Discovery of Link Keys in RDF Data Based on Pattern Structures: Preliminary Steps

Nacira Abbas, Jérôme David, Amedeo Napoli

Discovery of Link Keys in RDF Data Based on Pattern Structures: Preliminary Steps

Nacira Abbas[1], Jérôme David[2], and Amedeo Napoli[1]

[1] Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France
Nacira.Abbas@inria.fr, Amedeo.Napoli@loria.fr
[2] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France
Jerome.David@inria.fr

Abstract. In this paper, we are interested in the discovery of link keys among two different RDF datasets based on FCA and pattern structures. A link key identifies individuals which represent the same real world entity. Two main strategies are used to automatically discover link keys, ignoring or not the classes to which the individuals belong to. Indeed, a link key may be relevant for some pair of classes and not relevant for another. Then, discovering link keys for one pair of classes at a time may be computationally expensive if every pair should be considered. To overcome such limitations, we introduce a specific and original pattern structure where link keys can be discovered in one pass while specifying the pair of classes associated with each link key, focusing on the discovery process and allowing more flexibility.

Keywords: Link key· Link Key Discovery· Pattern Structures· Linked Data· RDF.

1 Introduction

RDF (Resource Description Framework) triples $\langle$`subject,property,object`$\rangle$ are the basic bricks of the web of data. The elements in the triples are described using terms of RDF Schema and OWL ontologies. The same real world entity can be represented in different datasets by different individuals, i.e. subjects. Data interlinking is the task of finding identity links across datasets. Two main approaches are proposed to perform this task. The first one measures a similarity between subjects considering that the closest the subjects, the more likely they are the same [13, 11]. The second one is based on rules, which express sufficient conditions for two subjects to be the same [12, 2, 1]. One method using the latter approach is based on *link keys* [3] that extend the notion of a key used in databases. Link keys are rules allowing to infer identity links between RDF datasets. A link key takes the form of two sets of pairs of properties associated with a pair of classes. The pairs of properties express sufficient conditions for two subjects, from the associated pair of classes, to be the same. An example of a link key is:

$$\{\langle \texttt{designation}, \texttt{title}\rangle\}, \{\langle \texttt{creator}, \texttt{author}\rangle\}, \langle \texttt{Book}, \texttt{Novel}\rangle$$

stating that whenever an instance a_1 of the class `Book` has the same values for the property `designation` as an instance b_1 of the class `Novel` for the property `title`,

and that a_1 and b_1 share at least one value for the properties `creator` and `author`, then a_1 and b_1 denote the same entity.

Usually link keys are not provided. For this reason, a first algorithm was proposed in [3] for automatically discovering link keys from datasets. This algorithm starts from two RDF datasets, discovers link key candidates then, evaluates these candidates according to quality measures. The quality of a link key candidate is evaluated according to two measures [3]. The first one, called *coverage*, relies on the intuition that the more subjects linked by a link key candidate, the more complete this link key is. The second measure, called *discriminability*, assumes that subjects, in each dataset, have to be distinct. It measures the capability of a link key candidate to discriminate between subjects. In order to take into account these two measures, an harmonic mean might be used.

The question of using Formal Concept Analysis (FCA) to discover link keys has arisen naturally, since a link key candidate presents some mathematical properties which are similar to the ones of a formal concept in FCA [4, 5].

To discover link keys candidates, the existing methods apply one of the following strategies. The first one takes as input all the subjects from two datasets, ignoring the classes to which the subjects belong to, e.g. `Book`. This strategy generates link key candidates that apply to the whole datasets, i.e. it does not specify the pair of classes associated with each link key candidate. These candidates are then evaluated considering the whole datasets and again without taking into account the pair of classes. This evaluation, however, is not accurate since a link key candidate may be relevant for a pair of classes e.g. $\langle$`Book`,`Dictionary`$\rangle$ and not relevant for another pair e.g. $\langle$`Book`,`Novel`$\rangle$. The second strategy consists in finding link key candidates for one particular pair of classes at a time, then evaluating these candidates w.r.t. this particular pair. This strategy repeats the same process for all the pairs of classes issued from the two datasets. This allows a more accurate evaluation of the link key candidates. However, we do not know in advance which classes to take as input at a time. Consequently, a naive approach would be to consider all the pairs of classes from the Cartesian product of the sets of classes of the given datasets, or to require a class alignment [7]. The first solution is computationally expensive and the second one is not always possible because we do not have systematically a class alignment.

In this paper, we propose a method based on Pattern Structures, a generalization of Formal Concept Analysis [8], that overcomes these limits. This method allows to find link key candidates in one pass, i.e. without iterating on every pair of classes, while specifying the pairs of classes associated with each link key candidate without requiring an a priori alignment. Moreover, datasets may classify the same entities differently, for example, in one dataset `"Marie Curie"` is an instance of the class `"Woman"` and at the same time an instance of the class `"Scientist"`, while in an another dataset, `"Marie Curie"` is an instance of a unique class `"FemaleScientist"`. In this work, we propose to take into account this difference in abstraction, by generalizing the notion of a link key associated with a pair of classes to a link key associated with a pair of class expressions. For example, such a link key candidate could be associated with the pair $\langle$`Woman and Scientist`,`FemaleScientist`$\rangle$ where `"Woman and Scientist"` is a class expression.

The plan of the paper is as follows. First we give some definitions and notations. Then, we present how the problem of link key discovery is encoded in FCA. After that, we formalize the problem with pattern structure and we show how to discover link keys from two datasets in one pass while specifing which pairs of class expressions are associated with these candidates.

2 Preliminaries

An RDF dataset [10] is a set of triples $\langle \texttt{s,p,o} \rangle \in (U \cup B) \times U \times (U \cup B \cup L)$, where U is a set of IRIs (Internationalized Resource Identifier), B a set of blank nodes i.e. variables or "anonymous resources" and L a set of literals, i.e. values depending on datatypes. To avoid any confusion with FCA objects, we refer to an "object" in an RDF triple as "RDF object". Figure 1 represents two RDF datasets where an example of an RDF triple is $\langle b_{13}, \texttt{title,Les Misérables} \rangle$, expressing that the subject b_{13} is related through the property `title` to the RDF object `Les Misérables`. For short, we will write that b_{13} has the value `Les Misérables` for the property `title`. The properties in RDF are not functional, i.e. for one property, a subject may be related to more than one RDF object or no RDF object at all. The set of values of a subject s for the property p is given by $p(s)$ such as $p(s) = \{o | \langle s,p,o \rangle \in D\}$ e.g. $\texttt{author}(b_{15}) = \{\texttt{Stephen King, Peter Straub}\}$. The property `rdf:type` is used in RDF to expess that a subject belongs to a particular class i.e. that a subject is an instance of a class. For example, the triple $\langle b_{13}, \texttt{rdf:type,Novel} \rangle$ means that b_{13} is an instance of the class `Novel`. Given a dataset D, the sets $S(D), P(D), Cl(D)$ denote respectively the set of subjects, the set of properties and the set of classes in D. The set of instances of a class C is $S(C) = \{s \mid \langle s, \texttt{rdf:type}, C \rangle \in D\}$.

2.1 Link keys

Given two datasets D_1 and D_2, we aim to discover identity links between these datasets. An identity link is a statement of the form $\langle s_1, \texttt{owl:sameAs}, s_2 \rangle$ expressing that the subject s_1 from D_1 and the subject s_2 from D_2 represent the same real world entity. For example, given D_1 and D_2 as represented in Figure 1, the data interlinking task should discover the identity link $\langle a_{13}, \texttt{owl:sameAs}, b_{13} \rangle$ because the subjects a_{13} and b_{13} represent both the same book `Les Misérables` written by `Victor Hugo`. For short, we write $\langle a_{13}, b_{13} \rangle$ and we call this pair a *link*. A link key is used to generate such links.

We distinguish two notions related to link keys. The first one is *link key expression* which is the syntactic form of a link key, i.e. it does not necessarily generate actual links. The second one is *link key candidate* which is a link key expression that generates at least a link and it is maximal w.r.t. its generated link set. Actually, link key discovery methods find link key candidates and evaluate them thanks to adapted measures such as proposed in [3]. Finally, the most relevant candidates will be selected as valid link keys and used to infer identity links among the datasets.

The notion of a link key has been defined in [5]. In this paper, we generalize the notion of a link key candidate associated with a pair of classes to a link key candidate associated with a pair of *class expressions*. We restrict ourselves to a subset of class

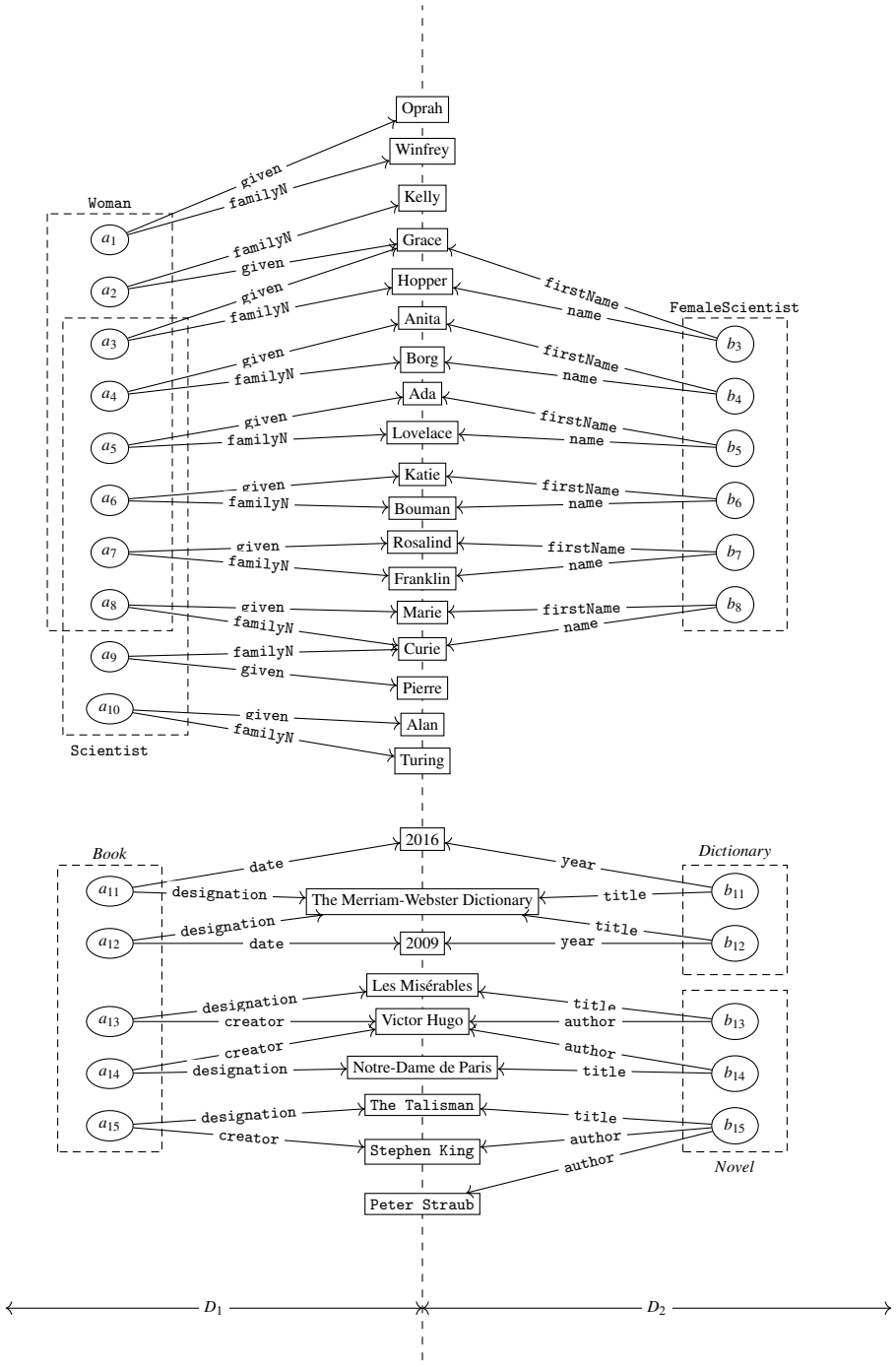

Fig. 1. Example of two RDF datasets. On the left-hand side, the dataset D_1 populated with instances of the classes: `Woman`, `Scientist` and `Book`. On the right-hand side, the dataset D_2 populated with instances of the classes: `FemaleScientist`, `Dictionary` and `Novel`

expressions from description logics [6], composed of concept names related by the Boolean operators $\sqcap_{DL}$ and $\sqcup_{DL}$ (respectively conjunction and disjunction in description logics). The intuition behind this generalization is that the ontologies on which the datasets rely may use different levels of abstraction to describe entities. For example the set of female scientists may be described by the intersection of the classes `Woman` and `Scientist` in one dataset and by the named class `FemaleScientist` in another dataset. In this case, it would be more accurate to define a link key that discovers links between the class intersection `Woman` and `Scientist` and the class `FemaleScientist`. The restriction to $\sqcap_{DL}$ and $\sqcup_{DL}$ operators, comes from the fact that we are dealing only with asserted `rdf:type` in the RDF dataset. Following the semantics of operators in description logics, the set of subjects belonging to $C_1 \sqcap_{DL} C_2$ is $S(C_1 \sqcap_{DL} C_2) = S(C_1) \cap S(C_2)$ and the set of subjects belonging to $C_1 \sqcup_{DL} C_2$ is $S(C_1 \sqcup_{DL} C_2) = S(C_1) \cup S(C_2)$.

Firstly below we introduce the definition of a link key expression associated with a pair of class expressions.

Definition 1 (Link key expression associated with a pair of class expressions). *Let us consider two datasets D_1 and D_2. Let Eq and $In \neq \emptyset$ be subsets of pairs of properties, such as, $Eq \subseteq P(D_1) \times P(D_2)$, $In \subseteq P(D_1) \times P(D_2)$, $Eq \subseteq In$. Let CE_1 be a class expression over $Cl(D_1)$ and CE_2 a class expression over $Cl(D_2)$. $k = (Eq, In, \langle CE_1, CE_2 \rangle)$ is a link key expression associated with the pair of class expressions $\langle CE_1, CE_2 \rangle$ over D_1 and D_2.*

Actually we replace the pair of named classes in a link key expression as defined in [5] with a pair of class expressions. As example of link key expression is
$k = (\{\langle$`given,year`$\rangle\}, \{\langle$`given,year`$\rangle\}, \langle$`Woman`$\sqcap_{DL}$`Scientist,FemaleScientist`$\rangle)$.

A link key expression associated with a pair of class expressions may generate links among these class expressions. We define this link set as follows.

Definition 2 (Link set generated by a link key expression associated with a pair of class expressions). *Given two datasets D_1 and D_2. Let $k = (Eq, In, \langle CE_1, CE_2 \rangle)$ be a link key expression associated with the pair of class expressions $\langle CE_1, CE_2 \rangle$ over D_1 and D_2. The* link set generated by *k is the subset $L_k \subseteq (S(CE_1) \times S(CE_2))$ defined as $L_k = \{\langle s_1, s_2 \rangle \in S(CE_1) \times S(CE_2)) \mid p_1(s_1) = p_2(s_2) \neq \emptyset$ for all $\langle p_1, p_2 \rangle \in Eq$ and $p_1(s_1) \cap p_2(s_2) \neq \emptyset$ for all $\langle p_1, p_2 \rangle \in In\}$.*

As the properties in RDF are not functional, we compare the values of subjects in two ways (i) Eq are pairs of properties for which two subjects share all their values and (ii) In are those pairs of properties for which two subjects share at least one value. For example $\langle a_{15}, b_{15} \rangle \in L_k$ where:
$k = (\{\langle$`designation,title`$\rangle\}, \{\langle$`designation,title`$\rangle, \langle$`creator,author`$\rangle\}, \langle$`Book,Novel`$\rangle)$
because $\langle a_{15}, b_{15} \rangle \in S($`Book`$) \times S($`Novel`$)$ and `designation`$(a_{15}) =$ `title`$(b_{15}) \neq \emptyset$ and `creator`$(a_{15}) \cap$ `author`$(b_{15}) \neq \emptyset$.

A link key candidate is a link key expression that generates at least a link and it is maximal on the link set that it generates. To define "maximality" we have to define an order between link key expressions.

Definition 3 (Meet, join of link key expressions associated with a pair of class expressions). *Given two datasets D_1 and D_2. Let $k_1 = (Eq_1, In_1, \langle CE_1^1, CE_2^1 \rangle)$ and*

$k_2 = (Eq_2, In_2, \langle CE_1^2, CE_2^2 \rangle)$ be link key expressions over D_1 and D_2. The meet $\sqcap$ and the join $\sqcup$ of k_1 and k_2 are defined as follows:

$$k_1 \sqcap k_2 = (Eq_1 \cap Eq_2, In_1 \cap In_2, \langle (CE_1^1 \sqcup_{DL} CE_1^2), (CE_2^1 \sqcup_{DL} CE_2^2) \rangle)$$
$$k_1 \sqcup k_2 = (Eq_1 \cup Eq_2, In_1 \cup In_2, \langle (CE_1^1 \sqcap_{DL} CE_1^2), (CE_2^1 \sqcap_{DL} CE_2^2) \rangle)$$

The link set of a link key expression $k_1 \sqcap k_2$ is equal to the union of the link sets of k_1 and k_2. The less the number of pairs of properties to compare in a link key expression, the more the pairs of subjects satisfying these pairs of properties. Thus the larger the classes in a link key expression, and dually for $k_1 \sqcup k_2$.
As an example from the datasets represented in Figure 1, the meet of two link key expressions $k_1 = (\{\langle \texttt{given,firstName} \rangle\}, \{\langle \texttt{given,firstName} \rangle\}, \langle \texttt{Woman,FemaleScientist} \rangle)$ and $k_2 = (\{\langle \texttt{given,firstName} \rangle\}, \{\langle \texttt{given,firstName} \rangle, \langle \texttt{name,familyN} \rangle\}, \langle \texttt{(Woman}\sqcap_{DL}\texttt{Scientist),FemaleScientist} \rangle)$ is
$k_1 \sqcap k_2 = (\{\langle \texttt{given,firstName} \rangle\}, \{\langle \texttt{given,firstName} \rangle\}, \langle \texttt{Woman,FemaleScientist} \rangle)$.
The join $k_1 \sqcup k_2 = (\{\langle \texttt{given,firstName} \rangle\}, \{\langle \texttt{given,firstName} \rangle, \langle \texttt{name,familyN} \rangle\}, \langle \texttt{(Woman}\sqcap_{DL}\texttt{Scientist),FemaleScientist} \rangle)$.

Now we formally define a link key candidate associated with a pair of class expressions.

Definition 4 (Link key candidate associated with a pair of class expressions). *Let us consider two datasets D_1 and D_2. Let $k = (Eq, In, \langle CE_1, CE_2 \rangle)$ be a link key expression associated with the pair of class expressions $\langle CE_1, CE_2 \rangle$ over D_1 and D_2. k is a* link key candidate *for D_1 and D_2 if*

- *$L_k \neq \emptyset$, and*
- *$k = \bigsqcup_{h \in [k]} h$ such that $[k] = \{h \mid L_k = L_h\}$*

Intuitively the link sets generated by link key expressions form a partition of the set of link key expressions. Link key candidates are the maximal elements of the classes of this partition. This definition matches the definition of a closed set. This explains the use of Formal Concept Analysis [9] for link key discovery since the intent and the extent of a formal concept are closed sets.

2.2 Link key discovery with Formal Concept Analysis

The link key discovery based on Formal Concept Analysis is detailed in [5]. Given two datasets D_1 and D_2 and a pair of classes $\langle C_1, C_2 \rangle \in Cl(D_1) \times Cl(D_2)$. The *$LK-$formal context* or the *formal context for link key candidates* associated with a pair of classes $\langle C_1, C_2 \rangle$ is the triple $\langle (S(C_1) \times S(C_2)), \{\exists, \forall\} \times P(D_1) \times P(D_2), I \rangle$ such that:

- The set of objects of the $LK-$formal context is the set of pairs of subjects $\langle s_1, s_2 \rangle \in (S(C_1) \times S(C_2))$.
- The set of attributes of the $LK-$formal context is the set of pairs of properties $\langle p_1, p_2 \rangle \in P(D_1) \times P(D_2)$ preceded by a quantifier in $\{\exists, \forall\}$ i.e. $\forall \langle p_1, p_2 \rangle$ and $\exists \langle p_1, p_2 \rangle$.
- The relation I between an object and an attribut is defined as follows:

$$\langle s_1, s_2 \rangle \; I \; \forall \langle p_1, p_2 \rangle \;\; \text{iff } p_1(s_1) = p_2(s_2) \neq \emptyset$$
$$\langle s_1, s_2 \rangle \; I \; \exists \langle p_1, p_2 \rangle \;\; \text{iff } p_1(s_1) \cap p_2(s_2) \neq \emptyset$$

Link key discovery based on FCA takes as input one pair of classes at a time. To discover link key candidates, in one pass, we take as input the pair of classes ⟨owl:Thing, owl:Thing⟩, where owl:Thing is a class containing all the subjects. The generated link key candidates will be associated with the pair ⟨owl:Thing,owl:Thing⟩ which means that they apply to the whole datasets. Figure 2 presents the concept lattice related to the $LK-$formal context associated with the pair of classes ⟨owl:Thing, owl:Thing⟩ for the datasets D_1 and D_2 introduced in Figure 1.

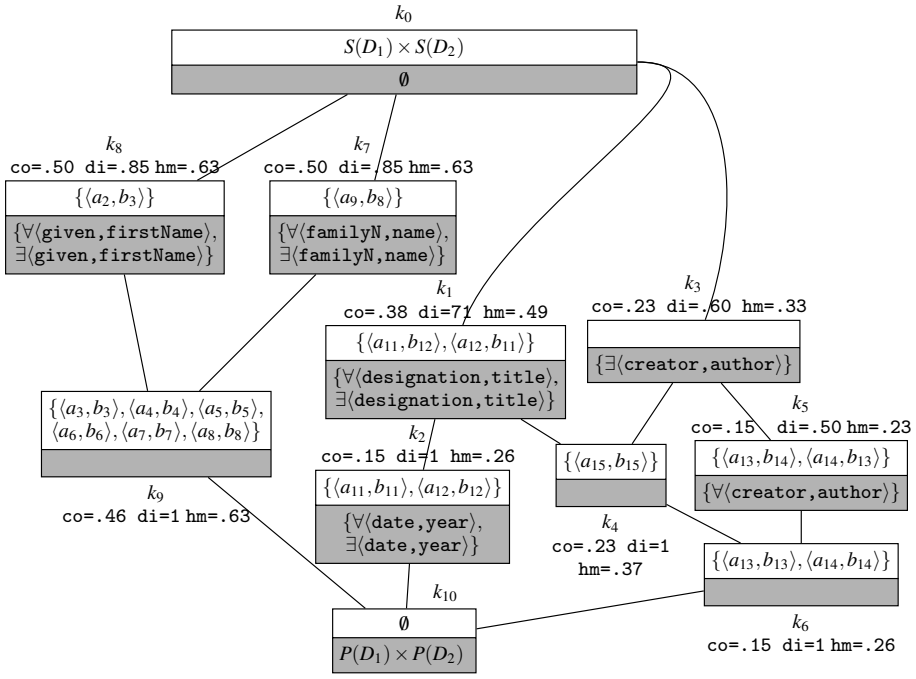

Fig. 2. The lattice of the $LK-$formal context associated with the pair of classes ⟨owl:Thing, owl:Thing⟩

It was shown in [5] that the intents of the formal concepts of the lattice generated from the $LK-$formal context $\langle C_1 \times C_2, \{\exists, \forall\} \times P(D_1) \times P(D_2), I\rangle$ are the link key candidates for the pair of classes $\langle C_1, C_2\rangle$. In this case, if (A,B) is a formal concept, the link key candidate in (A,B) is $k_B = (Eq, In, \langle C_1, C_2\rangle)$ where $Eq = \{\forall\langle p_1, p_2\rangle \in B\}$, $In = \{\exists\langle p_1, p_2\rangle \in B\}$ and the link set generated by k_B is the extent of this formal concept $L_{k_B} = A$. For example, k_8 corresponds to the link key candidate ($\{\langle$given,firstName$\rangle\}$, $\{\langle$given,firstName$\rangle\}$,$\langle$owl:Thing,owl:Thing$\rangle$).

Now, to evaluate the quality of a link key candidate in terms of coverage and discriminability, let us consider $L \subseteq (S(CE_1) \times S(CE_2))$ and $\pi_1(L) = \{s_1 \in S(CE_1) | \langle s_1, s_2\rangle \in L\}$, $\pi_2(L) = \{s_2 \in S(CE_2) | \langle s_1, s_2\rangle \in L\}$, where $\pi_1(L)$ is the set of instances of the class

expession CE_1 appearing in L and $\pi_2(L)$ is the set of instances of the class expession CE_2 appearing in L.

The coverage of a link key candidate k associated with a pair of class expressions $\langle CE_1, CE_2 \rangle$ is denoted by $co(k)$ and defined as:

$$co(k) = \frac{|\pi_1(L_k) \cup \pi_2(L_k)|}{|S(CE_1) \cup S(CE_2)|}$$

The coverage is the proportion between the number of instances identified (linked) by k and the overall number of instances of the two class expressions. This means that the coverage measures how general a link key candidate is. When $co(k)$ is equal to 1, this means that all instances from CE_1 and CE_2 are identified by the link key candidate k.
The discriminability of a link key candidate k associated with a pair of class expressions $\langle CE_1, CE_2 \rangle$ is denoted by $di(k)$ and defined as:

$$di(k) = \frac{min(|\pi_1(L_k), \pi_2(L_k)|)}{|L_k|}$$

$di(k)$ is the proportion of the minimum number of instances identified by k and the number of links generated by k. When $di(k)$ is equal to 1, then the link key is perfectly discriminant, i.e. it generates one-to-one mappings.

Coverage and discriminability are aggregated by the harmonic mean denoted $hm(k)$ and defined as:

$$hm(k) = \frac{2}{\frac{1}{co(k)} + \frac{1}{di(k)}}$$

The coverage, discriminability, and harmonic mean of the link key cadidate k_9 in the lattice in Figure 2, are respectively, `co=0.46`, `di=1`, `hm=0.63`. The link key discovery methods return as link keys the link key candidates whose harmonic mean value is the highest. In this example, they return the link key candidates k_7, k_8 and k_9, because they have the highest *hm*. However, even if these candidates are relevant for the pair $\langle$`(Woman`$\sqcap_{DL}$`Scientist),FemaleScientist`$\rangle$, they do not generate any links for the pair $\langle$`Book,Dictionary`$\rangle$. In fact for this latter pair, the link key candidate k_2 is more relevant, because it generates only and all correct links, even if it has a low `hm`. For the pair $\langle$`Book,Novel`$\rangle$, the candidate k_4 is more relevant. Even if they are relevant, these link key candidates show a low `hm` because they are evaluated considering the whole datasets i.e. $\langle$`owl:Thing, owl:Thing`$\rangle$.

We propose in the following our main contribution which is a method based on pattern structures for discovering relevant link keys for given pairs of classes.

3 Link key discovery within Pattern Structures

In the following, we propose a method based on pattern structures, a generalization of FCA [8], that, given two datasets, discovers link key candidates in one pass (without iterating on every pair of classes) while specifying the classes associated with each link key candidate.

3.1 A Pattern Structure for Link Key Discovery

We define the pattern structure for link key candidates discovery where the set of objects is the set of pairs of subjects issued from two datasets. Actually these pairs correspond to potential links. In the pattern structure, the description of a potential link is given by the maximal link key expression that generates this link. Then the meet of two descriptions corresponds to the meet of link key expressions as introduced in Definition 3.

Definition 5 (Pattern structure for link key candidate discovery). *Given two datasets D_1 and D_2. The pattern structure for link key candidate discovery between D_1 and D_2, called hereafter the LK−*pattern structure*, is the triple $(S(D_1) \times S(D_2), (E, \sqcap), \delta)$ where:*

- *The set of objects $S(D_1) \times S(D_2)$ is the set of pairs of subjects over D_1 and D_2.*
- *E is the set of potential object descriptions. A description is a link key expression $k = (Eq, In, \langle CE_1, CE_2 \rangle)$ over D_1 and D_2.*
- *$(E, \sqcap)$ is a meet semilattice where the meet $\sqcap$ of two descriptions $k_1 = (Eq_1, In_1, \langle CE_1^1, CE_2^1 \rangle)$ and $k_2 = (Eq_2, In_2, \langle CE_1^2, CE_2^2 \rangle)$ is given in Definition 3: $k_1 \sqcap k_2 = (Eq_1 \cap Eq_2, In_1 \cap In_2, \langle (CE_1^1 \sqcup_{DL} CE_1^2), (CE_2^1 \sqcup_{DL} CE_2^2) \rangle)$. The descriptions are partially ordered by $\sqsubseteq$ defined w.r.t. the similarity operator $\sqcap$. If $k_1 \sqcap k_2 = k_1 \Leftrightarrow k_1 \sqsubseteq k_2$.*
- *The mapping $\delta : S(D_1) \times S(D_2) \rightarrow E$ associates each pair of subjects $\langle s_1, s_2 \rangle \in S(D_1) \times S(D_2)$ to its description $\delta(\langle s_1, s_2 \rangle) = (Eq, In, \langle CE_1, CE_2 \rangle)$ where, $Eq = \{\langle p_1, p_2 \rangle | p_1(s_1) = p_2(s_2) \neq \emptyset\}$, $In = \{\langle p_1, p_2 \rangle | p_1(s_1) \cap p_2(s_2) \neq \emptyset\}$, CE_1 (resp. CE_2) is the conjunction of the classes of s_1 (resp. s_2) over $Cl(D_1)$ (resp. $Cl(D_2)$).*

The $LK-$pattern structure for the datasets in Figure 1 is given in Table 1. The set of objects is the set of pairs of subjects from $S(D_1) \times S(D_2)$. The set of potential object descriptions E is the set link key expressions over D_1 and D_2. For example, such a description is given by $k = (\{\langle \texttt{designation,title} \rangle\}, \{\langle \texttt{designation,title} \rangle\}, \langle \texttt{Book,Dictionary} \rangle)$. We may calculate the description of the pair $\langle a_8, b_8 \rangle \in S(D_1) \times S(D_2)$ as follows:

$$\begin{aligned} \delta(\langle a_8, b_8 \rangle) = (&\{\langle \texttt{given,firstName} \rangle, \langle \texttt{familyN,name} \rangle\}, \\ &\{\langle \texttt{given,firstName} \rangle, \langle \texttt{familyN,name} \rangle\}, \\ &\langle \texttt{(Woman} \sqcap_{DL} \texttt{Scientist),FemaleScientist} \rangle) \end{aligned}$$

The meet of the two descriptions k_1 and k_2 can be calculated as follows:

$$\begin{aligned} k_1 &= (\{\langle \texttt{given,firstName} \rangle, \langle \texttt{familyN,name} \rangle\}, \{\langle \texttt{given,firstName} \rangle, \langle \texttt{familyN,name} \rangle\}, \\ &\quad \langle \texttt{(Woman} \sqcap_{DL} \texttt{Scientist),FemaleScientist} \rangle) \\ k_2 &= (\{\langle \texttt{given,firstName} \rangle\}, \{\langle \texttt{given,firstName} \rangle\}, \langle \texttt{Woman,FemaleScientist} \rangle) \\ k_1 \sqcap k_2 &= (\{\langle \texttt{given,firstName} \rangle\}, \{\langle \texttt{given,firstName} \rangle\}, \langle \texttt{Woman,FemaleScientist} \rangle) \\ k_1 \sqcap k_2 &= k_2, \text{ hence, } k_2 \sqsubseteq k_1 \end{aligned}$$

The derivation operators $.^{\square}$ form a Galois connection between $2^{S(D_1) \times S(D_2)}$ and E and defined as follows:

$$L^{\square} = \prod_{\langle s_1, s_2 \rangle \in L} \delta(\langle s_1, s_2 \rangle) \qquad L \subseteq S(D_1) \times S(D_2)$$

$$k^{\square} = \{\langle s_1, s_2 \rangle \in S(D_1) \times S(D_2) | k \sqsubseteq \delta(\langle s_1, s_2 \rangle)\} \qquad k \in E$$

Objects	Descriptions		
$S(D_1) \times S(D_2)$	Eq	In	$\langle CE_1, CE_2 \rangle$
$\langle a_2, b_3 \rangle$	{⟨given,firstName⟩}	{⟨given,firstName⟩}	⟨Woman, FemaleScientist⟩
$\langle a_3, b_3 \rangle$	{⟨given,firstName⟩, ⟨familyN,name⟩}	{⟨given,firstName⟩, ⟨familyN,name⟩}	⟨(Woman⊓$_{DL}$Scientist), FemaleScientist⟩
$\langle a_4, b_4 \rangle$	{⟨given,firstName⟩, ⟨familyN,name⟩}	{⟨given,firstName⟩, ⟨familyN,name⟩}	⟨(Woman⊓$_{DL}$Scientist), FemaleScientist⟩
$\langle a_5, b_5 \rangle$	{⟨given,firstName⟩, ⟨familyN,name⟩}	{⟨given,firstName⟩, ⟨familyN,name⟩}	⟨(Woman⊓$_{DL}$Scientist), FemaleScientist⟩
$\langle a_6, b_6 \rangle$	{⟨given,firstName⟩, ⟨familyN,name⟩}	{⟨given,firstName⟩, ⟨familyN,name⟩}	⟨(Woman⊓$_{DL}$Scientist), FemaleScientist⟩
$\langle a_7, b_7 \rangle$	{⟨given,firstName⟩, ⟨familyN,name⟩}	{⟨given,firstName⟩, ⟨familyN,name⟩}	⟨(Woman⊓$_{DL}$Scientist), FemaleScientist⟩
$\langle a_8, b_8 \rangle$	{⟨given,firstName⟩, ⟨familyN,name⟩}	{⟨given,firstName⟩, ⟨familyN,name⟩}	⟨(Woman⊓$_{DL}$Scientist), FemaleScientist⟩
$\langle a_9, b_8 \rangle$	{⟨familyN,name⟩}	{⟨familyN,name⟩}	⟨Scientist, FemaleScientist⟩
$\langle a_{11}, b_{11} \rangle$	{⟨date,year⟩, ⟨designation,title⟩}	{⟨date,year⟩, ⟨designation,title⟩}	⟨Book,Dictionary⟩
$\langle a_{11}, b_{12} \rangle$	{⟨designation,title⟩}	{⟨designation,title⟩}	⟨Book,Dictionary⟩
$\langle a_{12}, b_{11} \rangle$	{⟨designation,title⟩}	{⟨designation,title⟩}	⟨Book,Dictionary⟩
$\langle a_{12}, b_{12} \rangle$	{⟨date,year⟩, ⟨designation,title⟩}	{⟨date,year⟩, ⟨designation,title⟩}	⟨Book,Dictionary⟩
$\langle a_{13}, b_{13} \rangle$	{⟨creator,author⟩, ⟨designation,title⟩}	{⟨creator,author⟩, ⟨designation,title⟩}	⟨Book,Novel⟩
$\langle a_{13}, b_{14} \rangle$	{⟨creator,author⟩}	{⟨creator,author⟩}	⟨Book,Novel⟩
$\langle a_{14}, b_{13} \rangle$	{⟨creator,author⟩}	{⟨creator,author⟩}	⟨Book,Novel⟩
$\langle a_{14}, b_{14} \rangle$	{⟨creator,author⟩, ⟨designation,title⟩}	{⟨creator,author⟩, ⟨designation,title⟩}	⟨Book,Novel⟩
$\langle a_{15}, b_{15} \rangle$	{⟨designation,title⟩}	{⟨creator,author⟩, ⟨designation,title⟩}	⟨Book,Novel⟩

Table 1. $LK-$pattern structure for the datasets given in Figure 1

As usual, L is a closed set if $L^{\square\square} = L$ and k is a closed set if $k^{\square\square} = k$. Then a pattern concept verifies: $L^{\square} = k$ and $k^{\square} = L$. The link key expression k is a link key candidate for the two datasets D_1 and D_2 if and only if (L,k) is a pattern concept of the $LK-$pattern structure for D_1 and D_2.

3.2 Discussion

Figure 3 represents the pattern concept lattice generated from the $LK-$pattern structure in Table 1. Each intent of a pattern concept represents a link key candidate associated with a pair of class expressions. For example the link key candidate k_9 is associated with the pair of class expressions ⟨(Woman⊓$_{DL}$Scientist),FemaleScientist⟩. It should be noticed that the link key candidate k_9 in Figure 3 corresponds to the link key candidate k_9 in Figure 2 (calculated with plain FCA). In the pattern concept lattice we can see that the link key k_9 is associated with the pair of class expressions ⟨(Woman⊓$_{DL}$Scientist),FemaleScientist⟩, whereas this was not possible using plain FCA (see [5]). Moreover, different link key candidates may have the same sets of pairs of properties (Eq and In) but are associated with different pairs of class expressions. For example, k_{1a} and k_{1b}, which correspond to the link key candidate k_1 in Figure 2, have the same sets of pairs of properties {⟨designation,title⟩},{⟨designation,title⟩}, but they are associated with different pairs of class expressions: k_{1a} is associated with ⟨Book,(Dictionary⊔$_{DL}$Novel)⟩ and k_{1b} is associated with ⟨Book,Novel⟩. The properties

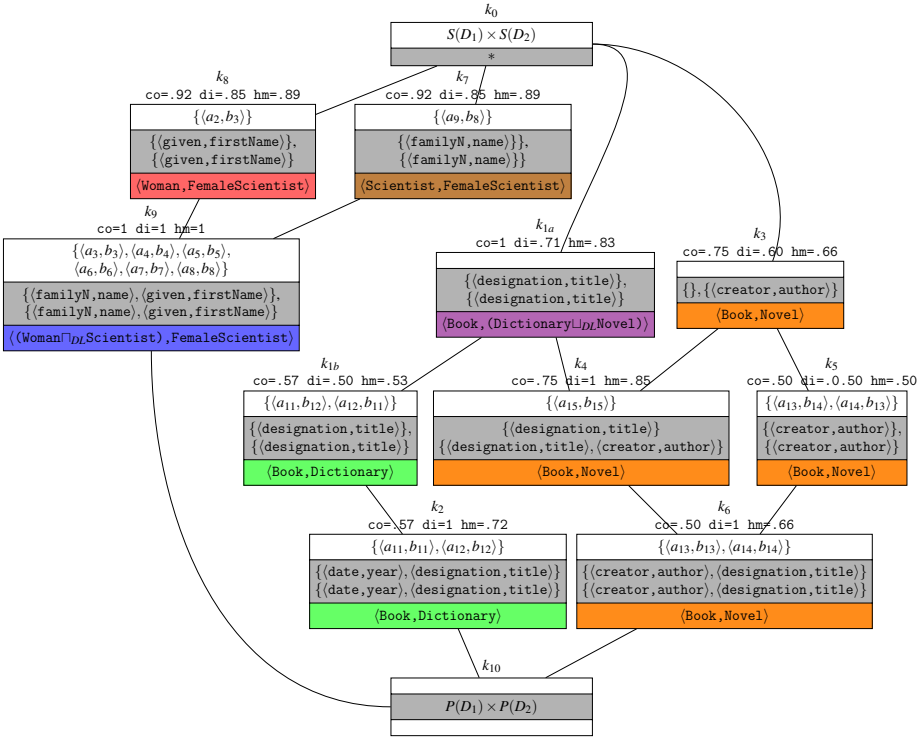

Fig. 3. Pattern concept lattice generated from the $LK-$pattern structure in Table 1

appearing in such link key candidates are used to describe subjects belonging to different classes. For example, the property `title` appearing in k_{1a} and k_{1b} is used to describe the instances of the classes `Dictionary` and `Novel`. Furthermore, we may notice that one pair of classes can be associated with more than one link key candidate. For example, we can see the pair of classes ⟨`Book,Novel`⟩ (in orange) which is associated with four link key candidates k_3, k_4, k_5 and k_6. This means that there are four possible choices to select a link key candidate for the pair ⟨`Book,Novel`⟩ among the four candidates k_3, k_4, k_5 and k_6.

Specifying the pairs of classes associated with a link key candidate is a critical task to properly evaluate this candidate. For example, the link key candidate k_4, in Figure 2, shows a low harmonic mean `hm=0.37`, because it is evaluated on the whole datasets. Consequently, k_4 is poorly ranked by a system based on FCA. This means that k_4 will not be returned as a relevant candidate despite the fact that it generates all the correct links between the classes `Book` and `Novel` while no other candidate is able to generate those links. By contrast, in Figure 3, k_4, shows a good harmonic mean `hm=0.85` because it is evaluated on the "right pair" of classes ⟨`Book,Novel`⟩. The candidate k_4 will be returned by a $LK-$pattern structure as a relevant candidate for the pair of classes ⟨`Book,Novel`⟩. Hence, we can appreciate the importance of introducing the notion of

$LK-$pattern structure and the discovery of link key candidates associated with pairs of classes.

4 Conclusion

Link keys are used to discover identity links across RDF datasets. In this paper, given two datasets, we propose a method based on pattern structures and introduce the notion of $LK-$pattern structure to discover link key candidates. An added value of the present method is to allow the discovery of link key candidates while specifying the classes to which they apply. This is a substantial improvement for properly evaluating the discovered link key candidates. For future work we plan to study the scalability and the efficiency of the method by running experiments on real-world datasets.We also intend to extend this research work by taking advantage of domain ontologies related to the datasets under study.

Acknowledgments

This work has been supported by the ANR project Elker (ANR-17-CE23-0007-01) and the BnF in the context of the agreement between Inria and Ministère de la culture.

References

1. Al-Bakri, M., Atencia, M., David, J., Lalande, S., Rousset, M.C.: Uncertainty-sensitive reasoning for inferring same as facts in linked data. In: Proceedings of ECAI. pp. 698–706 (2016)
2. Al-Bakri, M., Atencia, M., Lalande, S., Rousset, M.C.: Inferring same-as facts from linked data: an iterative import-by-query approach. In: Proceedings of AAAI (2015)
3. Atencia, M., David, J., Euzenat, J.: Data interlinking through robust linkkey extraction. In: ECAI. pp. 15–20 (2014)
4. Atencia, M., David, J., Euzenat, J.: What can FCA do for database linkkey extraction? In: Proceedings of FCA4AI workshop. pp. 85–92. No commercial editor. (2014)
5. Atencia, M., David, J., Euzenat, J., Napoli, A., Vizzini, J.: Link key candidate extraction with relational concept analysis. DAM. **273**, 2–20 (2020)
6. Baader, F., Calvanese, D., McGuinness, D., Patel-Schneider, P., Nardi, D., et al.: The description logic handbook: Theory, implementation and applications (2003)
7. Euzenat, J., Shvaiko, P.: Ontology Matching, Second Edition. Springer (2013)
8. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: ICCS. pp. 129–142. Springer (2001)
9. Ganter, B., Wille, R.: Formal Concept Analysis - Mathematical Foundations. Springer (1999)
10. Hitzler, P., Krotzsch, M., Rudolph, S.: Foundations of semantic web technologies. CRC press (2009)
11. Ngomo, A.C.N., Auer, S.: Limes—a time-efficient approach for large-scale link discovery on the web of data. In: Proceedings of IJCAI (2011)
12. Saïs, F., Pernelle, N., Rousset, M.C.: L2r: A logical method for reference reconciliation. In: Proceedings of AAAI. pp. 329–334 (2007)
13. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Silk-a link discovery framework for the web of data. LDOW **538** (2009)