Revealing the Conceptual Schemas of RDF Datasets

Subhi Issa, Pierre-Henri Paris, Fayçal Hamdi, Samira Si-Said Cherfi

10 May, 2019

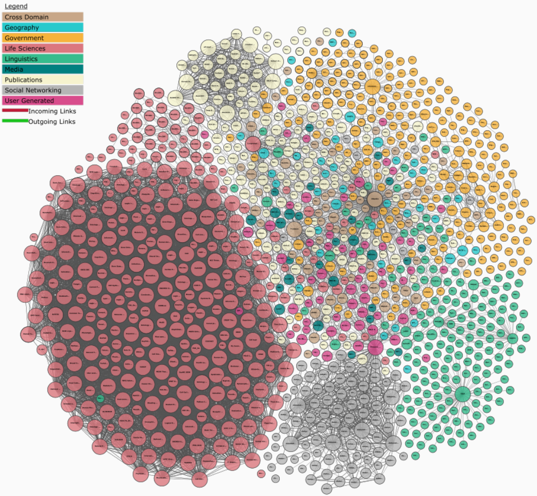
Conservatoire National des Arts et Métiers - CEDRIC

Table of contents

1. Introduction
2. Completeness
3. LOD-CM Prototype
4. Use cases
5. Conclusion & Future Works

Introduction

Linked Open Data is everywhere, but how good is it ?



The diagram is maintained by Andrejs Abele and John McCrae. http ://lod-cloud.net/

Introduction

What is the meaning of “Quality” ?

A popular definition for Quality is **fitness for use**. This means that data quality depends on the actual use case

Data Quality Dimension : a set of data quality attributes that represent a single aspect or construct of data quality

Linked Data Quality Dimensions

- Completeness
- Availability
- Performance
- Interlinking
- Licensing
- Versatility
- Timeliness
- Consistency
- Interoperability
- Understandability
- Trustworthiness
- Relevancy
- Interpretability
- Semantic accuracy
- Syntactic validity
- Conciseness
- Representation conciseness
- Security

Completeness

Linked Data Completeness

Completeness refers to the degree which all required information is presented in a particular dataset.

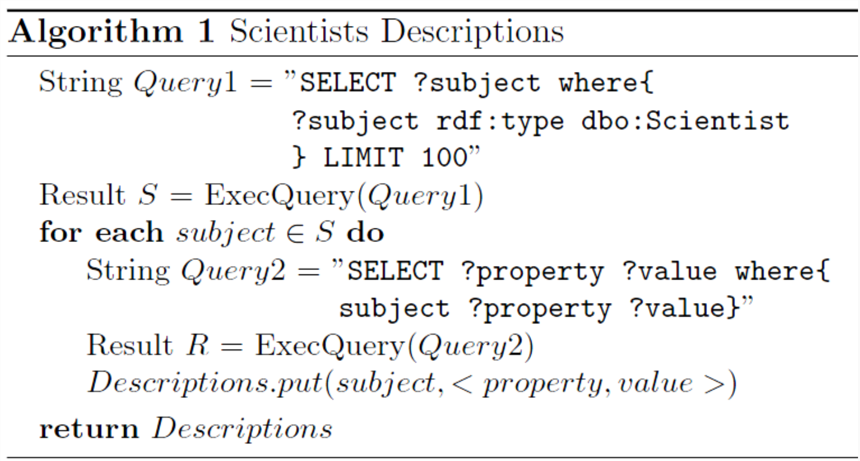
LD Completeness :

- Schema completeness, the degree where the classes and properties of an ontology are represented
- Property completeness, measure of the missing values for a specific property
- Population completeness, the percentage of all real-world objects of a particular type
- Interlinking completeness, the degree where instances in the dataset are interlinked

A reference schema (or gold standard) is required to assess completeness !

Motivating Example

- Giving the properties-values of 100 scientists



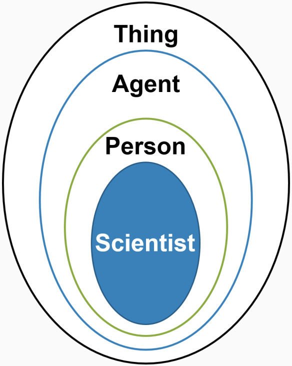
Algorithm 1 Scientists Descriptions

```
String $Query1$ = "SELECT ?subject where{
                  ?subject rdf:type dbo:Scientist
                  } LIMIT 100"
Result $S$ = ExecQuery($Query1$)
for each $subject \in S$ do
    String $Query2$ = "SELECT ?property ?value where{
                      subject ?property ?value}"
    Result $R$ = ExecQuery($Query2$)
    $Descriptions.put(subject, < property, value >)$
return $Descriptions$
```

Motivating Example

$$Scientist \sqsubseteq Person \sqsubseteq Agent \sqsubseteq Thing$$

$Scientist\_Schema = \{Properties\ on\ Scientist\} \cup$
$\{Properties\ on\ Person\} \cup \{Properties\ on\ Agent\} \cup$
$\{Properties\ on\ Thing\}$



Motivating Example

$$\begin{aligned} Comp(Albert\_Einstein) &= \frac{|Properties\ on\ Albert\_Einstein|}{|Scientist\_Schema|} \\ &= \frac{21}{664} = 3,61\% \end{aligned}$$

The property *weapon* is in *Scientist_Schema*, but it is not relevant to the *Albert_Einstein* instance

Linked Data Completeness : a Mining-based Approach

We postulate that :

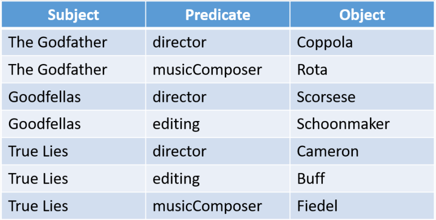
- Property frequently used by several instances of a given class is more important than less often used for the same instance

We propose to :

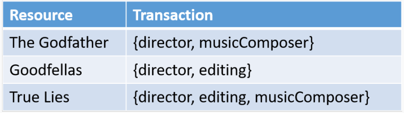
- Find properties used more frequently than others to describe instances of a given class

Linked Data Completeness : a Mining-based Approach

1st step: properties mining



Subject	Predicate	Object
The Godfather	director	Coppola
The Godfather	musicComposer	Rota
Goodfellas	director	Scorsese
Goodfellas	editing	Schoonmaker
True Lies	director	Cameron
True Lies	editing	Buff
True Lies	musicComposer	Fiedel



Resource	Transaction
The Godfather	{director, musicComposer}
Goodfellas	{director, editing}
True Lies	{director, editing, musicComposer}

Linked Data Completeness : a Mining-based Approach

2$^{\text{nd}}$ step: completeness calculation

$$MFP = \{\{director, musicCompoer\}, \{director, editing\}\}$$

Resource	Transaction
The Godfather	{director, musicComposer}
Goodfellas	{director, editing}
True Lies	{director, editing, musicComposer}

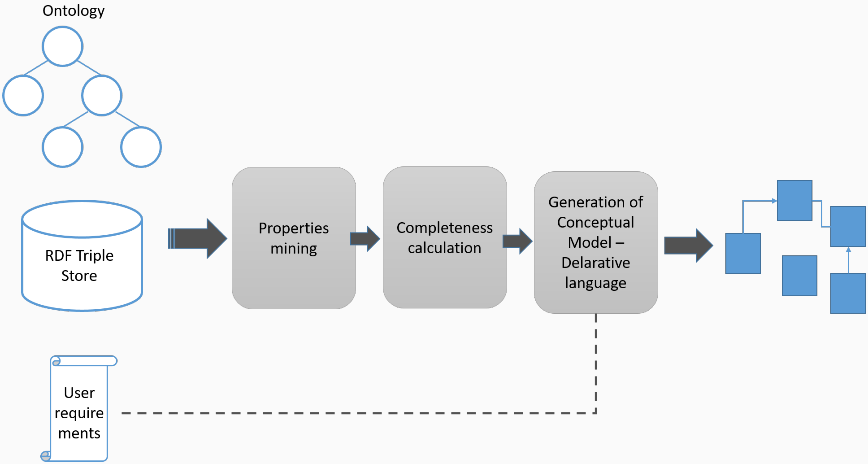
$$CP(I) = \frac{1}{|T|} \sum_{k=1}^{|T|} \sum_{j=1}^{|MFP|} \frac{\delta(P(t_k, p_j)}{|MPF|}$$

$$CP(I) = \frac{\left(\frac{1}{2}\right) + \left(\frac{1}{2}\right) + \left(\frac{2}{2}\right)}{3} = 0.67$$

Recover a Conceptual Schema from RDF Datasets

- Infer conceptual schemas from existing data - No predefined schema
- Conceptual Schema depends on :
  - Universe of discourse
  - User's requirements
- Enhance user's understanding of the representative system
- Provide a point of reference for system designers to extract schema specifications tagged with the completeness value

Recover a Conceptual Schema from RDF Datasets



Recover a Conceptual Data Model from RDF Datasets

Types of properties :

- Attribute : relate instances of class to literal data (e.g., string, number, etc.)
- Relationship : relate instances to other instances

Types of links :

- Inheritance link : describes the relation between the class and the superclass
- Association link : describes the relation between two classes and point to the property
- Dotted link : expresses that a class has been inferred to complete the relationship

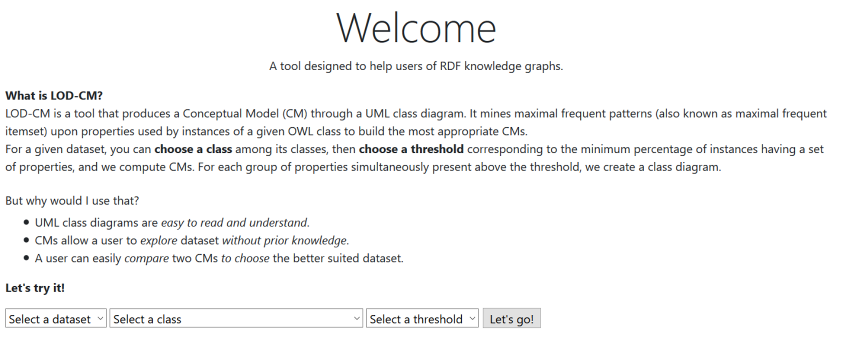
LOD-CM Prototype

Experimental setup

- DBpedia version 2016-10
  - English edition
  - 1.1 billion RDF triples
  - 468 classes
  - 1378 properties
- Data HDT dumps
- Implemented in C#
- PlantUML tool to create diagrams

LOD-CM

User interface [1]



Welcome

A tool designed to help users of RDF knowledge graphs.

What is LOD-CM?
LOD-CM is a tool that produces a Conceptual Model (CM) through a UML class diagram. It mines maximal frequent patterns (also known as maximal frequent itemset) upon properties used by instances of a given OWL class to build the most appropriate CMs.
For a given dataset, you can **choose a class** among its classes, then **choose a threshold** corresponding to the minimum percentage of instances having a set of properties, and we compute CMs. For each group of properties simultaneously present above the threshold, we create a class diagram.

But why would I use that?

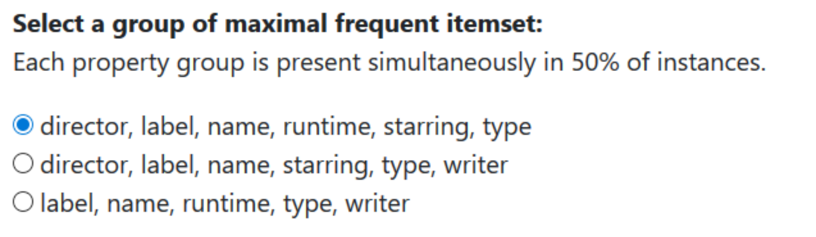
- UML class diagrams are *easy to read and understand*.
- CMs allow a user to *explore* dataset *without prior knowledge*.
- A user can easily *compare* two CMs *to choose* the better suited dataset.

Let's try it!

[1] `http://cedric.cnam.fr/lod-cm`

LOD-CM

Example : Class name : Film, Completeness : 50%



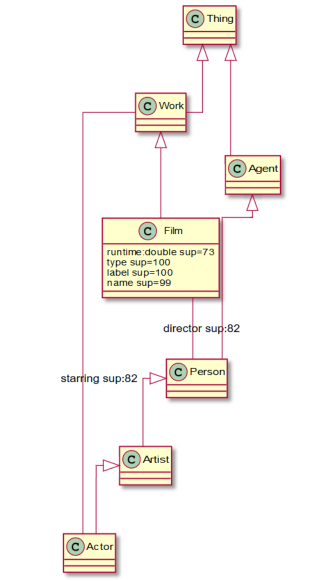
Select a group of maximal frequent itemset:

Each property group is present simultaneously in 50% of instances.

● director, label, name, runtime, starring, type
○ director, label, name, starring, type, writer
○ label, name, runtime, type, writer

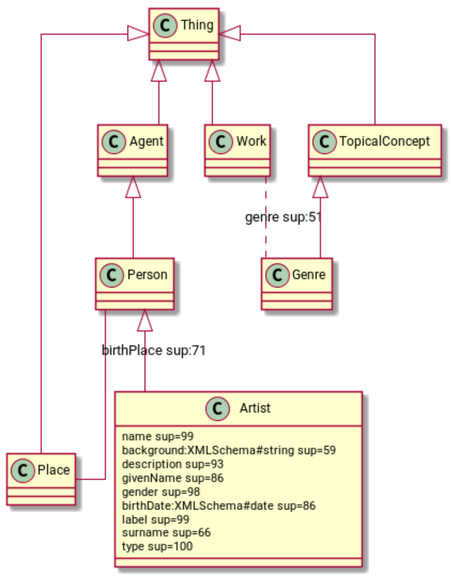
LOD-CM

Example : Class name : Film, Completeness : 50%



LOD-CM

Example : Class name : Artist, Completeness : 50%



Use cases

Use cases

- Browse dataset without examining data in detail
- Choose the dataset that will be most suitable for its intended use
- Facilitate data browsing

  Based on user requirements :

  - Inheritance relationship
  - Relations between classes
  - Completeness value of each property

Conclusion & Future Works

Conclusion & Future works

- Reveal conceptual schemas from RDF data sources
- Extract schema and present it as a model using user-specified threshold
- Model composes classes, relationships and properties enriched with completeness value

We plan to :

- Investigate the effectiveness of our prototype against additional Linked Open Data datasets such as Yago, Wikidata, etc.
- allow the user to compare conceptual schemas from different datasets

Linked Data Completeness : a Mining-based Approach

The proposed method

- Properties mining :

$$\mathcal{MFP} = \{\hat{P} \in \mathcal{FP} \mid \forall \hat{P}' \supsetneq \hat{P} : \frac{|T(\hat{P}')|}{|\mathcal{T}|} < \xi\}$$

where ξ is a user-specified threshold

- Completeness calculation :

$$\mathcal{CP}(\mathcal{I}') = \frac{1}{|\mathcal{T}|} \sum_{k=1}^{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{MFP}|} \frac{\delta(E(t_k), \hat{P}_j)}{|\mathcal{MFP}|}$$

such that : $\hat{P}_j \in \mathcal{MFP}$, and $\delta(E(t_k), \hat{P}_j) = \begin{cases} 1 & \text{if } \hat{P}_j \subset E(t_k) \\ 0 & \text{otherwise} \end{cases}$