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

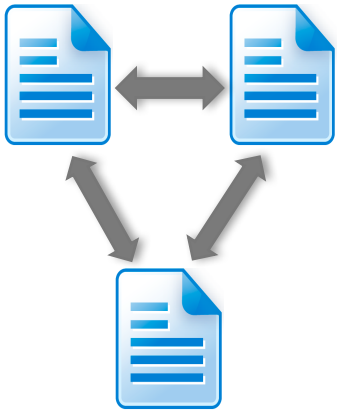
FATIHA SAÏS

LRI, CNRS & PARIS SUD UNIVERSITY

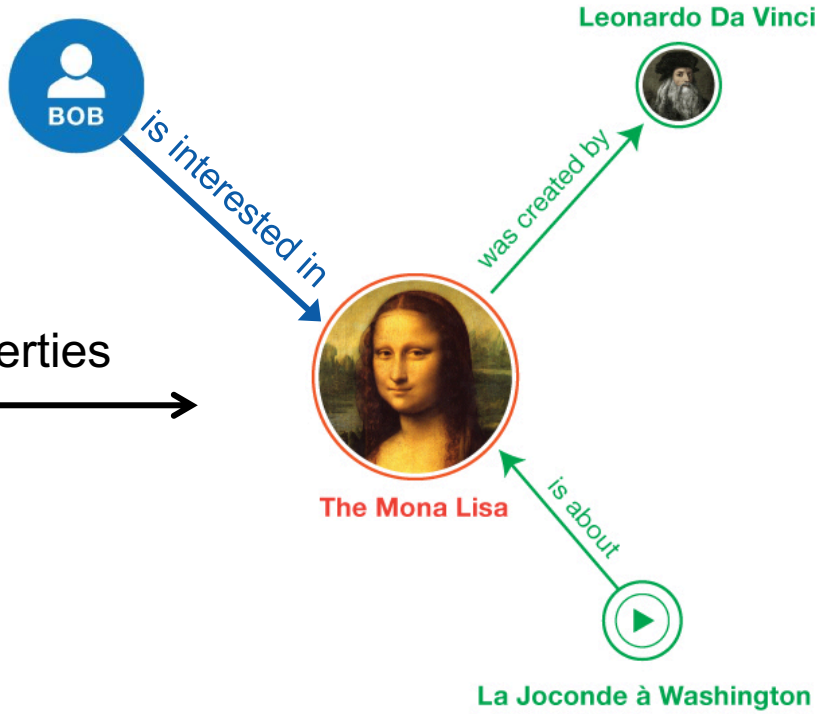
LIST SEMINAR- MARCH 23th 2017, LUXEMBOURG

FROM THE WWW TO THE WEB OF DATA

- applying the principles of the WWW to data



data is links,
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**

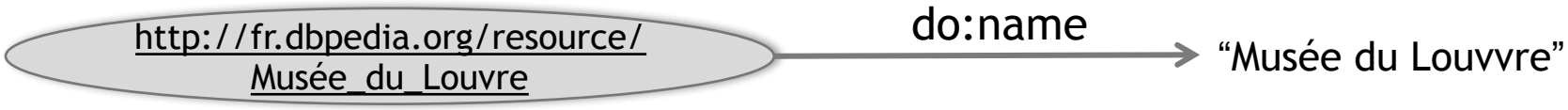
→so people can discover more things

→bridging disciplines and domains

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

RDF – RESOURCE DESCRIPTION FRAMEWORK

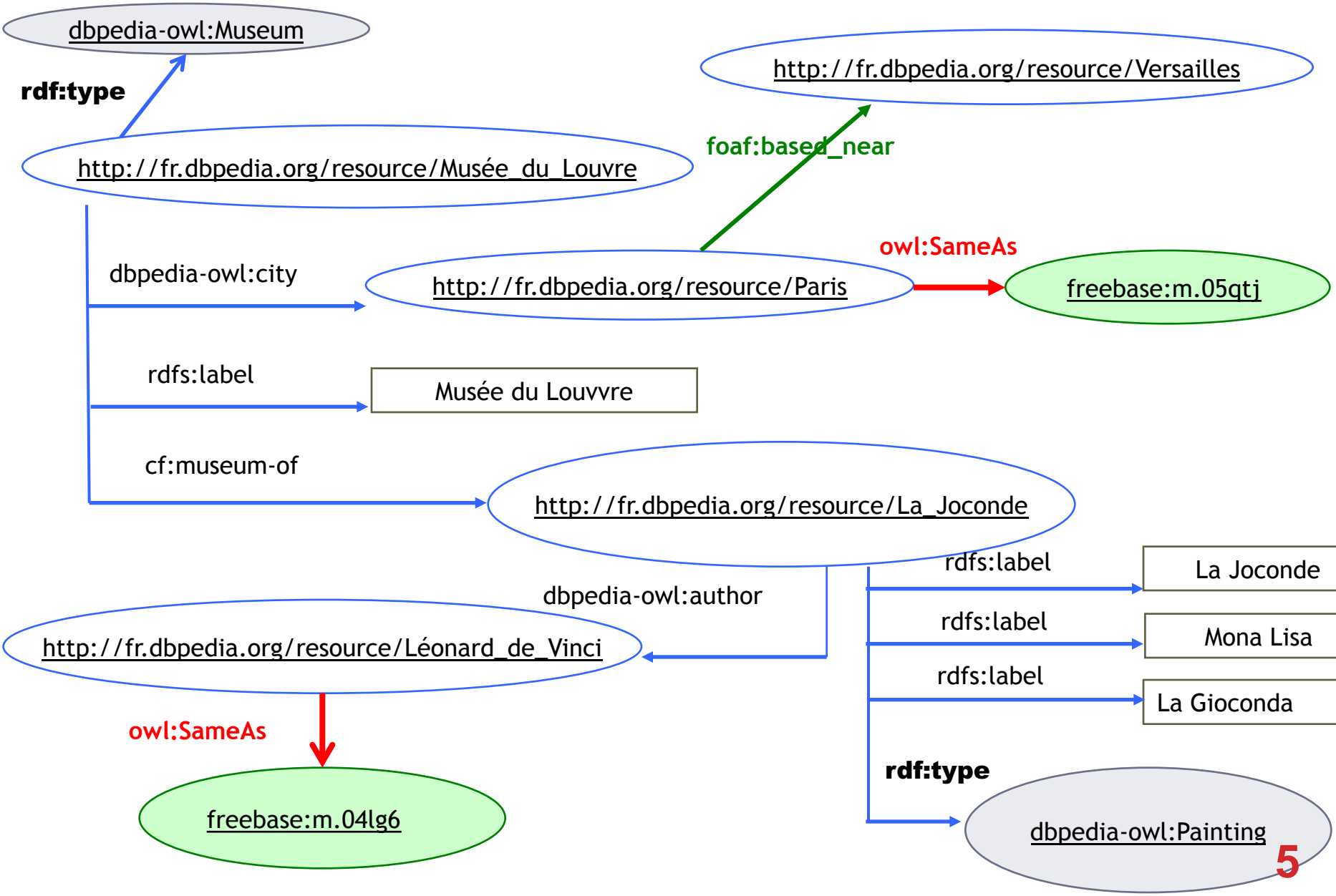
- **Statements of < subject predicate object >**



Subject Predicate Object

… is called a triple

Data linking: example



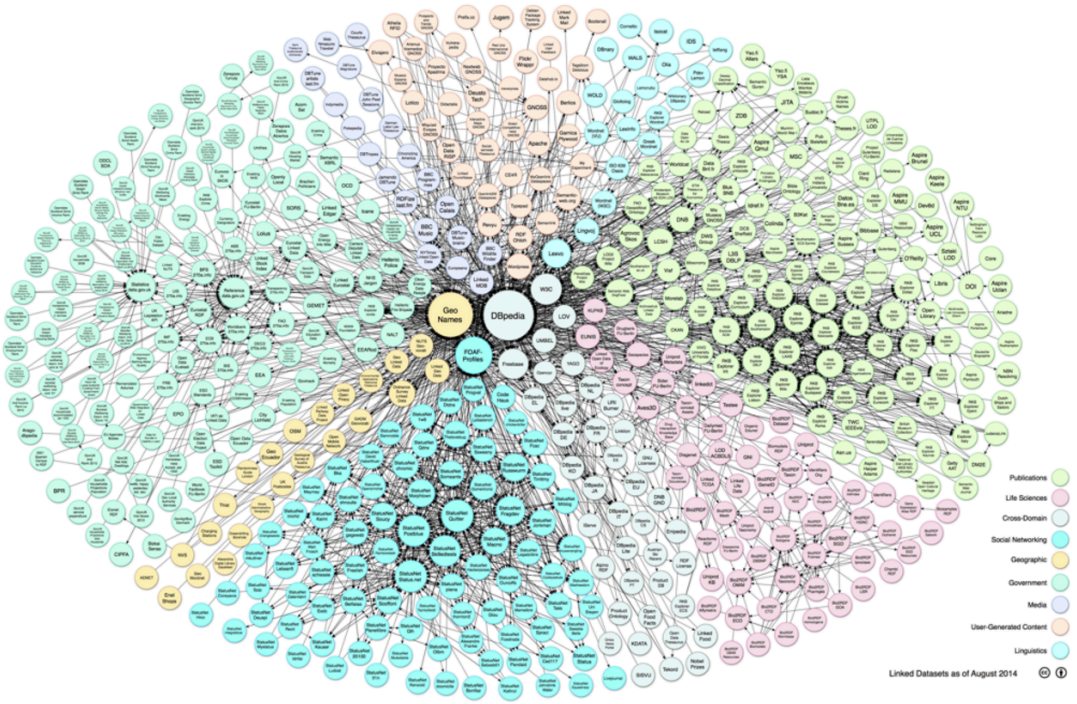
OUTLINE

- ❑ Introduction
- ❑ Part 1: Data linking
- ❑ Part 2: Key discovery
- ❑ Part 3: Data fusion
- ❑ Conclusion and some future challenges

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

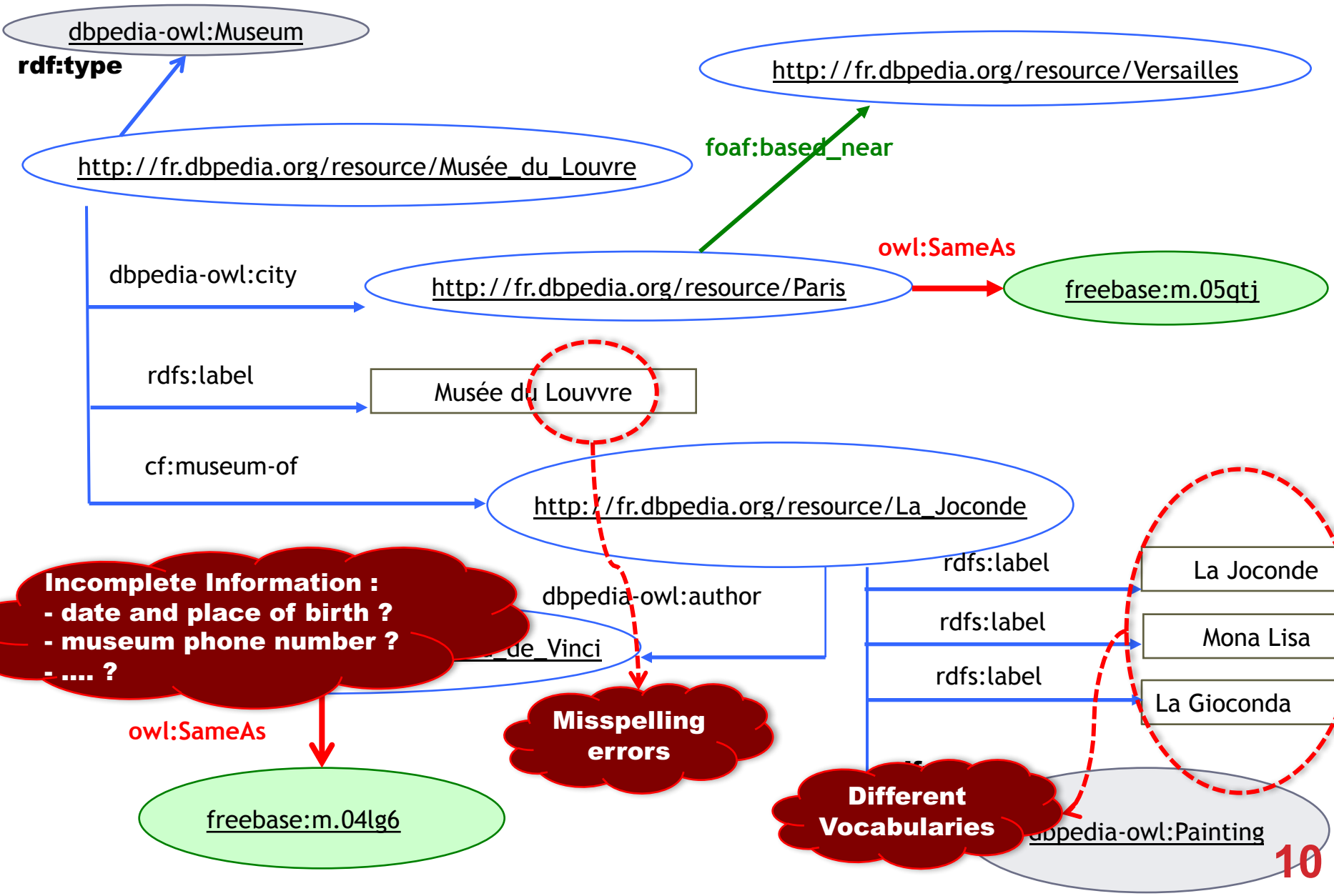
- **SameAs Link discovery** consists in detecting whether two descriptions of resources refer to the same real world entity (e.g. same person, same article, same gene).

- **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** ∈ $O(U_1 \times U_2)$, 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-

cord and be sign ring e of files 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

N2R: A NUMERICAL METHOD FOR REFERENCE RECONCILIATION

[Saïs et al. 2009]

N2R: A NUMERICAL METHOD FOR REFERENCE RECONCILIATION

- **N2R computes a similarity score for pair of references (instances) obtained from their common description.**
  - Uses known similarity measures, e.g. Jaccard, Jaro-Winkler.
  - Exploits ontology axioms: keys, functionality of properties and disjunction between classes.

SIMILARITY DEPENDENCY MODELLING

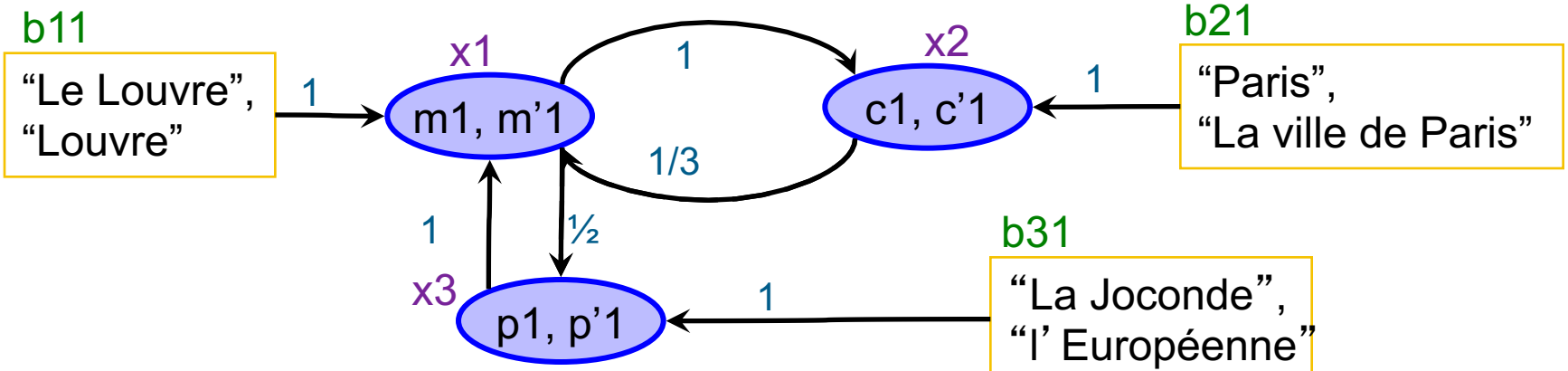
RDF facts in source S1:
Located(m1, c1), MuseumName(m1, “le Louvre”)
Contains(m1, p1), CityName(c1, “Paris”)
PaintingName(p1, “la Joconde”)

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

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

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

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

DF(x_i), considered in the maximum

NDF(x_i), considered in the average

➔ A non linear system

NON LINEAR EQUATION SYSTEM RESOLUTION

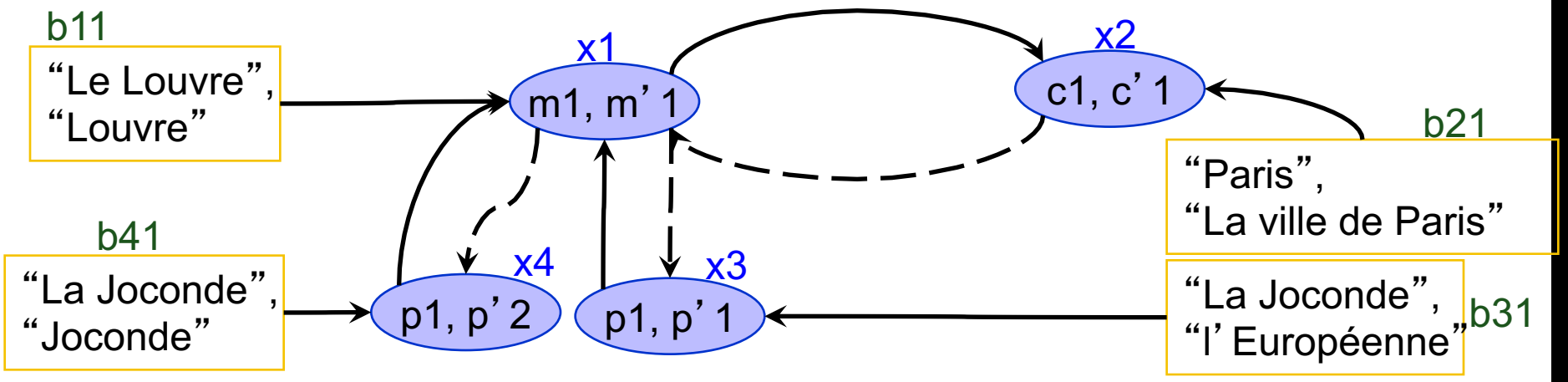
- **An iterative method inspired form *Jacobi*.**
  - Initialize the variable s x_i at 0.
  - Refine iteratively the value of each x_i by using the values x_i computed at a precedent iteration.

  - Termination: a fix-point with a precision ε

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

➔**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

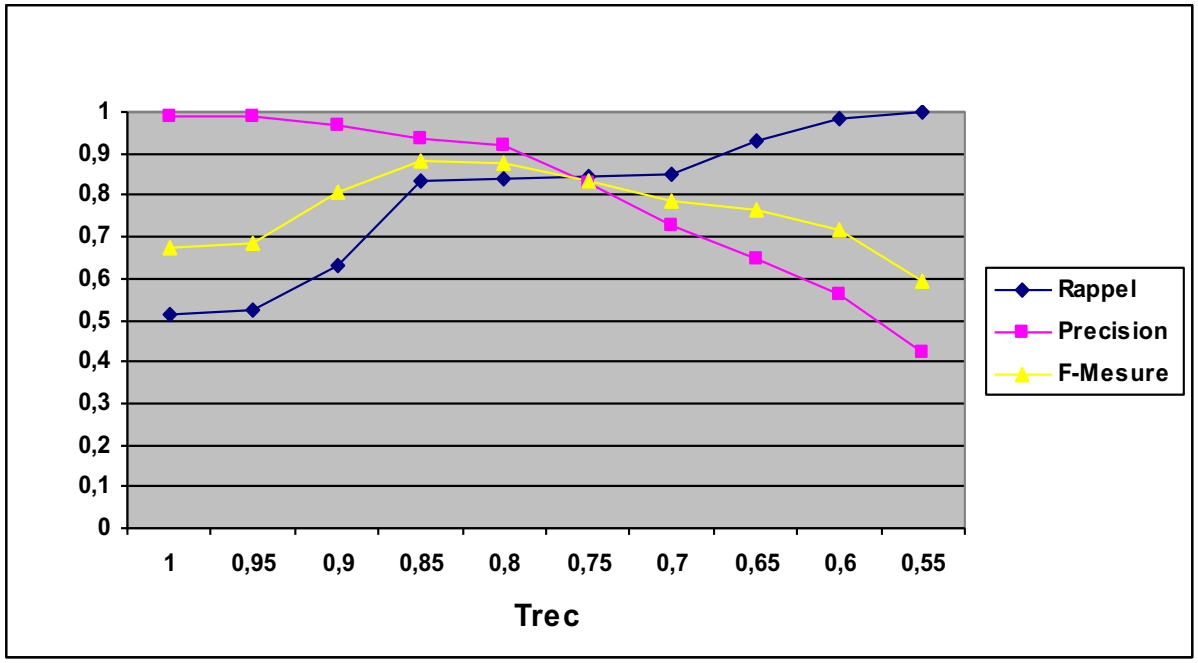
λ= 1/(| CAttr | + | CRel |) ε = 0.02

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

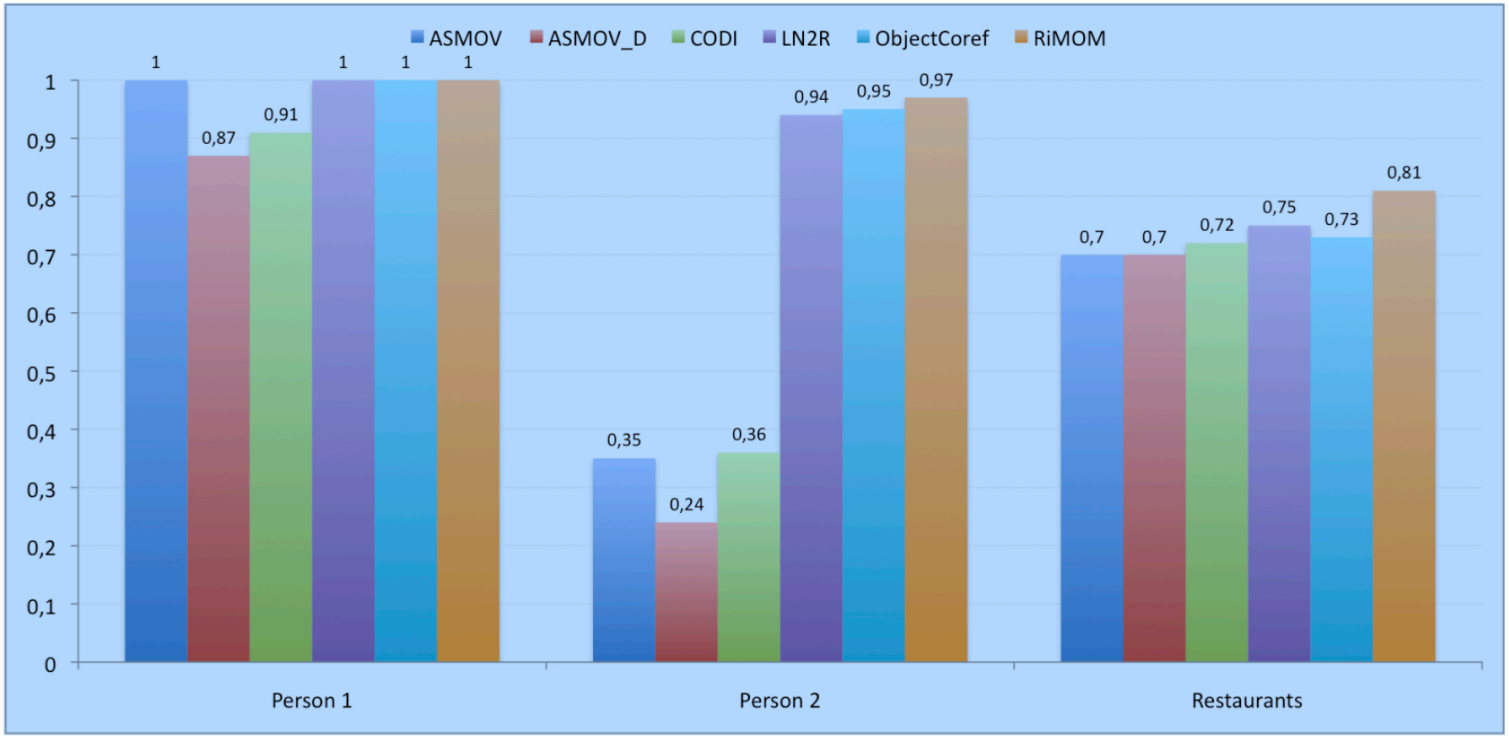
N2R: RESULTS ON CORA



- Trec=1 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]

N2R: RESULTS IN OAEI 2010

OAEI 2010 – Instance Matching track (PR), 2nd



OUTLINE

- ❑ Introduction
- ❑ Part 1: Data linking
- ❑ Part 2: Key discovery
- ❑ Part 3: 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:

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

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**
- **Data with errors**
- **Data with 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

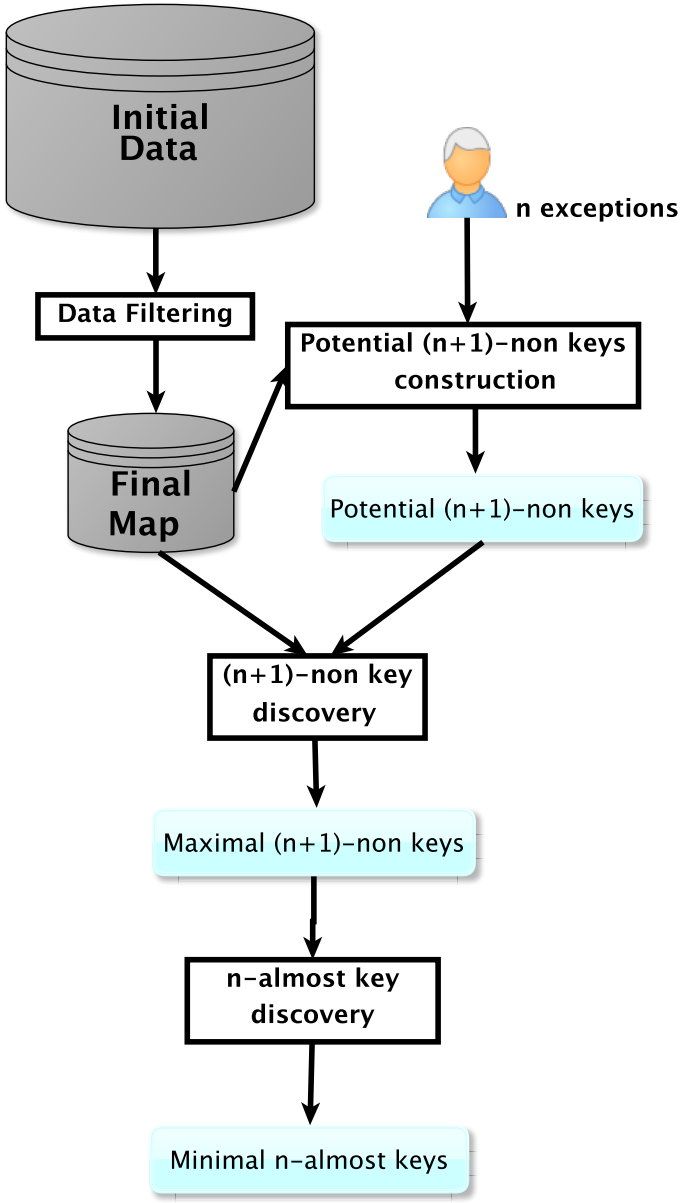
Discover sets of properties that are not *n*-almost keys first

- ***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

Antimonotonic pruning

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

Seen intersection pruning

- Avoiding already explored sets of instances

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)

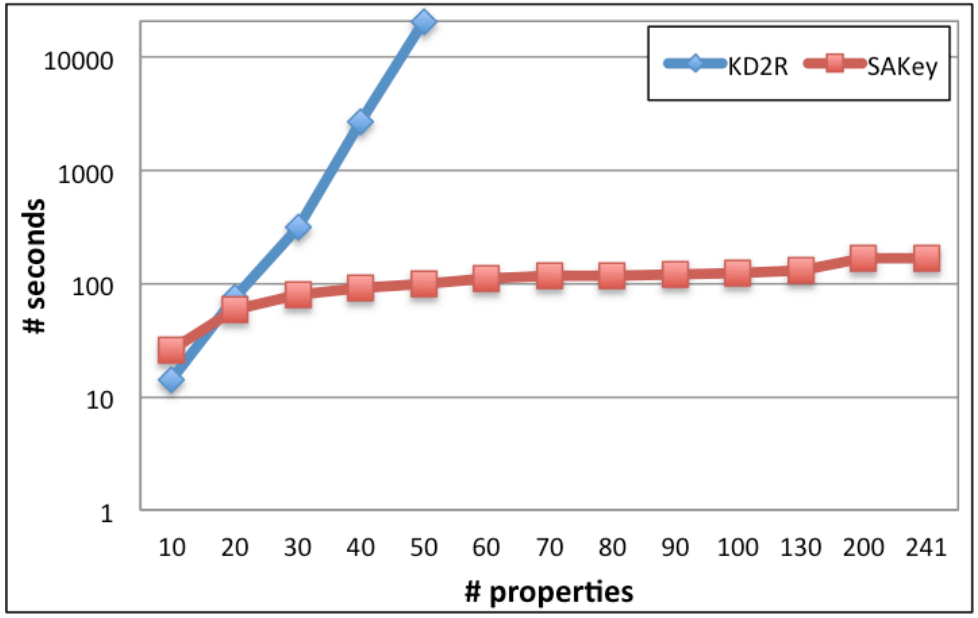
Conclusion

- 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

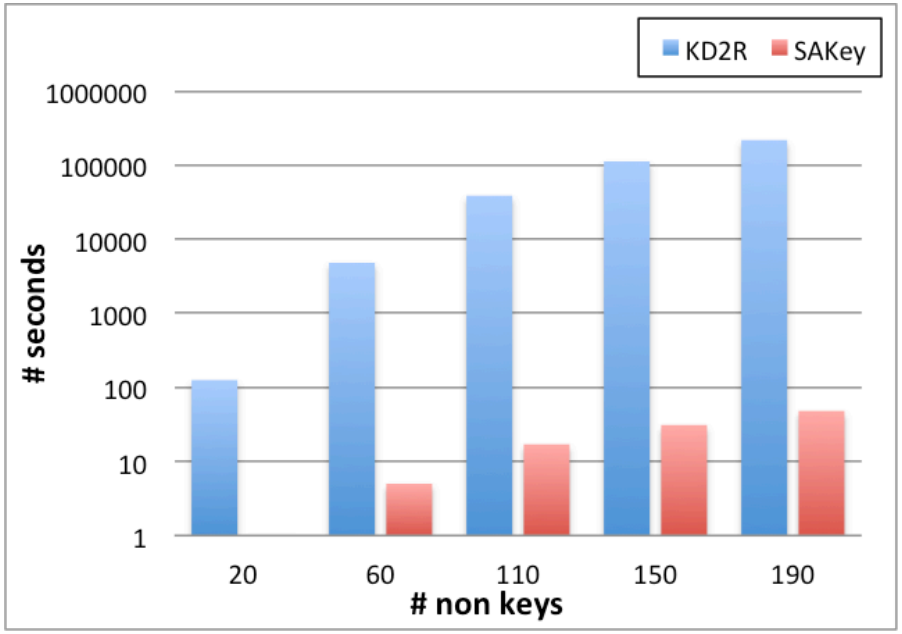
Dbpedia class= DB:NaturalPlace



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



OUTLINE

- ❑ Introduction
- ❑ Part 1: Data linking
- ❑ Part 2: Key discovery
- ❑ **Part 3: Data fusion**
- ❑ **Conclusion and some future challenges**

PART 3: DATA FUSION

DATA FUSION

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

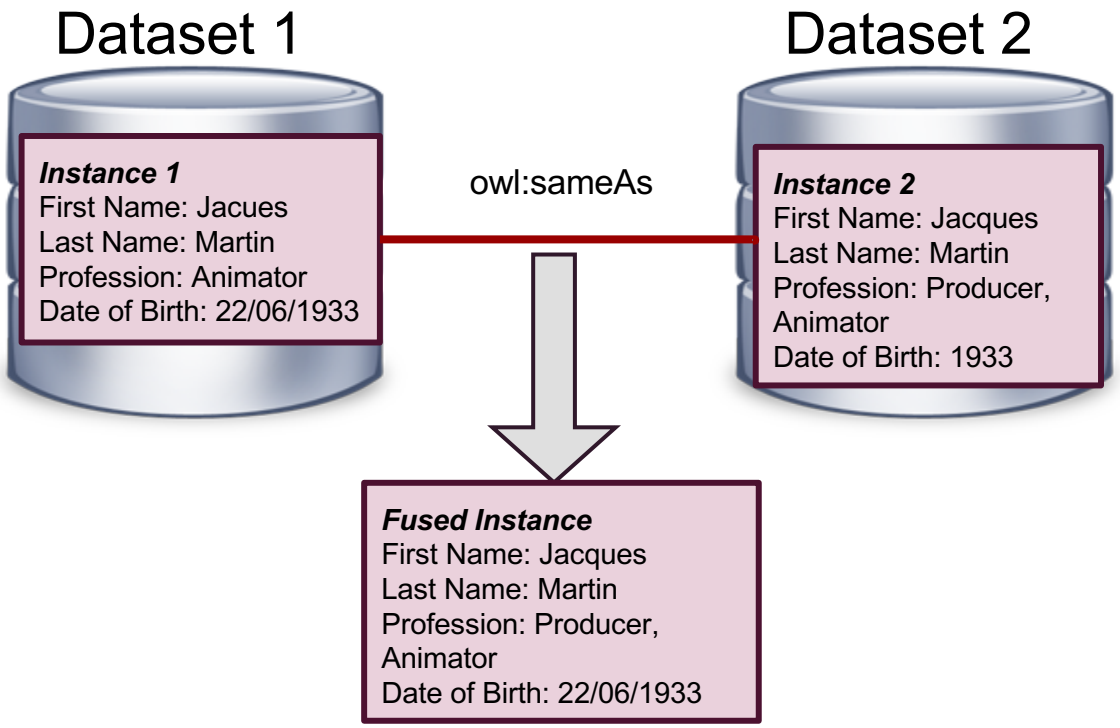
[Bleiholder & Naumann, 2008]

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 5

- Values selection

→ Sort values by quality score

→ Mono-valued: select best value

→ Multi-valued: all plausible values

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

KEEP TRACK OF FUSION DECISIONS

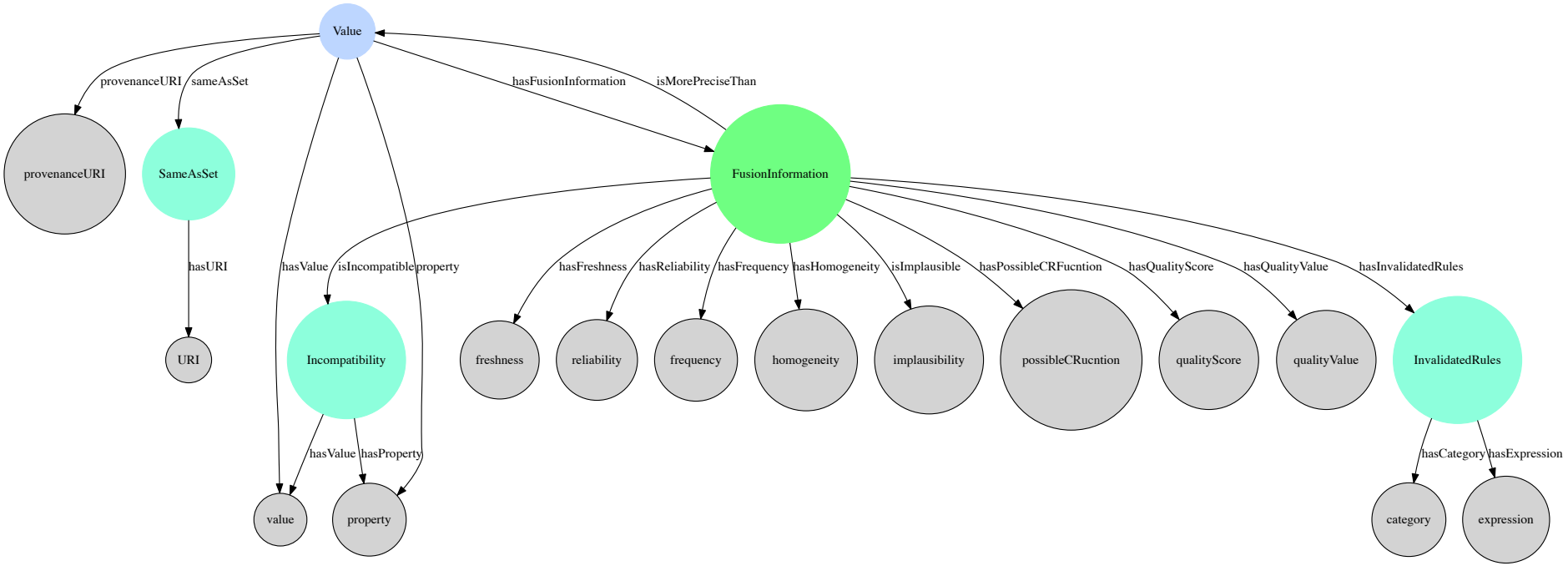
Why is a value selected?

How was the fusion decision taken?

The system stores all the quality aspects

Annotate values with quality information

THE ANNOTATION ONTOLOGY



EXAMPLE OF THREE BOOKS

@prefix ob : < https : //schema.org/Book > .

uri	ob :title	ob : nbPages	ob :author	ob :contributor	ob :publisher
b1	A Semantic Web Primer	238	Grigoris Antoniou	Paul Groth Frank V. Harmelen Rinke Hoekstra	The MIT Press (MA)
b2	A Semantic Web Primer	0	G. Antoniou	P. Groth F. V. Harmelen R. Hoekstra	MIT Press MA
b3	A Semantic Web Primer, second edition (cooperative information Systems Series	288	Grigoris Antoniou	Paul Groth Frank Van Harmelen Rinke Hoekstra	MIT Press Massachusetts

uri	ob :dateCreated	ob :dateModified	ob :datePublished	ob :keywords
b1	12/07/2007	01/05/2008	03/01/2008	Computer Science Knowledge representation Semantic Web
b2	December 7th 2007	April 30th 2004	March 1st 2008	Artificial Intelligence Description Logic Semantic Web
b3	December 2007	January 2008	March 2008	Semantic Web AI Knowledge representation & reasoning

USE OF ANNOTATION ONTOLOGY

#same-as-set-1 = {#books1-Book1, #books1-Book2, #books1-Book3}

- Data fusion result

<#title-val-12 has-same-as-set #same-as-set-1>

<#title-val-12 has-provenance-URI #books1-Book2>

<#title-val-12 has-value "**A Semantic Web Primer** " >

<#title-val-12 has-property **#title**>

<#title-val-12 has-fusion-information **#fusion-information-12**>

<fusion-information-12 has-homogeneity 0.66>

<fusion-information-12 has-occurence-frequency 0.66>

<fusion-information-12 has-source-freshness 1.0>

<fusion-information-12 has-source-reliability 1.0>

<fusion-information-12 is-implausible false>

<fusion-information-12 has-quality-score 0.83>

<fusion-information-12 has-quality-value "Excellent">

<#title-val-13 has-same-as-set #same-as-set-1>

<#title-val-13 has-provenance-URI #books1-Book2>

<#title-val-13 has-value **"A Semantic Web Primer, second edition (cooperative information Systems Series** " >

<#title-val-13 has-property **#title**>

<#title-val-13 has-fusion-information **#fusion-information-13**>

<fusion-information-13 has-homogeneity 0.33>

<fusion-information-13 has-occurence-frequency 0.33>

<fusion-information-13 has-source-freshness 1.0>

<fusion-information-13 has-source-reliability 1.0>

<fusion-information-13 is-implausible false>

<fusion-information-13 has-quality-score 0.73>

<fusion-information-13 has-quality-value "Good">

<fusion-information-13 is-more-recise-than #title-val-11>

<fusion-information-13 is-more-recise-than #title-val-12>

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
- ❑ Part 3: 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 requalification
  - **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

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