Three approaches to find definitions in RDF data

Justine Reynaud, Yannick Toussaint and Amedeo Napoli

LORIA (Université de Lorraine, INRIA, CNRS), Vandœuvre-les-Nancy, France

FCA4AI — July 14, 2018

Introduction

From RDF assertions, such as

```
Nancy in France   Paris in France
Nancy a City      Paris a City

Rome in Italy     Le_Louvre in France
Rome a City       Le_Louvre a Museum
```

French_Cities = {Paris, Nancy}

How to infer *definitions* in order to *complete* web of data ?

$$\texttt{French\_Cities} \equiv (\mathrm{a},\ \texttt{City}) \sqcap (\texttt{in},\ \texttt{France})$$

We compare three algorithms with different approaches.

Data representation

```
Nancy in France    Paris in France
Nancy a City       Paris a City
Rome in Italy      Le_Louvre in France
Rome a City        Le_Louvre a Museum
```

	(in, France)	(in, Italy)	(a, City)	(a, Museum)
Nancy	×		×	
Rome		×	×	
Paris	×		×	
Le_Louvre	×			×

Data representation

```
Nancy in France
Nancy a City
Rome in Italy
Rome a City
```

```
Paris in France
Paris a City
Le_Louvre in France
Le_Louvre a Museum
```

French_Cities = {Paris, Nancy}
Museums_in_Paris = {Le_Louvre}
European_Capital = {Paris, Rome}

	(in, France)	(in, Italy)	(a, City)	(a, Museum)	French_Cities	Museums_in_Paris	European_Capital
Nancy	×		×		×		
Rome		×	×				×
Paris	×		×		×		×
Le_Louvre	×			×		×	

Data representation

```
Nancy in France     Paris in France
Nancy a City        Paris a City
Rome in Italy       Le_Louvre in France
Rome a City         Le_Louvre a Museum
```

French_Cities = {Paris, Nancy}
Museums_in_Paris = {Le_Louvre}
European_Capital = {Paris, Rome}

	(in, France)	(in, Italy)	(a, City)	(a, Museum)	French_Cities	Museums_in_Paris	European_Capital
Nancy	×		×		×		
Rome		×	×				×
Paris	×		×		×		×
Le_Louvre	×			×		×	

$$\{\texttt{Nancy}\}' = \{(\texttt{in}, \texttt{France}),\ (\texttt{a}, \texttt{City}),\ \texttt{French\_Cities}\}$$

$$\{(\texttt{in}, \texttt{France}),\ (\texttt{a}, \texttt{City})\}' = \{\texttt{Nancy},\ \texttt{Paris}\}$$

Association rules – Eclat [Zaki, 2000]

- Searching for dependencies between sets of attributes
- Quality measure based on confidence

$$conf(X \rightarrow Y) = \frac{|\, X' \cap Y' \,|}{|\, X' \,|}$$

- Rules are unidirectional
- Post-processing in order to select rules satisfying criteria

Quasi-definition

A quasi-definition $X \leftrightarrow Y$ holds with a confidence θ iff

$$min(conf(X \rightarrow Y), conf(Y \rightarrow X)) = \theta$$

Redescriptions – ReReMi [Galbrun and Miettinen, 2012]

- Searching for two sets of attributes that occurs in the same objects
- Quality measure based on Jaccard coefficient

$$Jacc(X \leftrightarrow Y) = \frac{| X' \cap Y' |}{| X' \cup Y' |}$$

- Rules are bidirectional

Redescriptions – ReReMi [Galbrun and Miettinen, 2012]

- Searching for two sets of attributes that occurs in the same objects
- Quality measure based on Jaccard coefficient

$$Jacc(X \leftrightarrow Y) = \frac{\mid X' \cap Y' \mid}{\mid X' \cup Y' \mid}$$

- Rules are bidirectional

	FR	IT	City	Museum	FC	MP	EC
Nancy	×		×		×		
Rome		×	×				×
Paris	×		×		×		×
Le_Louvre	×			×		×	

(in, France) $\leftrightarrow$ French_Cities

Redescriptions – ReReMi [Galbrun and Miettinen, 2012]

- Searching for two sets of attributes that occurs in the same objects
- Quality measure based on Jaccard coefficient

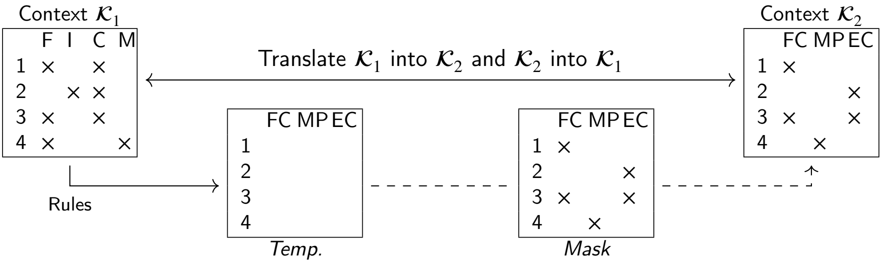
$$Jacc(X \leftrightarrow Y) = \frac{\mid X' \cap Y' \mid}{\mid X' \cup Y' \mid}$$

- Rules are bidirectional

	FR	IT	City	Museum	FC	MP	EC
Nancy	×		×		×		
Rome		×	×				×
Paris	×		×		×		×
Le_Louvre	×			×		×	

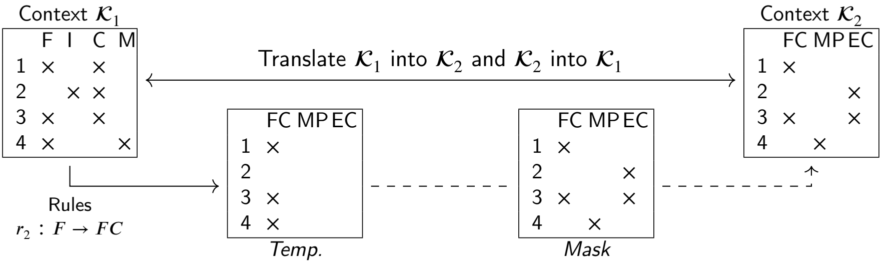
(in, France), (a, City) $\leftrightarrow$ French_Cities

Translation rules – Translator [van Leeuwen and Galbrun, 2015]



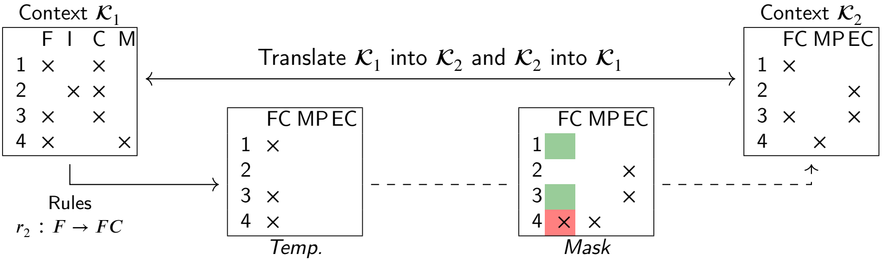
- Searching for rules that allow to construct one context from the other
- Rules may be unidirectional or bidirectional

Translation rules – Translator [van Leeuwen and Galbrun, 2015]



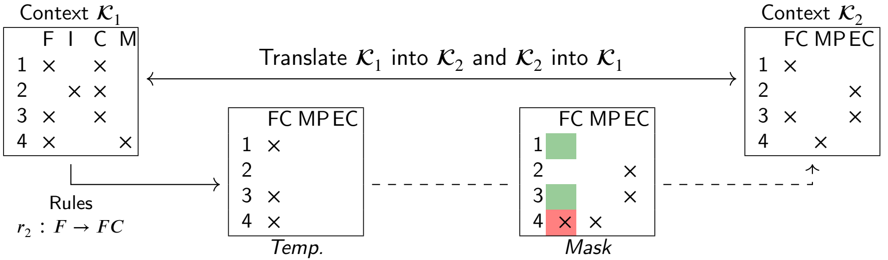
- Searching for rules that allow to construct one context from the other
- Rules may be unidirectional or bidirectional

Translation rules – Translator [van Leeuwen and Galbrun, 2015]



- Searching for rules that allow to construct one context from the other
- Rules may be unidirectional or bidirectional

Translation rules – Translator [van Leeuwen and Galbrun, 2015]



- Searching for rules that allow to construct one context from the other
- Rules may be unidirectional or bidirectional
- Adds the best rule at each step
- Quality metric inspired from *minimum description length* (MDL)

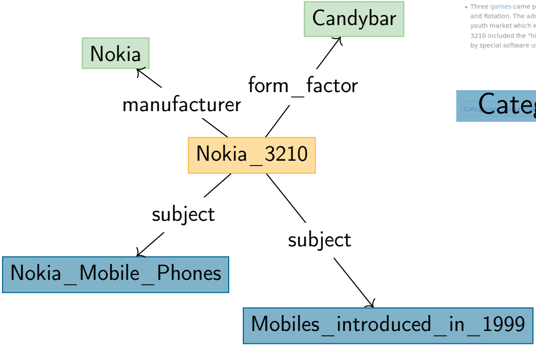
$$\Delta(X \rightarrow Y) = \underbrace{L(Mask^-) - L(Mask^+)}_{\text{Information gain}} - \underbrace{L(X \cup Y)}_{\text{Rule length}} \qquad L(X) = -\sum_{x \in X} \log_2 P(x \mid \mathcal{K})$$

Algorithms: Comparison

	Eclat	ReReMi	Translator
Data	Bool.	Bool., Num., Cat.	Bool.
Quality measure	Confidence $\frac{\lvert X' \cap Y' \rvert}{\lvert X' \rvert}$	Jaccard $\frac{\lvert X' \cap Y' \rvert}{\lvert X' \cup Y' \rvert}$	Compression based on MDL
Symmetric rule	No	Yes	Both

- Eclat needs a post-process to build bi-directional rules
- ReReMi and Eclat compute confidence in a very similar way
  ReReMi should return a subset of the rules found by Eclat
- Translator aims to mine a good set of rules instead of a set of good rules

From Wikipedia to DBpedia



Experiment

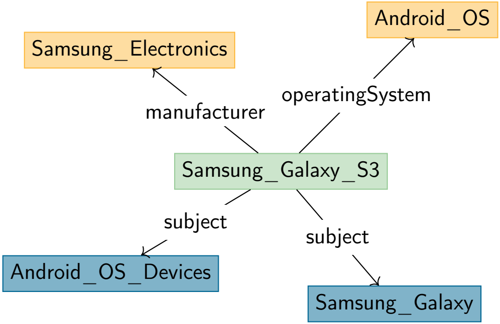
Datasets Triples from four domains of DBpedia

Turing_Award_laureates, Smartphones, Sports_cars, French_Films

Objects Subjects of the triples

Categories Pairs $(subject, C)$ from the categories

Descriptions Pairs (p, o) from the other triples



Smartphones

8500 Triples
600 Resources
400 Categories
1800 Descriptions

Results

R **Samsung_Galaxy**
(manufacturer Samsung_Electronics), (operatingSystem Android_(operating_system))

ET **Samsung_Galaxy, Samsung_mobile_phones, Smartphones**
(a Device), (manufacturer Samsung_Electronics), (operatingSystem Android_(operating_system))

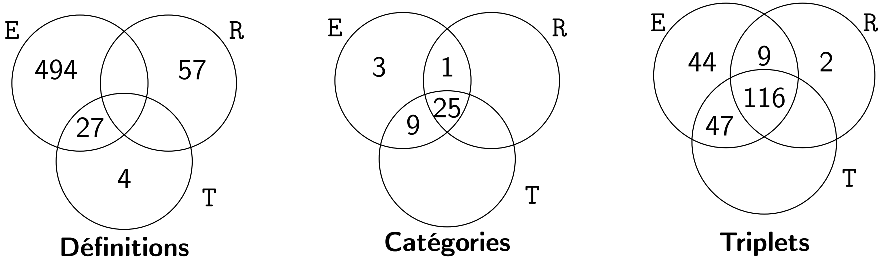
`Smartphones`

X	`E`	`R`	`T`
$\lvert\mathcal{R}^X\rvert$	810	98	41
$\lvert\mathcal{D}^X\rvert$	521	57	31
Précision	.64	.58	.76
$\overline{\lvert C_i\rvert} - \overline{\lvert D_i\rvert}$	4.3	1.6	3.1
$\lvert C_i\rvert - \lvert D_i\rvert$	7.8	1.8	3.1

Results

R **Samsung_Galaxy**
(manufacturer Samsung_Electronics), (operatingSystem Android_(operating_system))

ET **Samsung_Galaxy, Samsung_mobile_phones, Smartphones**
(a Device), (manufacturer Samsung_Electronics), (operatingSystem Android_(operating_system))



Future Work

How to include domain knowledge like classes and/or predicates hierarchy?

i.e. dealing with a partial order on the attributes

Can we find class disjointness instead of definitions?

i.e. searching for rules with a very low quality measure

How to deal with scalability?

i.e. evaluating a huge amount of rules

Thanks. Questions ?

Eclat

I want to be exhaustive, no matter if they're is a lot of equivalent rules.

ReReMi

I want a few rules easy to interpret and it's important they're valid.

Translator

I want a small set of rules representing the whole set of data, even if it's more difficult to interpret.

References

Galbrun, E. and Miettinen, P. (2012).
From Black and White to Full Color: Extending Redescription Mining Outside the Boolean World.
Statistical Analysis and Data Mining, 5(4):284–303.

van Leeuwen, M. and Galbrun, E. (2015).
Association Discovery in Two-View Data.
TKDE, 27(12):3190–3202.

Zaki, M. J. (2000).
Scalable algorithms for association mining.
TKDE, 12(3):372–390.

SPARQL Query

```
SELECT DISTINCT ?s ?p ?o WHERE {
    ?s ?p ?o .
    ?s dct:subject dbc:Smartphones .
    ?p a owl:ObjectProperty .
    FILTER (isURI(?o))
    FILTER (!STRSTARTS(STR(?o), "http://www.wikidata.org/"))
    FILTER (!STRSTARTS(STR(?o), "http://dbpedia.org/class/yago/"))
    FILTER (!STRSTARTS(STR(?p), "http://xmlns.com/foaf/0.1/"))
    FILTER (
        (?p != dbp:wordnet_type) AND (?p != dbp:website)
        AND (?p != prov:wasDerivedFrom) AND (?p != dbo:thumbnail)
        AND (?p != rdfs:comment) AND (?p != rdfs:label)
        AND (?p != rdfs:seeAlso) AND (?p != owl:sameAs)
        AND (?p != owl:differentFrom) AND (?p != foaf:depiction)
        AND (?p != dbo:wikiPageExternalLink)
    )
}
```

The query was run on DBpedia 2016-04.

Translator

- Searching for a set of rules that enable to construct one context from the other
- Greedy approach : adds the better rule at each step
- Quality measure based on *minimum description length* :

$$\Delta(X \to Y) = \underbrace{L(\mathit{Mask}^-) - L(\mathit{Mask}^+)}_{\text{Information gain}} - \underbrace{L(X \cup Y)}_{\text{Rule length}}$$

$$L(X) = -\sum_{x \in X} \log_2 P(x \mid \mathcal{K})$$

- Rules may be unidirectional or bidirectional

Statistiques sur les jeux de données extraits

	Triplets	Objets	Attributs	
			Cat.	Descr.
`Turing_Award`	2 642	65	503	857
`Smartphones`	8 418	598	359	1 730
`Sports_cars`	9 047	604	435	2 295
`French_films`	121 496	6 039	6 028	19 459

Results

Turing_Award_laureates

X	E	R	T
$\lvert\mathcal{R}^X\rvert$	47	12	11
$\lvert\mathcal{D}^X\rvert$	30	9	9
Précision	.64	.75	.85
$\overline{\lvert C_i\rvert}$ — $\overline{\lvert D_i\rvert}$	2	1	3 5
$\lvert C_i\rvert$ — $\lvert D_i\rvert$	4	1	5

Smartphones

X	E	R	T
$\lvert\mathcal{R}^X\rvert$	810	98	41
$\lvert\mathcal{D}^X\rvert$	521	57	31
Précision	.64	.58	.76
$\overline{\lvert C_i\rvert}$ — $\overline{\lvert D_i\rvert}$	4.3	1.6	3.1
$\lvert C_i\rvert$ — $\lvert D_i\rvert$	7.8	1.8	3.1

Sports_cars

X	E	R	T
$\lvert\mathcal{R}^X\rvert$	132	52	31
$\lvert\mathcal{D}^X\rvert$	95	30	23
Précision	.72	.68	.74
$\overline{\lvert C_i\rvert}$ — $\overline{\lvert D_i\rvert}$	2.8	1.3	2.6
$\lvert C_i\rvert$ — $\lvert D_i\rvert$	4.5	1.4	4.1

French_films

X	E	R	T
$\lvert\mathcal{R}^X\rvert$	546	36	93
$\lvert\mathcal{D}^X\rvert$	371	12	89
Précision	.68	.33	.96
$\overline{\lvert C_i\rvert}$ — $\overline{\lvert D_i\rvert}$	2.8	1.2	2.3
$\lvert C_i\rvert$ — $\lvert D_i\rvert$	4.4	1.1	4.2

Evaluation

Three expert evaluated each rule as True or False.

From the rules evaluated True, we build a rules base $\mathcal{D}$ of 20 rules.

We say that $X \leftrightarrow Y$ covers $A \leftrightarrow B$ iff $A \subseteq X$ and $Y \subseteq B$.

Given a set of k rules returned by the algorithm X, we can compute the precision and the recall of those rules wrt the rule base :

$$\text{recall}(X) = \frac{\left|\text{cov}(\mathcal{D}, \mathcal{R}_X)\right|}{|\mathcal{D}|}$$

$$\text{precision}(X) = \frac{\left|\{R \in \mathcal{R}_X \mid \exists\, D \in \mathcal{D}, R \text{ covers } D\}\right|}{\left|\mathcal{R}_X\right|}$$

where $\left|\text{cov}(\mathcal{D}, \mathcal{R}_X)\right|$ is the number of rules from D covered by a rule of R.