

# Several Link Keys Are Better than One, or Extracting Disjunctions of Link Key Candidates

Manuel Atencia   Jérôme David   Jérôme Euzenat



Laboratoire d'Informatique de Grenoble  
Montbonnot, France

`Firstname.Lastname@inria.fr`

`https://moex.inria.fr`

Partly funded by Elker ANR project (ANR-17-CE23-0007-01)

Data interlinking

Link keys

Link key candidates extraction (with FCA)

Extraction of disjunctions of link keys candidates

Experiments

## Data interlinking

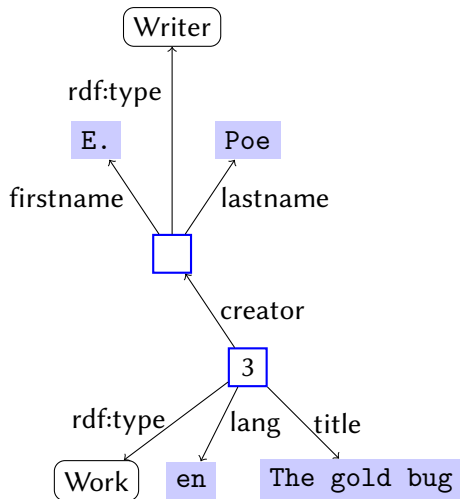
### Link keys

### Link key candidates extraction (with FCA)

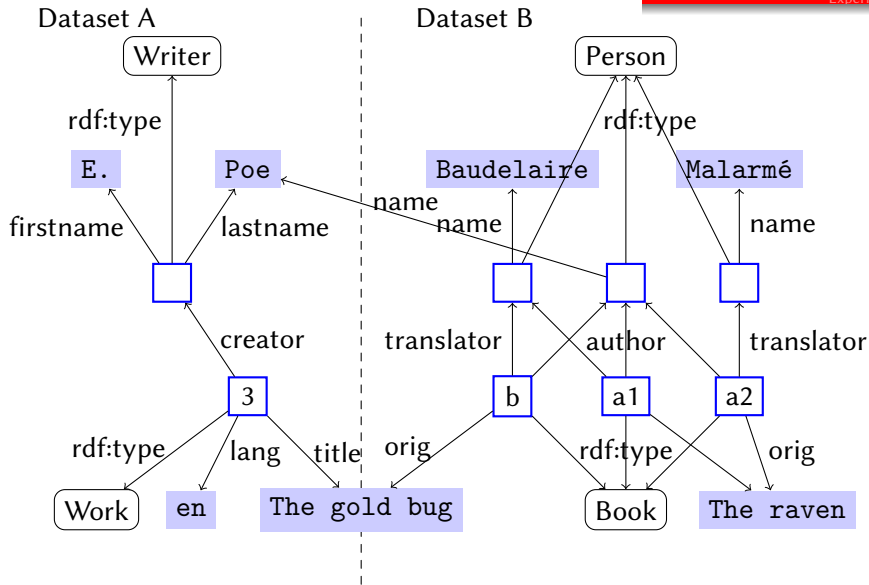
### Extraction of disjunctions of link keys candidates

### Experiments

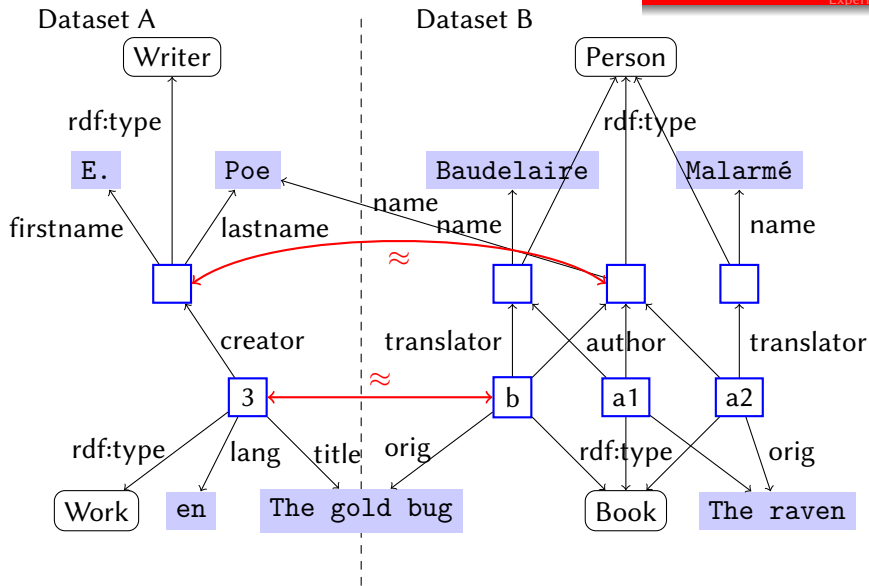
# The problem: RDF data interlinking



# The problem: RDF data interlinking



# The problem: RDF data interlinking



- ▶ NLP/IR based approaches
  - ▶ Change representation: from RDF space to VSM, or embedding spaces
  - ▶ Compute or learn a similarity on this new space
- ▶ Numerical specifications (Link Specifications)
  - ▶ Express or learn a similarity from RDF data
  - ▶ Generate links using frameworks such as SILK or LIMES
- ▶ Logical link specifications
  - ▶ Key-based: combine keys and alignments for deducing links
  - ▶ Link keys: can be extracted without requiring property alignment as input

- ▶ NLP/IR based approaches
  - ▶ Change representation: from RDF space to VSM, or embedding spaces
  - ▶ Compute or learn a similarity on this new space
- ▶ Numerical specifications (Link Specifications)
  - ▶ Express or learn a similarity from RDF data
  - ▶ Generate links using frameworks such as SILK or LIMES
- ▶ Logical link specifications
  - ▶ Key-based: combine keys and alignments for deducing links
  - ▶ **Link keys: can be extracted without requiring property alignment as input**



- ▶ There are models easy to interpret by humans
- ▶ They are logically grounded:
  - ▶ we can check consistency with the ontologies and data
  - ▶ subsumption between link keys or keys
- ▶ They produce links with high precision
  - ▶ ... but with limited recall

## Objective:

improve recall by considering combination of link keys

Data interlinking

Link keys

Link key candidates extraction (with FCA)

Extraction of disjunctions of link keys candidates

Experiments

Given two RDF dataset signatures:

$$D = \langle R, P, C \rangle \text{ and } D' = \langle R', P', C' \rangle.$$

R: object properties, P: datatype properties, C: classes

A *link key expression* has the form

$$\langle \{ \langle p_i, p'_i \rangle \}_{i \in EQ}, \{ \langle q_j, q'_j \rangle \}_{j \in IN}, \langle c, c' \rangle \rangle$$

such that:

- ▶  $p_i \in P \cup R, q_j \in P \cup R$  and  $c \in C$
- ▶  $p'_i \in P' \cup R', q'_j \in P' \cup R'$  and  $c' \in C'$
- ▶  $EQ$  and  $IN$  are (possibly empty) finite sets of indices

## A link key expression

$$\langle \{ \langle p_i, p'_i \rangle \}_{i \in EQ}, \{ \langle q_j, q'_j \rangle \}_{j \in IN}, \langle c, c' \rangle \rangle$$

is a *link key* iff the following holds:

For all pairs of instances  $o$  and  $o'$  belonging respectively to classes  $c$  and  $c'$ ,

**if**  $o$  and  $o'$  have the same sets of values (object) for each pairs of properties  $p_i$  and  $p'_i$  respectively,

**and**  $o$  and  $o'$  share at least one value (object) for each pairs of properties  $q_j$  and  $q'_j$  respectively,

**then** they are the same.

$$\text{if } \bigwedge_{i \in EQ} p_i(o) = p'_i(o') \neq \emptyset \text{ and } \bigwedge_{j \in IN} q_j(o) \cap q'_j(o') \neq \emptyset \\ \text{then } \langle o, \text{owl:sameAs}, o' \rangle$$

$D$ (Employés)					$D'$ (Staff)				
id	prenom	datenaiss	poste	bât.	firstname	birthdate	position	building	id
$i_2$	Paul	1967	Dir.	B2	Paul		Dir.	B2	$z_2$
$i_3$	Mary	1963	Dir.	B1	Mary		Dir.	B1	$z_3$
$i_4$	John	1963	Pr.	B1	John		Pr.	B1	$z_4$
$i_6$	Bill	1980	Pr.	B1	William	1980	Pr.		$z_6$
$i_7$	Ana	1947	Dir.	B2	Ana	1947	Dir.		$z_7$
$i_8$	John	1967	Pr.	B2	John	1967	Pr.		$z_8$

Example of link key expressions:

- ▶  $k = \langle \{\}, \{\langle \text{datenaiss}, \text{birthdate} \rangle\}, \langle \text{Employe}, \text{Staff} \rangle \rangle$
- ▶  $h = \langle \{\langle \text{datenaiss}, \text{birthdate} \rangle\}, \{\langle \text{poste}, \text{position} \rangle\} \langle \text{Employe}, \text{Staff} \rangle \rangle$
- ▶  $l = \langle \{\langle \text{datenaiss}, \text{birthdate} \rangle, \langle \text{poste}, \text{position} \rangle\}, \{\langle \text{poste}, \text{position} \rangle\}, \langle \text{Employe}, \text{Staff} \rangle \rangle$

And generated links:

- ▶  $L_k^{D,D'} = \{\langle i_7, z_7 \rangle, \langle i_8, z_8 \rangle, \langle i_6, z_6 \rangle, \langle i_2, z_8 \rangle\}$
- ▶  $L_l^{D,D'} = L_h^{D,D'} = \{\langle i_7, z_7 \rangle, \langle i_8, z_8 \rangle, \langle i_6, z_6 \rangle\}$

$D$ (Employés)					$D'$ (Staff)				
id	prenom	datenaiss	poste	bât.	firstname	birthdate	position	building	id
$i_2$	Paul	1967	Dir.	B2	Paul		Dir.	B2	$z_2$
$i_3$	Mary	1963	Dir.	B1	Mary		Dir.	B1	$z_3$
$i_4$	John	1963	Pr.	B1	John		Pr.	B1	$z_4$
$i_6$	Bill	1980	Pr.	B1	William	1980	Pr.		$z_6$
$i_7$	Ana	1947	Dir.	B2	Ana	1947	Dir.		$z_7$
$i_8$	John	1967	Pr.	B2	John	1967	Pr.		$z_8$

Example of link key expressions:

- ▶  $k = \langle \{\}, \langle \{\text{datenaiss, birthdate}\}\rangle, \langle \text{Employee, Staff} \rangle \rangle$
- ▶  $h = \langle \langle \{\text{datenaiss, birthdate}\}\rangle, \langle \{\text{poste, position}\}\rangle, \langle \text{Employee, Staff} \rangle \rangle$
- ▶  $l = \langle \langle \{\text{datenaiss, birthdate}\}\rangle, \langle \{\text{poste, position}\}\rangle, \langle \{\text{poste, position}\}\rangle, \langle \text{Employee, Staff} \rangle \rangle$

And generated links:

- ▶  $L_k^{D,D'} = \{ \langle i_7, z_7 \rangle, \langle i_8, z_8 \rangle, \langle i_6, z_6 \rangle, \langle i_2, z_8 \rangle \}$
- ▶  $L_l^{D,D'} = L_h^{D,D'} = \{ \langle i_7, z_7 \rangle, \langle i_8, z_8 \rangle, \langle i_6, z_6 \rangle \}$

$D$ (Employés)					$D'$ (Staff)					
id	prenom	datenaiss	poste	bât.		firstname	birthdate	position	building	id
$i_2$	Paul	1967	Dir.	B2		Paul		Dir.	B2	$z_2$
$i_3$	Mary	1963	Dir.	B1		Mary		Dir.	B1	$z_3$
$i_4$	John	1963	Pr.	B1		John		Pr.	B1	$z_4$
$i_6$	Bill	1980	Pr.	B1	←→	William	1980	Pr.		$z_6$
$i_7$	Ana	1947	Dir.	B2	←→	Ana	1947	Dir.		$z_7$
$i_8$	John	1967	Pr.	B2	←→	John	1967	Pr.		$z_8$

Example of link key expressions:

- ▶  $k = \langle \{\}, \langle \{\text{datenaiss, birthdate}\} \rangle, \langle \text{Employee, Staff} \rangle \rangle$
- ▶  $h = \langle \langle \{\text{datenaiss, birthdate}\} \rangle, \langle \{\text{poste, position}\} \rangle, \langle \text{Employee, Staff} \rangle \rangle$
- ▶  $l = \langle \langle \langle \{\text{datenaiss, birthdate}\} \rangle, \langle \{\text{poste, position}\} \rangle \rangle, \langle \{\text{poste, position}\} \rangle, \langle \text{Employee, Staff} \rangle \rangle$

And generated links:

- ▶  $L_k^{D,D'} = \{ \langle i_7, z_7 \rangle, \langle i_8, z_8 \rangle, \langle i_6, z_6 \rangle, \langle i_2, z_2 \rangle \}$
- ▶  $L_l^{D,D'} = L_h^{D,D'} = \{ \langle i_7, z_7 \rangle, \langle i_8, z_8 \rangle, \langle i_6, z_6 \rangle \}$

# Subsumption, meet and join of link key expressions

Let be two link key expressions over datasets  $D$  and  $D'$ :

$$k = \langle E, I, \langle c, c' \rangle \rangle \text{ and } h = \langle F, J, \langle c, c' \rangle \rangle$$

(intensional) subsumption

$$k \trianglelefteq h, \text{ if } E \subseteq F \text{ and } I \subseteq J$$

$$\implies \text{ if } k \trianglelefteq h, \text{ then } L_k^{D, D'} \supseteq L_h^{D, D'}, \text{ written } k \preceq^{D, D'} h$$

meet

$$k \Delta h = \langle E \cap F, I \cap J, \langle c, c' \rangle \rangle$$

$$\implies L_{k \Delta h}^{D, D'} \supseteq L_k^{D, D'} \cup L_h^{D, D'}$$

join

$$k \nabla h = \langle E \cup F, I \cup J, \langle c, c' \rangle \rangle$$

$$\implies L_{k \nabla h}^{D, D'} = L_k^{D, D'} \cap L_h^{D, D'}$$



Data interlinking

Link keys

Link key candidates extraction (with FCA)

Extraction of disjunctions of link keys candidates

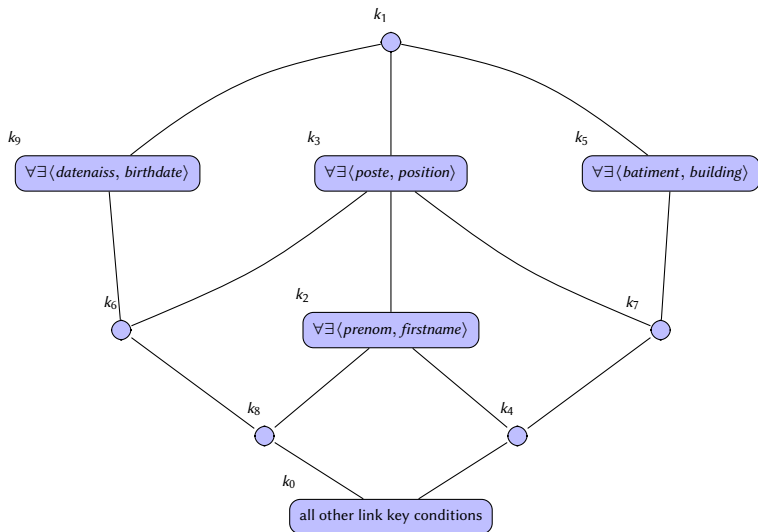
Experiments

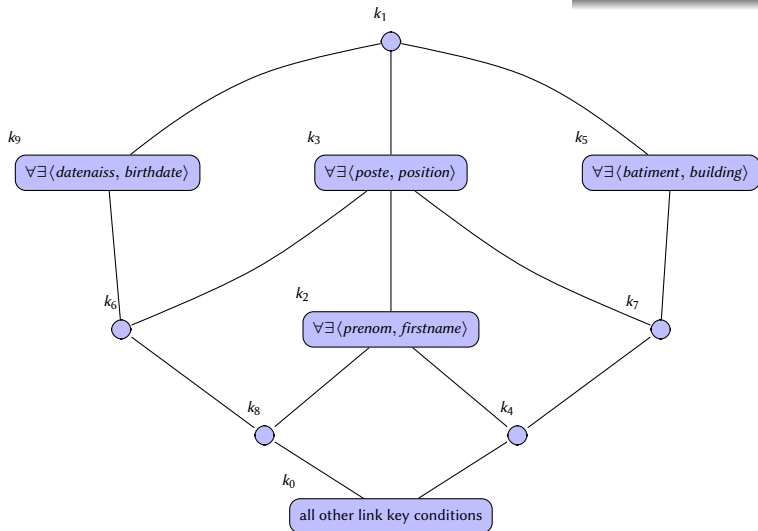
# Formal context for candidate link key extraction

- ▶ We provide a method for extracting link key candidates
  - ▶ subset of link key expressions that can generate links between the datasets
- ▶ It is based on Formal Concept Analysis

The *formal context for link key candidates*  $\langle G, M, I \rangle$  is:

G \ M	...	$\exists \langle p_i, p'_j \rangle$	...	...	$\forall \langle p_i, p'_j \rangle$	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\langle o, o' \rangle$	...	1 iff $p^D(o) \cap p'^{D'}(o') \neq \emptyset$	...	...	1 iff $p^D(o) = p'^{D'}(o')$	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮





How to select the "good" candidates ?

Let  $k = \langle E, I, \langle c, c' \rangle \rangle$  be a link key expression,

## Discriminability

$$\delta^{D,D'}(k) = \begin{cases} 1.0 & \text{if } L_k^{D,D'} = \emptyset \\ \frac{\min(|\pi(L_k^{D,D'})|, |\pi'(L_k^{D,D'})|)}{|L_k^{D,D'}|} & \text{otherwise} \end{cases}$$

## Coverage

$$\gamma^{D,D'}(k) = \begin{cases} 1.0 & \text{if } c^D = c'^{D'} = \emptyset \\ \frac{|\pi(L_k^{D,D'}) \cup \pi'(L_k^{D,D'})|}{|c^D \cup c'^{D'}|} & \text{otherwise} \end{cases}$$

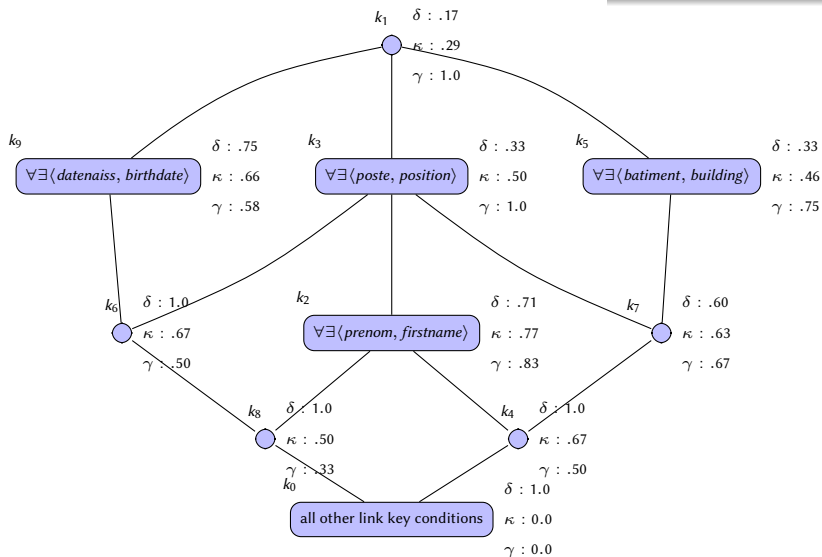
## hmean

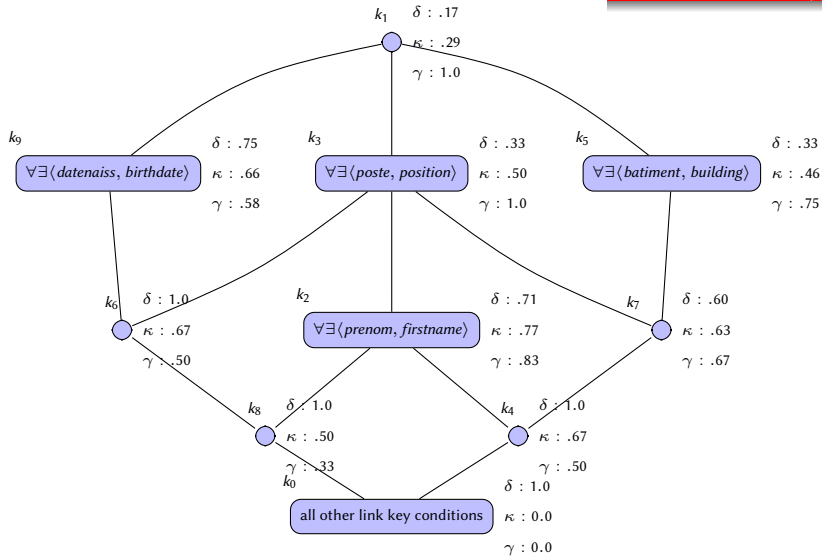
$$\kappa^{D,D'}(k) = \frac{2 \times \gamma^{D,D'}(k) \times \delta^{D,D'}(k)}{\gamma^{D,D'}(k) + \delta^{D,D'}(k)}$$

with  $\pi(L) = \{o \in D; \langle o, o' \rangle \in L\}$  and  $\pi'(L) = \{o' \in D'; \langle o, o' \rangle \in L\}$

If  $h \preceq^{D,D'} k$ , then  $\gamma^{D,D'}(h) \geq \gamma^{D,D'}(k)$

# Extracted link key candidates ( $\langle K, \trianglelefteq \rangle$ )





There is no perfect candidate link key

Data interlinking

Link keys

Link key candidates extraction (with FCA)

Extraction of disjunctions of link keys candidates

Experiments



## Conjunction of link key expressions

Notation:  $k \wedge h$

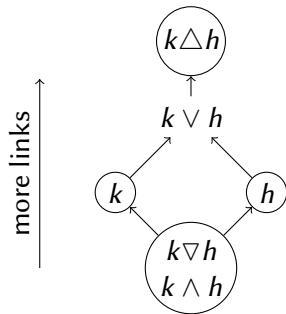
Link set:  $L_{k \wedge h}^{D, D'} = L_k^{D, D'} \cap L_h^{D, D'}$

## Disjunction of link key expressions

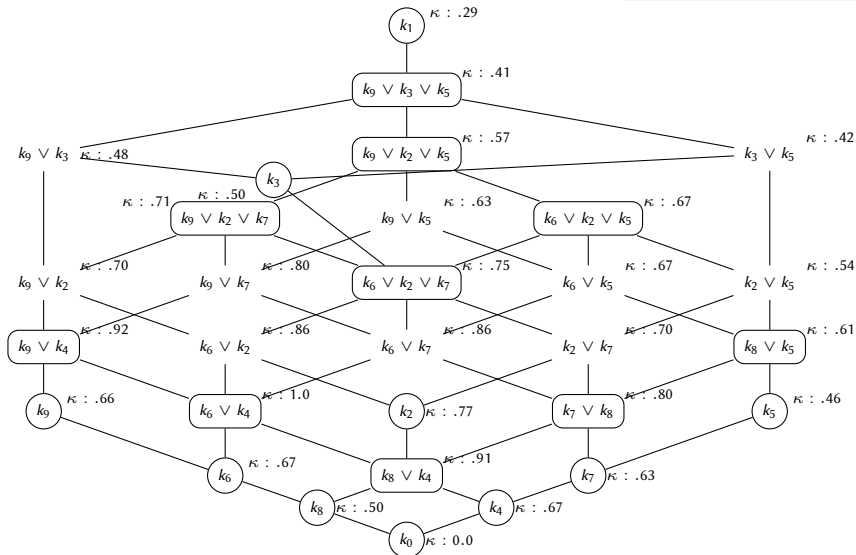
Notation:  $k \vee h$

Link set:  $L_{k \vee h}^{D, D'} = L_k^{D, D'} \cup L_h^{D, D'}$

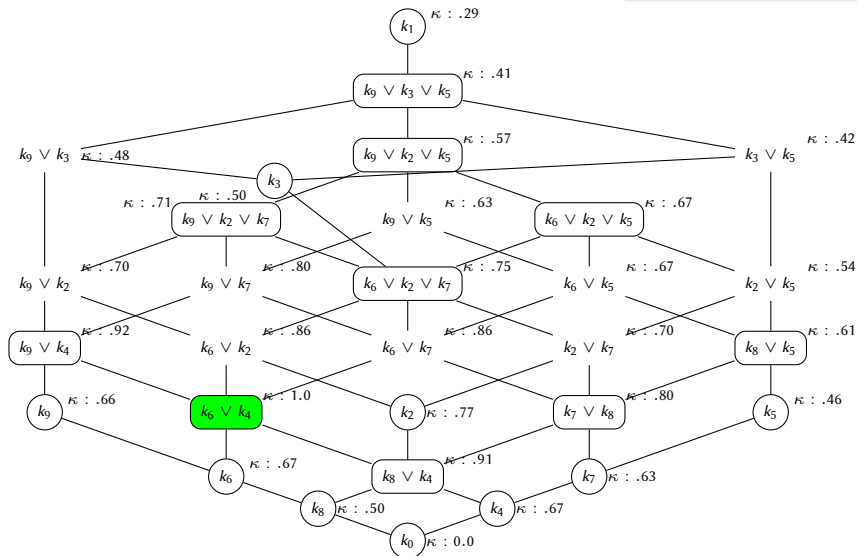
- ▶  $\vee$  and  $\wedge$  are commutative and associative,
- ▶  $\wedge$  is equivalent to  $\nabla$ : we only need to search for  $\vee$ .



1. Compute the link key candidates lattice
2. Enumerate antichains of link key candidates
3. Select the best one



# Antichain lattice ( $\langle K_{\subseteq}^{\vee}, \supseteq \rangle$ )



In our example:

- ▶ 10 candidates
- ▶ 30 antichains (12 maximal antichains)

In our example:

- ▶ 10 candidates
- ▶ 30 antichains (12 maximal antichains)

But...

- ▶ The number of antichains of a lattice is difficult to establish a priori, the worst case being  $2^n$
- ▶ The best disjunction does not necessarily contains the best link key candidate (with respect to  $\kappa$ )
- ▶ The best disjunction is not necessarily a maximal one

In our example:

- ▶ 10 candidates
- ▶ 30 antichains (12 maximal antichains)

But...

- ▶ The number of antichains of a lattice is difficult to establish a priori, the worst case being  $2^n$
- ▶ The best disjunction does not necessarily contains the best link key candidate (with respect to  $\kappa$ )
- ▶ The best disjunction is not necessarily a maximal one

So we cannot perform an exhaustive search...we need heuristics.

Two heuristics:

- ▶ *top-k*:
  1. select the top- $k$  candidates according to some evaluation measure ( $\kappa$ ),
  2. perform an exhaustive enumeration of antichains on this selection
- ▶ *expand-best*:
  1. Select the best antichain (starting from the atomic ones),
  2. Replace it by its expansion (all antichains containing this one),
  3. Stop the process after  $x$  iterations without improvement,
  4. Return the best antichain.



Data interlinking

Link keys

Link key candidates extraction (with FCA)

Extraction of disjunctions of link keys candidates

Experiments

## Hypothesis:

*Disjunctions of link key candidates generate better link sets, in terms of F-measure, than single link key candidates.*

## Datasets:

- ▶ Persons and Restaurants datasets (OAEI 2010)
- ▶ Doremus datasets (OAEI 2016)
- ▶ SPIMBench (OAEI 2018)
- ▶ Libraries

## Settings:

- ▶ Candidates extracted with Linkex using inverse, 2-length composition of properties
- ▶ Basic normalization of strings: remove diacritics, tokenize strings and sort the resulting bag of tokens
- ▶ Disjunctions extracted with top- $k$  ( $k = 10 \dots 30$ ,  $step = 5$ ) and expand-best strategies

## OAEI2010

Task	Single candidates				Disjunctions			
	#cand	Prec.	F-meas.	Rec.	Strategy	Prec.	F-meas.	Rec.
Restaurants	20	0.477	0.58	0.741	top-10	0.483	0.596	0.777
					expand-best	0.481	0.594	0.777
Person1	613	1	0.974	0.95	top-10	1	1	1
					expand-best	1	1	1
Person2	521	0.206	0.27	0.39	top-10	0.348	0.425	0.545
					expand-best	0.265	0.369	0.608

## Doremus (OAEI 2016)

Task	Single candidates				Disjunctions			
	#cand	Prec.	F-meas.	Rec.	Strategy	Prec.	F-meas.	Rec.
Doremus 1	27	0.833	0.714	0.625	top-10	0.793	0.754	0.719
					expand-best	0.806	0.794	0.781
Doremus 2	101	0.833	0.712	0.622	top-10	0.829	0.799	0.771
					expand-best	0.830	0.802	0.776
Doremus 3	38	0.622	0.571	0.683	top-10	0.569	0.667	0.805
					expand-best	0.596	0.694	0.829

## SPIMBench (OAEI 2018)

Task	Single candidates				Disjunctions			
	#cand	Prec.	F-meas.	Rec.	Strategy	Prec.	F-meas.	Rec.
SPIMBench	2 277	0.816	0.794	0.773	top-10	0.816	0.794	0.773
					expand-best	0.805	0.788	0.773

## Libraries (only partial reference)

Task	Single candidates				Disjunctions			
	#cand	Prec.	F-meas.	Rec.	Strategy	Prec.	F-meas.	Rec.
Libraries	933	0.656	0.614	0.578	top-10	0.563	0.616	0.679
					expand-best	0.363	0.474	0.681

- ▶ Top-10 strategy always find a disjunction better than (or equals to) the best single link key candidate
- ▶ Expand-best strategy always generates longer disjunctions than the top-10 strategy



In general disjunctions of link keys improve single results (F-measure)

- ▶ Definition of disjunction of link keys and semantics
- ▶ Relations between disjunctions and other link key expression
- ▶ Provide extraction strategies for extraction of disjunctions
- ▶ Fully unsupervised: no training sets nor alignments

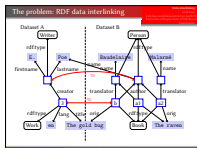
- ▶ Better evaluation measures
  - ▶ the measures are not always able to select the best disjunction
  - ▶ monotonic measures
- ▶ More complete and efficient extraction strategies
- ▶ Beyond disjunction composition

Several Link Keys Are Better than One, or  
Extracting Disjunctions of Link Key Candidates

Marcus Atencia Jérôme David Jérôme Euzenat

   
Laboratoire d'Informatique de Caen  
Normandie, France  
Fédération Française de  
Recherche en Informatique de  
Normandie (FFRIN)

Partly funded by Elcar ANR project (ANR-17-CE23-0007-01)



Link key

A link key expression  

$$\{(\{a_i, p_i\})_{i \in I_1}, \{a_j, q_j\}_{j \in I_2}\}$$
 is a link key iff the following holds:

For all pairs of instances  $a$  and  $a'$  of belonging respectively to classes  $c$  and  $c'$ ,

- $a$  and  $a'$  have the same sets of values (object) for each pair of properties  $p_i$  and  $q_j$  respectively,
- $a$  and  $a'$  share at least one value (object) for each pair of properties  $p_i$  and  $q_j$  respectively,
- that they are the same.

$\# \exists_{a, a'} (a \neq a' \wedge \bigwedge_{i \in I_1} p_i(a) = p_i(a') \wedge \bigwedge_{j \in I_2} q_j(a) = q_j(a') \wedge \bigvee_{i \in I_1} p_i(a) \neq p_i(a') \vee \bigvee_{j \in I_2} q_j(a) \neq q_j(a')$

Example

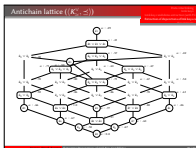
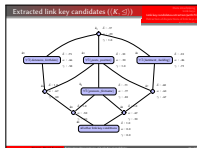
	D1 (Shakespeare)				D2 (Shall)			
id	property	object	value	instance	property	object	value	instance
1	has	1562	1562	Shakespeare	has	1562	1562	Shakespeare
2	has	1562	1562	Shakespeare	has	1562	1562	Shakespeare
3	has	1562	1562	Shakespeare	has	1562	1562	Shakespeare
4	has	1562	1562	Shakespeare	has	1562	1562	Shakespeare
5	has	1562	1562	Shakespeare	has	1562	1562	Shakespeare
6	has	1562	1562	Shakespeare	has	1562	1562	Shakespeare
7	has	1562	1562	Shakespeare	has	1562	1562	Shakespeare
8	has	1562	1562	Shakespeare	has	1562	1562	Shakespeare
9	has	1562	1562	Shakespeare	has	1562	1562	Shakespeare
10	has	1562	1562	Shakespeare	has	1562	1562	Shakespeare

Example of link key expressions:

- $\{a = \{1\}, \{a \text{ knows } b\} \text{ has } \{a\}\}$  (Shakespeare, Shall)
- $\{a = \{1\}, \{a \text{ knows } b\} \text{ has } \{a\}, \{a \text{ knows } c\} \text{ has } \{a\}\}$  (Shakespeare, Shall)
- $\{a = \{1\}, \{a \text{ knows } b\} \text{ has } \{a\}, \{a \text{ knows } c\} \text{ has } \{a\}, \{a \text{ knows } d\} \text{ has } \{a\}\}$  (Shakespeare, Shall)

Not link key expressions:

- $\{a = \{1\}, \{a \text{ knows } b\} \text{ has } \{a\}, \{a \text{ knows } c\} \text{ has } \{a\}\}$  (Shakespeare, Shall)
- $\{a = \{1\}, \{a \text{ knows } b\} \text{ has } \{a\}, \{a \text{ knows } c\} \text{ has } \{a\}, \{a \text{ knows } d\} \text{ has } \{a\}\}$  (Shakespeare, Shall)



QAED010 & Doremus Datasets

Task	Single candidates			Disjunctions		
	Recall	Prec.	F-score	Rec.	Prec.	F-score
Restaurants	20	0.477	0.38	0.741	0.51	0.504
	Top 10	0.51	0.316	0.377	0.51	0.504
Person1	0.13	1	0.174	0.94	1	1
	Top 10	1	1	1	1	1
Person2	121	0.204	0.27	0.34	0.143	0.125
	Top 10	0.143	0.125	0.143	0.143	0.143

Task	Single candidates			Disjunctions		
	Recall	Prec.	F-score	Rec.	Prec.	F-score
Chessmen	27	0.613	0.716	0.625	0.701	0.716
	Top 10	0.613	0.716	0.625	0.701	0.716
Chessmen 2	101	0.633	0.716	0.625	0.701	0.716
	Top 10	0.633	0.716	0.625	0.701	0.716
Chessmen 3	18	0.622	0.571	0.686	0.594	0.625
	Top 10	0.622	0.571	0.686	0.594	0.625



<https://moex.inria.fr>

Manuel . Atencia

Jerome . David @ inria . fr

Jerome . Euzenat