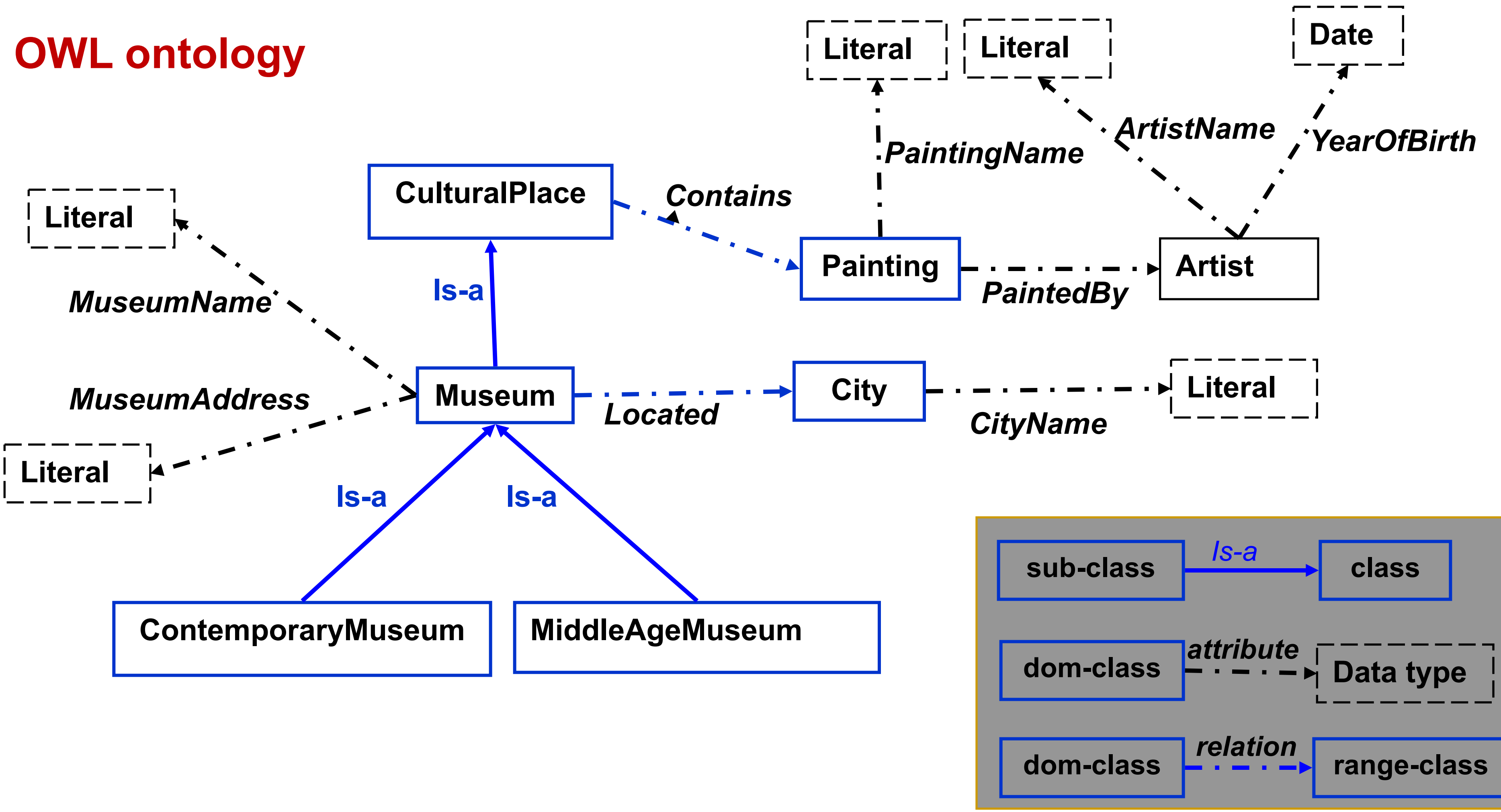
[Saïs et al' 07, Saïs et al'09]

- A combination of two methods:
  - L2R, a Logical method for reference reconciliation: applies logical rules to infer sure owl:sameAs and owl:differentFrom links
  - N2R, a Numerical method for reference reconciliation: computes similarity scores for each pair of references
- **Assumptions**
  - The datasets are conforming to the same ontology
  - The ontology contains axioms

LN2R
(GRAPH BASED, UNSUPERVISED AND INFORMED)

[Saïs et al’ 07, Saïs et al’09]

OWL ontology



LN2R
(GRAPH BASED, UNSUPERVISED AND INFORMED)

[Saïs et al' 07, Saïs et al'09]

Ontology axioms:

- Disjunction axioms between classes, DISJOINT(C, D)
- Functional properties axioms, PF(P)
- Inverse functional properties axioms, PFI(P)
- A set of properties that is functional or inverse functional axioms

Assumptions on the data

- Unique Name Assumption, UNA(src1)
- Local Unique Name Assumption, LUNA(R)

Example:

Authored(p, a1), Authored(p, a2), Authored(p, a3) …., Authored(p, an)
$\Rightarrow$ (a1 $\neq$a2), (a1 $\neq$ a3), (a2 $\neq$ a3) , …

L2R: A LOGICAL METHOD FOR REFERENCE RECONCILIATION

L2R: AUTOMATIC GENERATION OF INFERENCE RULES

Translation of UNA(src1)

R1:src1(X) $\wedge$ src1(Y) $\wedge$ (X $\neq$ Y) $\Rightarrow$ $\neg$Reconcile(X,Y) ; ...

Translation of LUNA(R)

R11(R) : R(Z, X) $\wedge$ R(Z, Y) $\wedge$ (X $\neq$ Y) $\Rightarrow$ $\neg$Reconcile(X,Y) ; ...

Translation of DISJOINT(C, D):

R5(C, D) : C(X) $\wedge$ D(Y) $\Rightarrow$ $\neg$ Reconcile (X, Y)

Translation of PF(R):

R6.1(R): Reconcile(X, Y) $\wedge$ R(X, Z) $\wedge$ R(Y, W) $\Rightarrow$ Reconcile (Z, W)

R6.1(Located): Reconcile(X, Y) $\wedge$ Located (X, Z)$\wedge$Located (Y, W) $\Rightarrow$ Reconcile (Z, W)

Translation of PF(A):

R6.2(A): Reconcile(X, Y) $\wedge$ A(X, Z) $\wedge$ A(Y, W) $\Rightarrow$SynVals(Z, W)

R6.2(MuseumName):Reconcile(X,Y) $\wedge$ MuseumName (X, Z) $\wedge$
MuseumName (Y,W) $\Rightarrow$SynVals(Z, W)

Algorithm: apply until saturation the resolution principle [Robinson'65], by following the unit strategy

L2R: INFERENCE ALGORITHM

- Apply until saturation the resolution principle [Robinson'65], by following the unit strategy

  Resolution rule : $\dfrac{C_1 : (L_1), C_2 : (L_2 \vee C)}{C_{1,2} : (C_\sigma)}$ Avec $L_{1\sigma} = \neg L_{2\sigma}$

- R ∪ F: Horn clauses without functions, where :
  - R: rules in the form of horn clauses
  - F: unit clauses fully instantiated,
    - Reference descriptions: RDF facts (class-facts, relation-facts and attribute-facts).
    - Facts that express the reference origin: src1(i) and src2(j)
    - Facts that express the synonymy and not synonymy between values: SynVals(v1, v2) or ¬ SynVals(v1, v2)
- Computation of the set SatUnit(R ∪ F)

N2R: A NUMERICAL METHOD FOR REFERENCE RECONCILIATION

N2R: A NUMERICAL METHOD FOR REFERENCE RECONCILIATION

- **N2R computes a similarity score for pair of references obtained from their common description.**
  - Uses known similarity measures, e.g. Jaccard, Jaro-Winkler.
  - Exploits ontology knowledge in a way to be coherent with L2R.
  - May consider the results of L2R: *Reconcile(i, i')*, ¬*Reconcile(i, i')* , *SynVals(v, v')* and ¬*SynVals(v, v')*.

SIMILARITY DEPENDENCY MODELLING

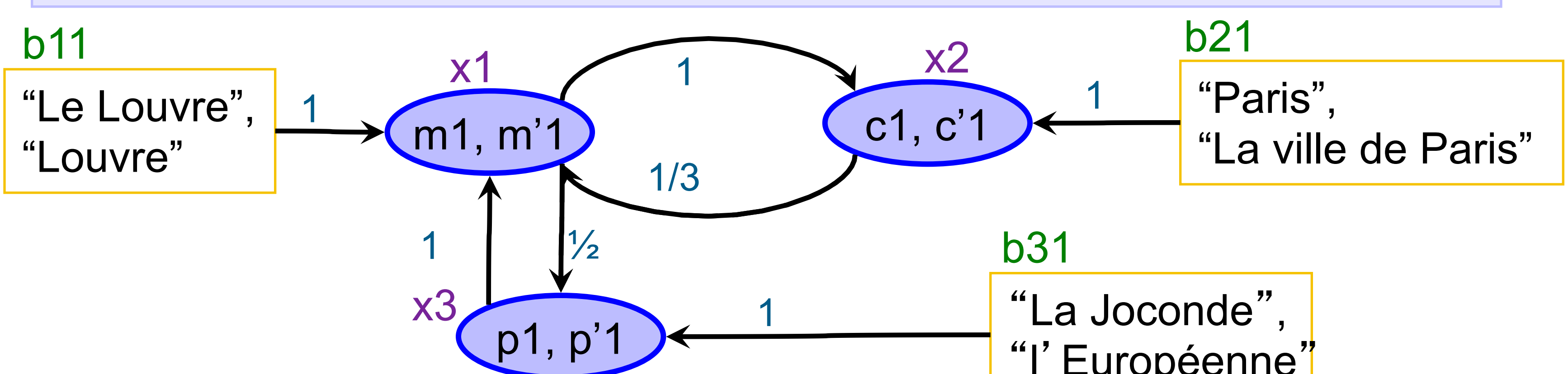
RDF facts in source S1:
Located(m1, c1), MuseumName(m1, “le Louvre”)
Contains(m1, p1), CityName(c1, “Paris”)
PaintingName(p1, “la Joconde”)

RDF facts in source S2 :
Located(m’1, c’1), MuseumName(m’1, “Louvre”)
Contains(m’1, p’1), CityName(c’1, “la Ville de Paris”)
PaintingName(p’1, “l’Europèenne”)

CAttr(m1, m’1) = {MuseumName} ,
CAttr(c1, c’1)= {CityName},CAttr(p1,p’1)={PaintingName}
CRel(m1, m’1)= {Located, Contains}
CRel(c1, c’1) = {Located }, CRel(p1,p’1) = {Contains}

MuseumName+(m1) = {”Le Louvre”},
MuseumName+(m’1) = {”Louvre”},
Located+(m1) = {c1}, Located+(m’1) = {c’1},
Located-(c1) = {m1} , Located-(c’1) = {m’1}, ….

(c1, c’1) is functionally dependent on (m1, m’1)



➔ Equation system

AN EQUATION SYSTEM FOR SIMILARITY COMPUTATION

- Variables: reference pairs similarity
- A variable x_i is assigned to each $Sim_r(ref, ref')$

- Equations: express the similarity computation for each $Sim_r(ref, ref')$:
  - b_i is the similarity score of the attribute values
  - λ_j is the weight associated to the common attributes and common relations x_i.

N2R: THE NON LINEAR EQUATION SYSTEM

$$x_i = \max\Bigg(\max\Big(\bigcup_{j=0}^{j=|DF_A(<ref,ref'>)|} (b_{ij-df}), \bigcup_{j=0}^{j=|DF_R(<ref,ref'>)|} (x_{ij-df}), \Big),$$

$$\Big(\sum_{j=0}^{j=|NDF_A(<ref,ref'>)|} (\lambda_{ij} * \ b_{ij-ndf}) + \sum_{j=0}^{j=|NDF_A^*(<ref,ref'>)|} (\lambda_{ij} * \ BS_{ij-ndf}) +$$

$$\sum_{j=0}^{j=|NDF_R(<ref,ref'>)|} (\lambda_{ij} * \ x_{ij-ndf}) + \sum_{j=0}^{j=|NDF_R^*(<ref,ref'>)|} (\lambda_{ij} * XS_{ij-ndf})) \Bigg)$$

DF(x_i), considered in the maximum

NDF(x_i), considered in the average

➔ A non linear system

NON LINEAR EQUATION SYSTEM RESOLUTION

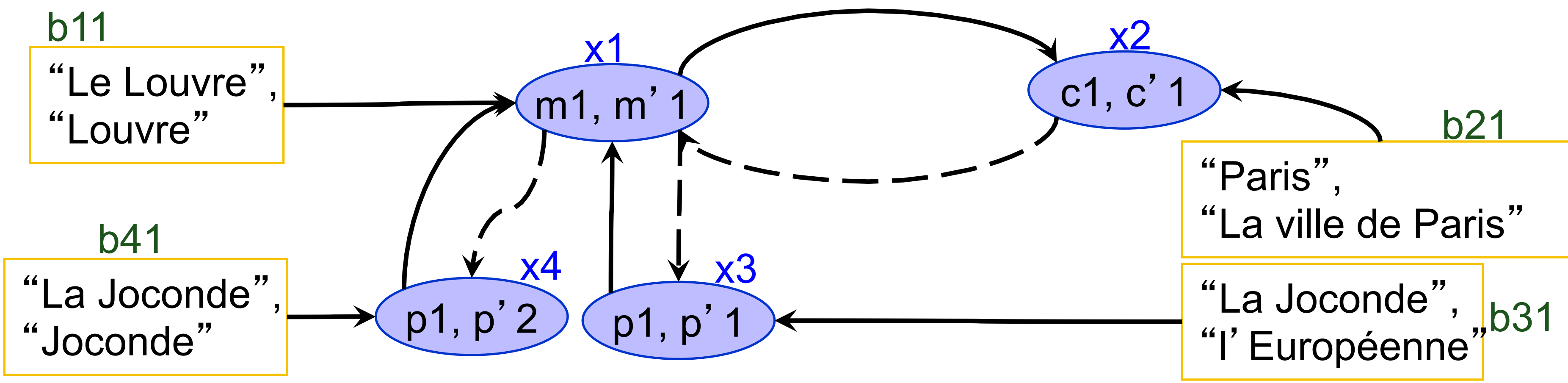
- **An iterative method inspired form *Jacobi*.**
  - Initialize the variable s x_i at 0.
  - Refine iteratively the value of each x_i by using the values x_i computed at a precedent iteration.

  - Termination: a fix-point with a precision ε

$$\forall\ x_i \quad |x_i^k - x_i^{k-1}| < \varepsilon$$

➔**Convergence proof.**

N2R: ILLUSTRATION



x1 = max(max(b11, x3), x4), λ * x2)

x2 = max(b21, x1)

x3 = max(b31, λ* x1)

x4 = max(b41 , λ * x1)

	x1	x2	x3	x4
Initialization	0.0	0.0	0.0	0.0
Iteration 1	0.8	0.3	0.1	0.7
Iteration 2	0.8	0.8	0.4	0.7
Iteration 3	0.8	0.8	0.4	0.7

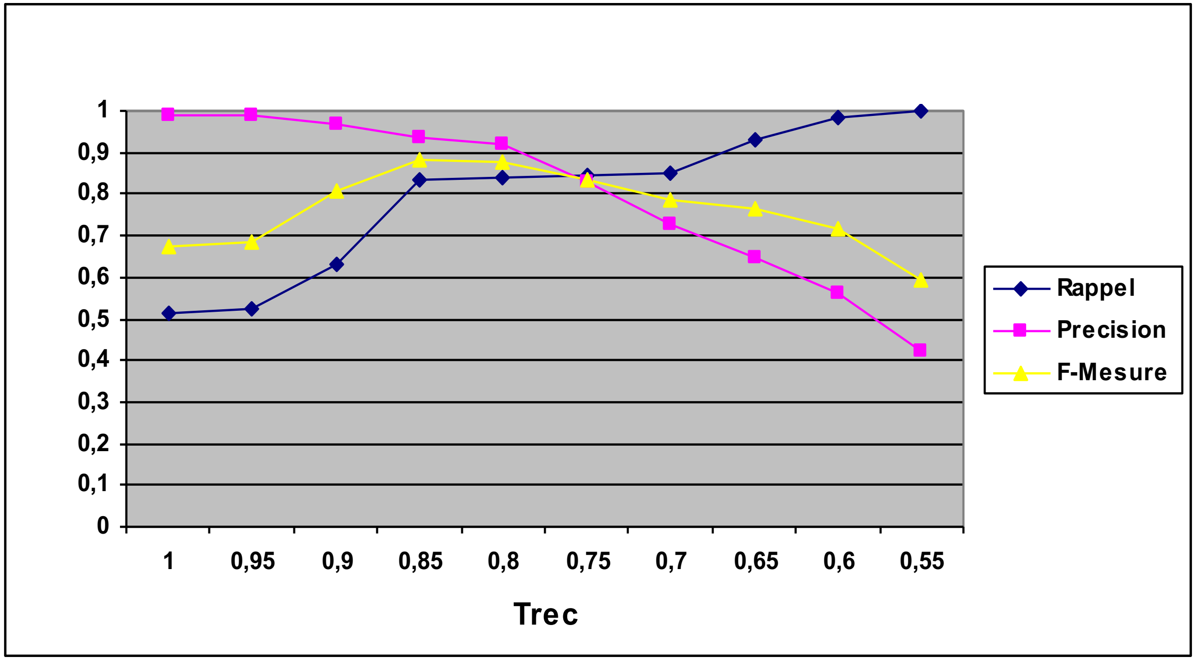
λ= 1/(| CAttr | + | CRel |) ε = 0.02

b11 = 0.8, b21 = 0.3, b31 = 0.1, b41 = 0.7

Solution: x1 = 0.8
x2 = 0.8
x3 = 0.4
x4 = 0.7

N2R EXPERIMENTS

N2R: RESULTS ON CORA



Trec=1, all the reconciliations obtained by L2R are also obtained by N2R.

Trec=1 to Trec=0.85, the recall increases of 33 % while the precision decreases only of 6 %.

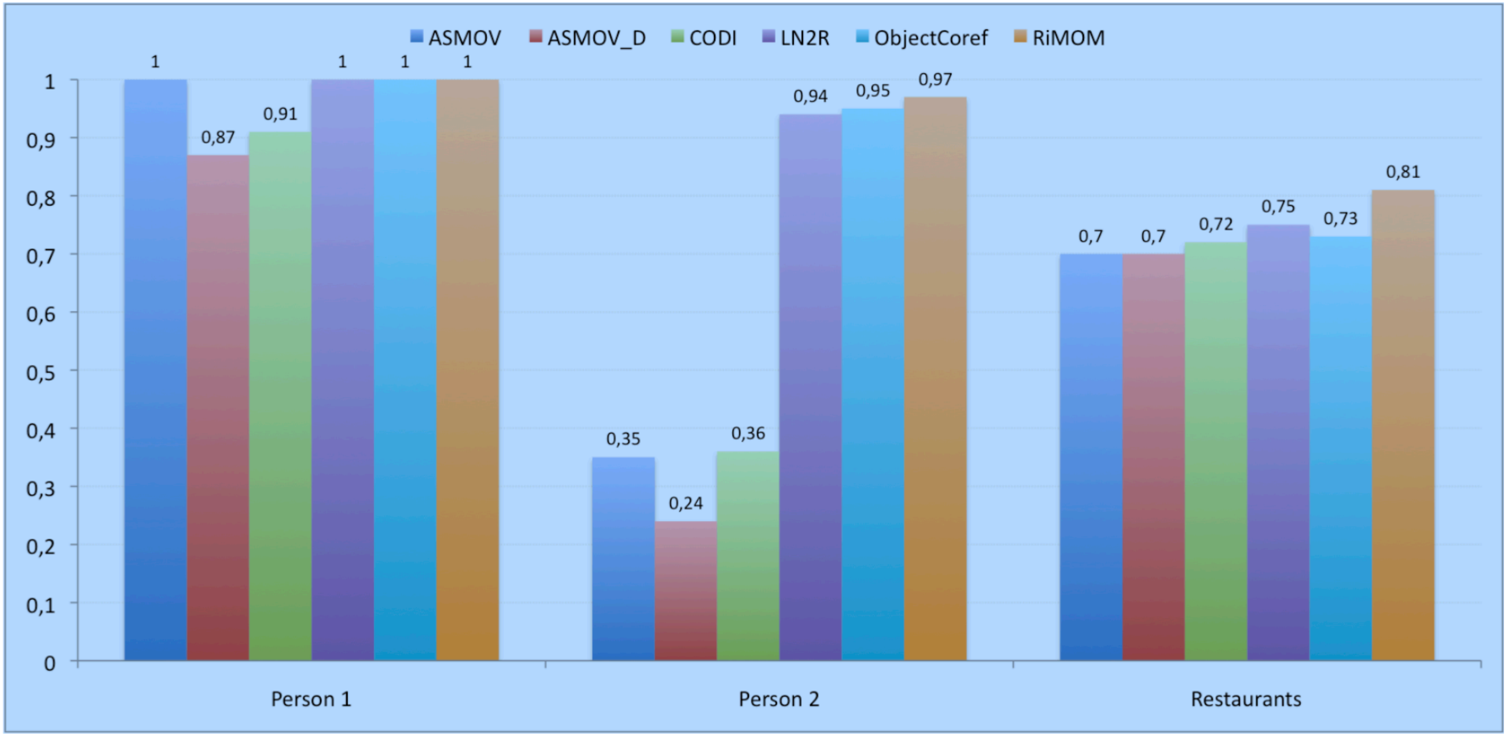
Trec = 0.85, the F-measure is of 88 %:

- Better than the results obtained by the supervised method of [Singla and Domingos'05]
- Worst than those (97 %) obtained by the supervised method of [Dong et al.'05]

II. LIAGE DE DONNÉES

1. LN2R - N2R : RÉSULTATS

OAEI 2010 – Instance matching track (PR), 2ème



CONCLUSION

Data linking: numerous and different approaches …

- **Informed approaches**: need knowledge to be declared in the ontology (generality) and/or ad-hoc knowledge given by an expert (a selection of properties, similarity functions)

→This kind of knowledge is not always available but can be learnt/discovered from the data (e.g., key/rule discovery approaches [Symeonidou et al. 14, Galarraga et al. 13]

- **Supervised approaches**: needs samples of linked data

  → It can be avoided by using assumptions like (UNA)

- **Graph-based approaches**: decision propagation (good recall but highly time consuming)

- **Logical approaches**: good precision but partial

➔ Few approaches generate differentFrom(i1,i2) or use dissimilarity evidence

SOME CHALLENGES

- **sameAs semantics:** reasoning on LOD ?
- **Link validation**: incorrect link detection
- **Link provenance**: representation, use
- **Data evolution** → Link evolution
- **Data privacy**: how link data in such contexts [Vatsalan13]?