Using Redescriptions and Formal Concept Analysis for Mining Definitions in Linked Data[⋆]

Justine Reynaud, Yannick Toussaint, and Amedeo Napoli

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
justine.reynaud@loria.fr, yannick.toussaint@loria.fr,
amedeo.napoli@loria.fr

Abstract. In this article, we compare the use of Redescription Mining (RM) and Association Rule Mining (ARM) for discovering class definitions in Linked Open Data (LOD). RM is aimed at mining alternate descriptions from two datasets related to the same set of individuals. We reuse RM for providing category definitions in DBpedia in terms of necessary and sufficient conditions (NSC). Implications and AR can be jointly used for mining category definitions still in terms of NSC. In this paper, we firstly, recall the basics of redescription mining and make precise the principles of definition discovery. Then we detail a series of experiments carried out on datasets extracted from DBpedia. We analyze the different outputs related to RM and ARM applications, and we discuss the strengths and limitations of both approaches. Finally, we point out possible improvements of the approaches.

Keywords: Redescription Mining - Association Rule Mining - Concept Analysis - Linked Open Data - Definition of categories

1 Introduction

The Linked Open Data (LOD) cloud is a very large reservoir of data based on elementary triples *(subject, predicate, complement)*[1], where a triple is denoted as $\langle s, p, c \rangle$, with s, p and c denoting resources. These triples can be related to form a (huge) directed graph $G = (V, E)$ where vertices in V correspond to resources –also termed as individuals– and edges in E corresponds to relations or predicate linking resources. A specific ordering can be integrated within this graph structure. Individuals can be grouped in a class using the Resource Description Framework (RDF) thanks to the special predicate `rdf:type`, and then individuals are "instances" of this class. In turn, using RDF Schema (RDFS),

[⋆] Paper accepted at ICFCA (International Conference on Formal Concept Analysis, June 25 – 28, Frankfurt, Germany). Supported by "Région Lorraine" and "Délégation Générale de l'Armement"

[1] The elements of a triple are usually referred as *(subject, predicate, object)*. For avoiding any confusion with objects from FCA, we adopt here the term *complement* instead of *object*.

the set of classes can be organized within a poset thanks to the partial ordering `rdfs:subClassOf`.

A class can be defined through an *extension* by enumeration of all individuals composing this extension. For example, the extension of the `Smartphone` class would include the set of all “known” smartphones in a given universe. Dually, a class may also be defined through an *intension* by enumeration of all characteristics common to individuals in the class. For example, the intension of the `Smartphone` class could be described as “a small computer equipped with a cellular antenna”. It should be noticed that “extensions” and “intensions” are not necessarily closed sets as extents and intents are in Formal Concept Analysis (FCA [11]).

A basic classification problem, related to clustering and FCA, is to provide a suitable definition to a class of individuals, i.e. a description based on a set of characteristics which are common to all individuals. This problem arises whenever there is a need for building classes for an ontology or a knowledge base related to a particular domain. In the LOD cloud, this classification problem takes a specific form. There are classes defined by an extension but usually without any corresponding intension. More concretely, we may consider individuals as subjects s whose description is composed of the set of available pairs (p, c). An application of this classification problem is related to the mining of definitions of DBpedia categories, in the line of the work in [2].

Actually, DBpedia categories are automatically extracted from Wikipedia. In Wikipedia, a category is a specific page which lists all the pages related to itself, as this is the case for example for the page `Category:Smartphones`[2]. In DBpedia, a category is a resource appearing in the range of the predicate `dct:subject`. Moreover, categories are widespread as there is more than one million of categories but, most of the time, a category does not have any “processable” description and there does not exist any ordering or structure among categories.

Accordingly, given a class defined by a set of instances, the classification problem aims at finding a corresponding definition in terms of a description made of sets of characteristics or properties related to all these instances. Afterwards, the class can be defined in terms of necessary and sufficient conditions for an individual to be a member of the class. The necessary condition means that all instances share the characteristics of the description while the sufficient condition means that any individual having those characteristics should be an instance of the class.

Actually, the present work is a continuation of a work initiated in [2] and in [19]. In the first paper [2], authors rely on FCA [11] and implication between concepts for discovering definitions in LOD. These definitions are based on pairs of implications, i.e. $C \implies D$ and $D \implies C$, which stand for necessary and sufficient conditions. A double implication is considered as a definition $C \equiv D$. Most of the time $C \implies D$ is an implication while $D \longrightarrow C$ is an association rule whose confidence is less than 1. This means that a plausible definition can

[2] https://en.wikipedia.org/wiki/Category:SmartPhones

be set provided that the data at hand are completed. In the second paper [19], authors propose a preliminary comparison between three approaches for mining definitions in LOD, (i) FCA, (ii) redescription mining and (iii) translation rule mining.

In the present paper, we make precise and discuss the mining of definitions in LOD using FCA and Redescription Mining (RM) [8,9]. RM aims at discovering alternate characterisations of a set of individuals from two sets of characteristics. The characterisations can be expressed thanks to Boolean connectors within propositional logic formulas. As experiments demonstrate it, FCA and RM are able to discover definitions which can be quite different, showing that both methods are indeed complementary and also the interest of such a comparison.

The paper is organised as follows. Problem statement and FCA are introduced in Section 2. Redescription Mining is detailed in Section 3, while Section 4 includes experiments which were carried out to evaluate the comparison and a discussion on the quality of the results. Section 5 discusses the Related work. Finally the conclusion ends the paper and sketches future work.

2 Problem statement

2.1 Basics of FCA and Association Rules

We rely on Formal Concept Analysis (FCA) [11] in order to represent our data. Given G a set of objects, M a set of attributes and $I \subseteq G \times M$ a binary relation between G and M, (G, M, I) is a formal context. Derivation operators (denoted $.'$) for a set of objects $A \subseteq G$ and a set of attributes $B \subseteq M$ are $A' = \{m \in M \mid \forall a \in A, aIm\}$ and $B' = \{g \in G \mid \forall b \in B, gIb\}$. The two compositions of the both derivation operators, denoted by $(.)''$, are closure operators.

A pair (A, B) is a "concept" whenever $A' = B$ and $B' = A$, where A and B are closed. A and B are called the "extent" and the "intent" of the concept respectively. The set of concepts is organized within a "concept lattice" thanks to the partial ordering defined by $(A_1, B_1) \leq (A_2, B_2)$ when $A_1 \subseteq A_2$ or dually $B_2 \subseteq B_1$.

An association rule between two sets of attributes A and B, denoted $A \rightarrow B$ means that "if we observe A, then we observe B" with a *confidence* which can be considered as a conditional probability:

$$\text{conf}(A \rightarrow B) = \frac{|A' \cap B'|}{|A'|}$$

where $(.)'$ corresponds to the derivation operator. An association rule is valid if its confidence is superior to a given threshold θ. When $conf(A \rightarrow B) = 1$, the rule is an implication, denoted by $A \Rightarrow B$. Moreover, if $B \Rightarrow A$, then A and B form a definition, denoted by $A \equiv B$.

2.2 Defining Categories in DBpedia

The content of *DBpedia* [17] is built with information extracted from *Wikipedia*, an online encyclopedia. In *Wikipedia*, a category, say X, is a specific kind of Wikipedia page listing all pages related to X (see page `Category:Smartphones`[3] for example). These categories are annotated by the users of *Wikipedia*. In *DBpedia*, a category appears in RDF triples in the range of the relation `dct:subject`. For example, the triple $\langle \mathtt{x}, \mathtt{dct}:\mathtt{subject}, \mathtt{Smartphones}\rangle$ states that the `x` subject belongs to the `Smartphones` "category".

Moreover, speaking in terms of knowledge representation and reasoning, the name of a category is a purely syntactic expression, and thus a category does not have any formal definition as one could expect (see discussion in [2] on this aspect). Then it is impossible to perform any classification within the set of categories as the latter are not defined in terms of necessary and sufficient conditions. This is precisely what we want to deal with, i.e. providing a definition to a category. This amounts to find pairs of the form $(C, \{d_1, \ldots, d_n\})$ where C denotes a category, such as `Nokia_Mobile_Phone` for example, and d_i denotes a pair (p, c), such as `(manufacturer,Nokia)` for example. Then the whole set of d_i will stand for a possible description of C in terms of a list of $(\textit{attributes}, \textit{values})$ pairs. A parallel can be drawn with concept definitions in Description Logics [3], where a form of definition is given by $C \equiv d_1 \sqcap \cdots \sqcap d_n$, such as:

$$\mathtt{Nokia\_Mobile\_Phone} \equiv \mathtt{Phone} \sqcap \exists \mathtt{manufacturer}.\mathtt{Nokia}$$

These definitions are useful for a practitioner aiming at contributing to *DBpedia*. Indeed, providing descriptions and then definitions to categories allows to be in agreement with knowledge representation principles, i.e. building sound and complete definitions of individual classes, as categories should be. In particular, this would help to find missing triples. For example, suppose that the definition $\mathtt{Nokia\_Mobile\_Phone} \equiv \mathtt{Phone} \sqcap \exists \mathtt{manufacturer}.\mathtt{Nokia}$ is lying in *DBpedia*. Then, if an element `x` belongs to `Nokia_Mobile_Phone`, then this element should be a phone with manufacturer Nokia, i.e. `x` is an instance of $\mathtt{Phone} \sqcap \exists \mathtt{manufacturer}.\mathtt{Nokia}$ ("necessary condition"). Conversely, if an element is an instance of $\mathtt{Phone} \sqcap \exists \mathtt{manufacturer}.\mathtt{Nokia}$, then `x` should be an instance of `Nokia_Mobile_Phone` ("sufficient condition"). This allows to complete incomplete triples if required.

2.3 A Practical Approach in FCA

Following the lines of [2] in the FCA framework, the discovery of category definitions relies on the construction of a context (G, M, I) from a set of triples denoted by ST. Given ST, G is the set of subjects, i.e. $G = \{s \mid \langle s, p, c\rangle \in ST\})$ and M is a set of pairs predicate-complement, i.e. $M = \{(p, c) \mid \langle s, p, c\rangle \in ST\})$. The incidence relation is defined as $sI(p, c) \iff \langle s, p, c\rangle \in ST$.

[3] https://en.wikipedia.org/wiki/Category:Smartphones

The set of pairs is partitioned into two sets: $M = M_{subj} \cup M_{descr}$ and $M_{subj} \cap M_{descr} = \emptyset$. The set M_{subj} is the set of all pairs (p, c) such that $\texttt{p} = \texttt{dct:subject}$. Since all the resources in the range of `dct:subject` are categories, hereafter, a pair $(\texttt{dct:subject}, C)$ will simply be denoted C, where C corresponds to the label of a category we are trying to define and will be referred as a *category*. The set M_{descr} is the set of all pairs (p, c) such that $\texttt{p} \neq \texttt{dct:subject}$. Hereafter, a pair $(p, c) \in M_{descr}$ will be referred as a *description* and denoted $\exists p{:}c$ where c is an abbreviation of an abstract class containing only c.

Then the discovery process is based on a search for implications of the form $B_1 \implies B_2$ where $B_1, B_2 \subseteq M$. Whenever an implication $B_1 \implies B_2$ is discovered, the converse rule is checked. If $B_2 \implies B_1$ is also an implication, then we have the definition $B_1 \equiv B_2$. If this is not the case, the set of triples involved in the context should be checked for potential incompleteness. One of the drawbacks of this approach is that it relies on implications. However, due to the incompleteness of LOD, a large number of definitions may be missed. To overcome this issue, an association rule $B_i \to B_j$ can be considered together with its converse $B_j \to B_i$, and we can wonder how far they are from being implications. Accordingly, in a previous work [19], we introduced the notion of a quasi-definition which is to a definition what an association rule is to an implication.

Definition 1 (Quasi-definition). *Given two sets of attributes B_i, B_j and a user-defined threshold θ, a quasi-definition $B_i \leftrightarrow B_j$ holds if $B_i \to B_j, B_j \to B_i$ and*

$$\min(\mathrm{conf}(B_i \to B_j), \mathrm{conf}(B_j \to B_i)) \geqslant \theta$$

The algorithm `Eclat` [21] is one of the existing algorithms used to compute association rules. It exhaustively enumerates all the frequent itemsets, i.e. itemsets whose support is above a given threshold. Here, we rely on `Eclat`, implemented in the `Coron` platform [12], to mine association rules.

Since we want to provide definitions of categories, we are interested in rules $c \to \{d_1, \ldots, d_n\}$ or, conversely, $\{d_1, \ldots, d_n\} \to c$ such that $c \in M_{subj}$ and $d_i \in M_{descr}$. Given an association rule R: $B_1 \to B_2$, the consequent can be decomposed into two rules R_C: $B_1 \to B_C$ and R_D: $B_1 \to B_D$ where $B_C = B_2 \cap M_{subj}$ and $B_D = B_2 \cap M_{descr}$ respectively. Since $B_C \subseteq B_2$, $B_2' \subseteq B_C'$, thus $|B_1' \cap B_2'| \leqslant |B_1' \cap B_C'|$, which means that if R holds, then R_C holds. Similarly, if R holds, then R_D holds. We take advantage of this property to build the quasi-definitions we are interested in, that is rules of the form $c \leftrightarrow \{d_1, \ldots, d_n\}$. For example, $\{\exists r_1{:}x_1, C_0\} \to \exists r_2{:}x_2$ is not kept because the antecedent include both categories and descriptions. On the other hand, $\exists r_1{:}x_1 \to \{\exists r_2{:}x_2, C_2\}$ can be decomposed into R_1: $\exists r_1{:}x_1 \to \exists r_2{:}x_2$ and R_2: $\exists r_1{:}x_1 \to C_2$. The rule R_2 is kept. If its converse is valid, we obtain the quasi-definition $C_2 \leftrightarrow \exists r_1{:}C_1$.

In the following, we present an alternative search for category definition based on "Redescription Mining" (RM), where the name of the category appears on the left hand side of the $\equiv$ symbol and a set of characteristics (composed of $\exists$*predicate.complement* expressions) appears on the right hand side.

3 Redescription Mining

3.1 Definitions

Redescription mining aims at searching for data subsets with multiple descriptions, as different views on the same set of objects [9]. Redescription mining takes as input a set of objects G and a set of attributes M partitioned into *views* V_i such as $M = V_1 \cup \cdots \cup V_n$ and $V_i \cap V_j = \emptyset$ if $i \neq j$. For example, the attributes can be partitioned w.r.t. the sources of the data (two different databases for example) or w.r.t. some criteria defined by a user. A value is associated to each pair $(object, attribute)$, which can be Boolean, numerical or nominal, and which depends on the domain of the attribute. An example of such a dataset is provided in Figure 1.

Views	V_1			V_2
Attributes	a_1	a_2	a_3	a_4
f_1		2	3	Triangle
f_2		3	3	Triangle
f_3	×	0	3	Triangle
f_4	×	2	3	Triangle
f_5	×	2	4	Rectangle

a_1: Has a right angle (Boolean)
a_2: Max number of equal sides (numerical)
a_3: Total number of sides (numerical)
a_4: Type (nominal)

Fig. 1: An example of dataset for redescription mining, with objects $\{f_1, \ldots, f_5\}$ and attributes $\{a_1, a_2, a_3, a_4\}$.

Given a set of objects G, a partition of a set of attributes M, redescription mining aims at finding a pair of "queries" (q_1, q_2), where q_1 and q_2 correspond to logical statements involving attributes and their values. These statements are expressed in propositional logic with the conjunction, disjunction and negation connectors. Below, a redescription say RD based on the pair (q_1, q_2) is denoted by $RD = q_1 \longleftrightarrow q_2$ or $RD = (q_1, q_2)$.

Given a redescription $RD = q_1 \longleftrightarrow q_2$, the set of objects G can be partitioned w.r.t. the queries which are satisfied by a subset of objects. There are four possible components in the partition, denoted by E_{ij} with $i, j \in \{0, 1\}$, depending on the fact that q_1 and/or q_2 are satisfied. For example, $E_{10}(RD)$ denotes the set of objects satisfying q_1 but not q_2 and $E_{11}(RD)$ denotes the set of objects satisfying both q_1 and q_2.

Redescriptions are mined w.r.t. a support, the Jaccard coefficient, and a p-value. The support of a redescription $RD = (q_1, q_2)$ is the proportion of objects in the dataset satisfying both queries q_1 and q_2, i.e. support$(RD) = \frac{|E_{11}(RD)|}{|G|}$.

The similarity between two datasets corresponding to two queries q_1 and q_2 is measured thanks to the Jaccard coefficient:

$$\text{jacc}(q_1 \leftrightarrow q_2) = \frac{|E_{11}(RD)|}{|E_{11}(RD)| + |E_{10}(RD)| + |E_{01}(RD)|}$$

Let us consider for example the redescription

$$RD : (a_2 \ = \ 2) \longleftrightarrow (a_4 \ = \ Triangle)$$

which is based on $q_1 = (a_2 = 2)$ and $q_2 = (a_4 = Triangle)$ w.r.t. the dataset in Figure 1. We have that $|E_{11}(RD)| = |\{f_1, f_4\}| = 2$, $|E_{10}(RD)| = |\{f_5\}| = 1$, $|E_{01}(RD)| = |\{f_2, f_3\}| = 2$ and $|E_{00}(RD)| = |\emptyset| = 0$. Then it comes that $\text{support}(RD) = \frac{2}{5}$ and $\text{jacc}(RD) = \frac{2}{2+1+2} = \frac{2}{5}$. If the threshold for the Jaccard coefficient is $\frac{1}{2}$, then the redescription cannot be retained. By contrast, the redescription $(a_2 = 2) \wedge (a_3 = 3) \longleftrightarrow (a_4 = Rectangle)$ returns a Jaccard coefficient of $\frac{1}{2}$, meaning this time it can be accepted.

The significance of a redescription is computed w.r.t. the p-value. For a redescription $RD = (q_1, q_2)$, the p-value is the probability that $|E_{11}(RD)|$ is at least as much as computed, knowing the support of q_1 and q_2 and assuming they are random independant sets. In other words, we should answer the question "is the Jaccard computed due to random chance?". The p-value varies between 0 and 1, the lower it is, the more significant is the redescription. A p-value under the threshold 0.05 means that the computed $|E_{11}(RD)|$ is not due to random chance and that the redescription can be accepted as such [9,18].

3.2 A Redescription Mining Algorithm

In this paper, we reuse the `ReReMi` algorithm to mine redescriptions [9]. `ReReMi` takes two files F_1 and F_2 as input, which correspond to two subsets of attributes or "views" V_1 and V_2 in the dataset, and returns a set of redescriptions.

Firstly, a "candidate redescription" based on a given set of pairs (q_1, q_2), where q_1 contains only one attribute $\{a_1\} \subseteq V_1$ and q_2 only one attribute $\{a_2\} \subseteq V_2$, is checked. The checking is not necessarily systematic for all possible pairs or combinations of pairs of attributes, as a set of initial pairs can be specified by an analyst. Doing so, the set of candidate redescriptions is progressively extended, i.e. one attribute is added at a time to one of the queries of the candidate redescription.

A query q can be extended with a new attribute a in four possible ways: $q_1 \wedge a$, $q_1 \vee a$, $q_1 \wedge \neg a$ or $q_1 \vee \neg a$. The redescription with the best Jaccard coefficient is added to the candidate redescriptions. However, this extension can be customised using for example only one of the possibilities, e.g. $q_1 \wedge a$. The algorithm is based on a beam search: at each step, the top k pairs of queries with the higher Jaccard coefficient are extended. The algorithm continues until there is no more candidate available, i.e. until there is no way to increase the Jaccard coefficient of the current candidate redescription or there is no more attributes available to extend the query. A maximal depth can also be specified by the analyst. Since the algorithm is based on a greedy approach, there is no guarantee to obtain the best redescriptions: the algorithm may stop at a local maxima. Finally, the set of the candidate redescriptions is returned to the analyst.

Table 1: Datasets extracted.

	Persons	Objects	Films
Small	Turing_Award	Samsung_Galaxy	Hospital_films
Medium	Women_Mathematicians	Smartphones, Sports_cars	Road_movies
Large	Mathematicians	—	French_films

Table 2: Statistics on the datasets extracted.

Dataset	Triples	$\|G\|$	$\|M\|$	$\|M_{subj}\|$	$\|M_{descr}\|$	Predicates	Density
Samsung_Galaxy	940	59	277	30	247	33	5.2e−2
Turing_Award_laureates	2642	65	1360	503	857	35	2.2e−2
Hospital_films	1984	71	1265	490	775	46	1.6e−2
Women_mathematicians	9652	552	4243	1776	2467	98	2.9e−3
Smartphones	8418	598	2089	359	1730	98	5.8e−3
Sports_cars	9047	604	2730	435	2295	61	4.7e−3
Road_movies	20056	689	9314	2652	6662	103	2.4e−3
Mathematicians	32536	1660	12279	3848	8431	202	1.2e−3
French_films	121496	6039	25487	6028	19459	111	6.4e−4

3.3 Redescription mining in Linked Open Data

For applying redescription mining to a set of linked data, i.e. a set of related RDF triples, we need first to transform this set of triples into a format that can be processed by the ReReMi algorithm. This operation is similar to the building of a context in the FCA framework. The attributes correspond to the predicates of the triples and they are separated into views.

We build an input "context" as described in section 2.3. The two views correspond to the two sets of the partition of attributes, that is, $M = M_{subj}$ and M_{desc}. Based on that, searching for category definitions can be achieved by searching for redescriptions (q_1, q_2) where $q_1 = a$ with $a \in M_{subj}$ and q_2 is a query based on a set of one or more attributes from M_{desc}. Actually, this search should output a definition based on characteristics shared by all the resources of the category, that is, a set of necessary and sufficient conditions for being a member of the category.

4 Experiments

4.1 Datasets

We extracted 9 different subsets of triples with various sizes[4], which cover three domains : Persons, Manufactured objects and Films (see in Table 1). The

[4] The datasets and the results of the experiments are available online, see https://gitlab.inria.fr/jreynaud/icfca19.

small datasets have less than 100 objects and 1500 attributes, meaning that there are less than 1500 unique pairs $(predicate, complement)$ in the extracted triples. The medium datasets have around 600 objects and between 2000 and 10000 attributes. Finally, the large datasets have more than 1500 objects and 10000 attributes. There is no manufactured object dataset of this size, therefore only two large datasets about persons and films are provided.

Statistics are given in Table 2. Overall, all the datasets are sparse, meaning that attributes have a low support. However, the density of manufactured objects seems to be higher than the density of persons and films. The number of predicates is low regarding the number of pairs (predicate, complement). This means that a lot of attributes share the same predicate and differ only on the complement.

4.2 Inputs

For mining association rules we used the Coron platform [12] with the `Eclat` algorithm. We set the minimum confidence to 0.5 and the minimum support to 3, 5 or 10, depending whether the dataset is small, medium or large respectively. The input file is a context built from all the triples, with all the attributes in M whenever they are categories or descriptions.

For mining the redescriptions, the attributes are partitioned. For each of the datasets, the partition of the attributes is built as follows: M_{subj} is constructed from the subset of triples whose predicate is `dct:subject` whereas M_{desc} is the complementary set. Here, there are only Boolean attributes and only conjunction is used in RM. From M_{subj} and M_{desc}, two tabular files compliant with `ReReMi` input are created, namely D_{subj} which contains attributes of the view M_{subj}, D_{desc} which contains attributes of the view M_{desc}. The thresholds used are 0.5 for Jaccard similarity (jacc $\geqslant 0.5$) and 3 for support (support $\geqslant 3$).

4.3 Extraction of Definitions

The `ReReMi` algorithm returns a set of redescriptions with their respective Jaccard coefficients. The `Eclat` algorithm returns a set of association rules that need to be processed. From the set of mined association rules, we build quasi-definitions $c \leftrightarrow \{d_1, \ldots, d_n\}$.

For measuring the precision of the algorithms, each rule (redescription or quasi-definitions) is manually evaluated by an analyst. Hereafter, a rule which is considered as "valid" by the expert is called a definition. This allows us to compute the precision as follows:

$$Prec_{RD} = \frac{|D_{RD}|}{|RD|} \quad \text{and} \quad Prec_{QD} = \frac{|D_{QD}|}{|QD|}$$

where RD (resp. QD) corresponds to the set of redescriptions (resp. association rules) and D_{RD} (resp. D_{QD}) corresponds to the number of redescriptions (resp. association rules) evaluated as valid by the domain expert. The set of definitions obtained from redescriptions is denoted D_{RD} and the set of definitions obtained from association rules is denoted D_{AR}.

Table 3 presents some redescriptions and quasi-definitions along with Jaccard coefficient and confidence for the datasets `Turing_Award_laureates` and `Smartphones`[5].

Table 3: Redescriptions and Association Rules extracted by `ReReMi` and `Eclat` for each the datasets `Turing_Award_laureates` and `Smartphones`, written in a Description Logics-like formalism. If the rule is valid (i.e. considered as true by the evaluator), the symbol $\equiv$ is used. Otherwise, the symbol $\not\equiv$ is used. The confidence corresponds to the minimal confidence between the two association rules $A \rightarrow B$ and $B \rightarrow A$.

N.	Redescriptions	jacc
	`Turing_Award_laureates`	
R1	Harvard_University_alumni ≡ ∃almaMater.Harvard_University	.89
R2	Stanford_University_alumni ≡ ∃almaMater.Stanford_University	.56
R3	National_Medal_of_Science_laureates ≡ ∃award.National_Medal_of_Science	1
R4	British_computer_scientists ≢ ∃award.Fellow_of_the_Royal_Society	.63
	`Smartphones`	
R5	Nokia_mobile_phones ≡ ∃manufacturer.Nokia	.82
R6	Samsung_Galaxy ≡ ∃manufacturer.Samsung_Electronics ⊓ ∃operatingSystem.Android_OS	.66
R7	Mobile_operating_systems ≡ Software ⊓ Work	.58
R8	MeeGo_Devices ≢ ∃operatingSystem.Sailfish_OS	.73
N.	Association Rules	conf
	`Turing_Award_laureates`	
R9	Harvard_University_alumni ≡ ∃almaMater.Harvard_University ⊓ Agent ⊓ Person ⊓ Scientist	.88
R10	Harvard_University_alumni ≡ ∃almaMater.Stanford_University ⊓∃award.Turing_Award ⊓ Agent ⊓ Person ⊓ Scientist	.75
R11	National_Medal_of_Science_laureates ≡ ∃award.National_Medal_of_Science ⊓ Agent ⊓ Person ⊓ Scientist	1
R12	Massachusetts_Institute_of_Technology_faculty ≢ ∃award.Turing_Award ⊓ Agent ⊓ Person ⊓ ∃birthPlace.New_York_City	.50
	`Smartphones`	
R13	Nokia_mobile_phones ≡ ∃manufacturer.Nokia ⊓ Device	.85
R14	Samsung_Galaxy ≡ ∃manufacturer.Samsung_Electronics ⊓ Smartphone ⊓ Device	.53
R15	Mobile_operating_systems ≡ Software ⊓ Work	.58
R16	Sony_mobile_phones ≢ Device ⊓∃ input.Capacitive_sensing ⊓∃ input.Proximity_sensor ⊓ ∃ input.Touchscreen	.64

There are many more quasi-definitions extracted than redescriptions, especially in the `French_films` dataset. In the domain of films, the rules extracted

[5] The entire set of redescriptions and quasi-definitions extracted are available online, see https://gitlab.inria.fr/jreynaud/icfca19.

are about directors, actors and distributors. In the domain of persons, they are about the universities they come from or the award they won. Finally, in the domain of objects, rules are about manufacturers and brands.

Most of the "invalid" mined redescriptions are based on a description which is too "approximate", i.e. there are possibly too many exceptions to the rule. For example, a large proportion of British computer scientists are also fellows of the Royal Society, but not all are award winners (see rule `R4`). In some other cases, there are not enough counter-examples in the dataset. For example, in redescription `R8`, there are too few `Meego` smartphones which are not running `Sailfish` in the dataset.

4.4 Discussion

Table 4: Results of the experiments for each dataset. In the redescription mining settings, the number of extracted redescriptions ($|RD|$) and evaluated as true ($|D_{RD}|$) are reported. In the association rules mining settings, in addition to the number of association rules extracted ($|AR|$), the number of quasi-definitions ($|QD|$) is reported. For both redescriptions and quasi-defintions, the number of categories that have been defined ($|Cat|$) is reported along with the precision.

	Redescriptions				Association Rules				
	$\|RD\|$	$\|D_{RD}\|$	$\|Cat\|$	P	$\|AR\|$	$\|QD\|$	$\|D_{QD}\|$	$\|Cat\|$	P
`Turing_Award_laureates`	33	16	16	0.48	563803	57	34	18	0.60
`Women_Mathematicians`	2	2	2	1.00	20483	6	5	5	0.83
`Mathematicians`	12	12	7	1.00	96807	29	24	14	0.83
`Samsung_Galaxy`	6	4	2	0.67	47004	20	20	5	1.00
`Smartphones`	24	22	17	0.92	3558380	34	26	12	0.76
`Sports_cars`	25	18	15	0.72	75030	49	35	17	0.71
`Hospital_films`	31	11	9	0.35	4345921	18	5	2	0.28
`Road_movies`	13	9	9	0.69	333491	34	18	13	0.53
`French_films`	6	6	4	1.00	371771	186	165	106	0.89

The number of extracted category definitions is reported in Table 4. The extracted rules depend on the data domain, and thus cannot be generalised to the whole *DBpedia*. For discovering more general definitions, we probably need to process larger datasets, e.g. considering a dataset about `Person` instead of `Turing_Award_laureates`. This would bring at the same time scalability issues that may be overcome with sampling. Further experiments in this direction could be considered in the future.

In Table 5, the number of predicates involved in definitions show that only a few predicates are involved in the definitions. Most of the time, there is only one attribute in the right side of a redescription, meaning that such an attribute is very discriminant and that redescriptions do not have any attribute in common. Then, it can be difficult to build a partial ordering between the defined categories.

Table 5: Number of predicates involved in the definitions extracted by ReReMi and Eclat.

	Pred.	Pred (D_{RD})	Pred (D_{QD})
Turing_Award_laureates	35	4	5
Women_Mathematicians	98	2	3
Mathematicians	202	4	5
Samsung_Galaxy	33	2	7
Smartphones	98	5	8
Sports_cars	61	3	5
Hospital_films	46	5	2
Road_movies	103	3	7
French_films	111	4	10

By contrast, with association rules, all attributes that may be added without loss of confidence are included. In the Turing_Award_laureates dataset for example, the redescription R1 and the quasi-definition R9 define the same category and both are valid. However, whereas R1 has only one attribute, R9 has four of them.

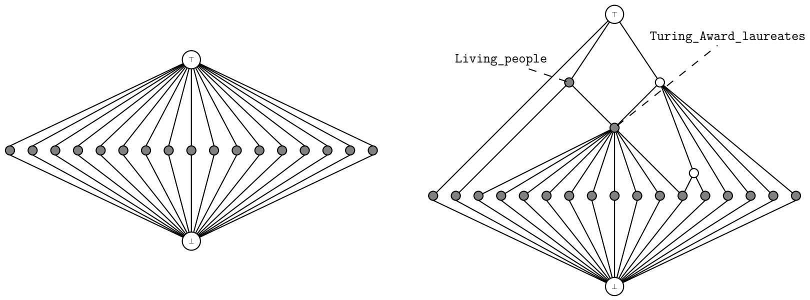

Fig. 2: Lattices build from the definitions obtained with redescriptions (left) and association rules (right) for the dataset Turing_Award_laureates. Gray nodes are object concepts, i.e. they define a category.

Simpler definitions like redescriptions may be useful and easier to understand. However, the attributes provided by quasi-definitions allow to build a classification. From the extracted rules, we built a context where G is the set of defined categories, and M is the set of attributes involved in a definition, and then we build the associated concept lattice. The results for the dataset Turing_Award_laureates are provided in Figure 2.

The Figure 3 presents the number and the precision of rules extracted. Compared to association rules, the number of redescriptions is 2 to 10 times less. In contrast with a previous assumption, there is no correlation between the num-

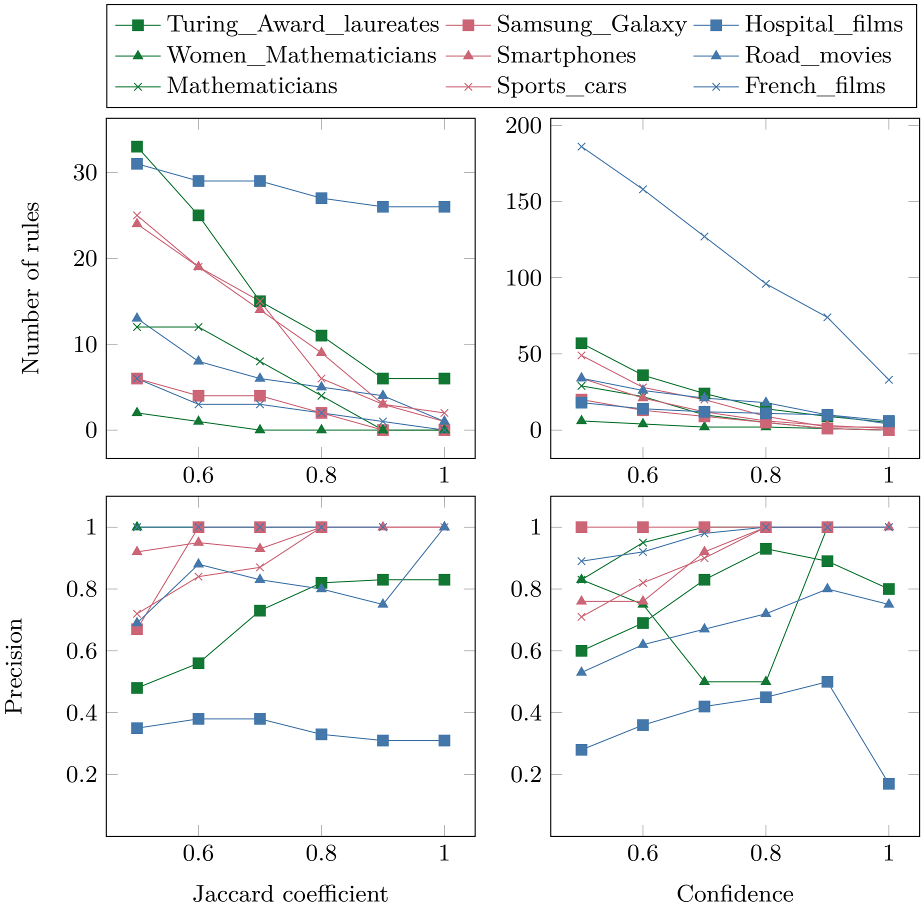

Fig. 3: Number and precision of the rules (redescriptions or quasi-defintions) extracted w.r.t. the Jaccard coefficient and the confidence.

ber of rules extracted and the density of the context [19]. However, `Smartphones` and `Sports_cars`, which are two similar datasets, have approximately the same number of extracted rules. This number could represent the “diversity” of the dataset: the more categories are defined, the more diverse is the dataset.

The precision increases w.r.t. the Jaccard coefficient threshold, meaning that the Jaccard coefficient is a suitable measure for redescription mining in LOD. Except for the dataset `Road_movies`, the Jaccard coefficient and confidence seem to return similar results for our task. The precision depends on the datasets and seems to be correlated to the number of extracted redescriptions. This would explain why redescriptions have a better precision than association rules.

The low precision of the `Hospital_films` dataset is hard to explain regarding its characteristics. However, from the extracted rules, it looks like this dataset suffers from an over-representation of some of its instances.

5 Related Work

The use of Formal Concept Analysis (FCA) [11] for mining LOD has been discussed in [13]. In order to mine association rules in LOD, some works rely on extensions of FCA, such as Logical Concept Analysis (LCA) [5] or Pattern Structures [10]. In [5], the authors propose a generalization of FCA where objects are variables and attributes are formulae in a given logic. In [10], the authors propose another generalisation of FCA, considering a partial order on attributes. Association rules [1,14] have been widely used in many applications, including LOD mining tasks. In these works, the mining task is performed either on RDF graphs or on the sets of triples. Other works focus on mining rules from assertions expressed in description logics.

In [6,7], the authors consider the RDF graph and propose the algorithm AMIE+, which mainly focuses on relations, without considering the domain and the range of the applications. The algorithm AMIE+ searches for association rules between relations of the form $r_1(x, y)$ and $r_2(x, z) \longrightarrow r_1(y, z)$ where x, y and z are random resources. For example, *people married to a person who lives in some place P also live in P* is the kind of rule that can be extracted by AMIE+.

In [4], an extension of FCA to conceptual graphs, called G-FCA, is proposed. Compared to RDF graphs, conceptual graphs are oriented bipartite graphs. The two kinds of nodes are classes in one hand and relations in the other hand. Contrasting RDF graphs which consider only binary relations, CGs handle n-ary relations. In this setting, concepts (called *Projected graph patterns*) have the following form: the intent corresponds to a graph pattern whereas the extent corresponds to the candidate solutions, i.e. the set of subgraphs matching the graph pattern. In [22], the authors use FCA in order to summarise RDF graphs. From an RDF graph, they build a formal context where objects are resources and attributes are classes and relations. The extracted concepts are used to produce a new RDF graph which summarise the original one.

Contrasting the other approaches, the authors in [20] rely on FCA and association rules to mine sets of triples. They build different formal contexts in order to discover specific relations, such as subsumption between two classes or transitivity of a relation for example. To this end, they build multiple contexts and mine association rules. The rules extracted can be interpreted as relations and they can be expressed in description logics. By contrast, in [15], the authors rely on rule mining and search for *obligatory class attributes*. Given a class, an obligatory attribute denotes a relation that every individual of the class should be involved in, e.g. every person has a birth date, and then `hasBirthdate` is an obligatory attribute of class `Person`. To this end, the authors focus on relations and are not interested in the range of relations. In [2], the authors take into account both relations and their range. They rely on pattern structures to build a context where objects are resources and attributes are pairs $\mathtt{A_i} = (\mathtt{predicate_i}, \mathtt{object_i})$. Then, implications $A_i \implies A_j$ are searched. Only implications whose converse has a high support are kept as candidate definitions.

We position ourselves in the continuity of these works. However, while most of the approaches search for implications and are based on association rules, we search for definitions and we use redescription mining. Redescription Mining (RM) [9] is predominantly used in application in the field of ecology such as for finding bioclimatic niches or properties of animal teeth. In [19] we propose a preliminary study in which we compare redescription mining to association rules mining and translation rule mining [16] applied to LOD. In this work, we extend our previous work and experiments and we focus on redescription mining and association rule mining.

6 Conclusion and Future Work

In this paper, we compared the use of redescription mining and association rule mining for discovering definitions of categories in DBpedia.

The experimental results show that the approach is well-founded, allowing to retrieve a subset of definitions. Compared to association rules, the definitions discovered by redescriptions are shorter. The Jaccard coefficient is well-suited for mining definitions in LOD. Other metrics used for association rules, such as stability and lift, could be used and compared to confidence and Jaccard coefficient.

Most of the time, the definition of a *DBpedia* category depends on only one attribute. Even if a large part of the observed results can be explained by the characteristics of the datasets, some artifacts remain, due to the data available in *DBpedia*. The dataset `Hospital_films` is one example of the results obtained when there are such artifacts.

The relation between predicates and complements have not been discussed here, and may be investigated in a future work.

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. ACM SIGMOD Rec. **22**(2), 207–216 (1993)
2. Alam, M., Buzmakov, A., Codocedo, V., Napoli, A.: Mining definitions from RDF annotations using Formal Concept Analysis. In: IJCAI. pp. 823–829 (2015)
3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook. Cambridge University Press (2003)
4. Ferré, S., Cellier, P.: Graph-FCA in Practice. In: Proceedings of 22nd ICCS. pp. 107–121 (2016)
5. Ferré, S., Ridoux, O.: A logical generalizationof formal concept analysis. In: Ganter, B., Mineau, G.W. (eds.) Conceptual Structures: Logical, Linguistic, and Computational Issues. pp. 371–384. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)

6. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.M.: AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In: WWW'13. pp. 413–422 (2013)
7. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast rule mining in ontological knowledge bases with AMIE+. VLDB Journal **24**(6), 707–730 (2015)
8. Galbrun, E., Miettinen, P.: Siren: An Interactive Tool for Mining and Visualizing Geospatial Redescriptions. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1544–1547. KDD'12, ACM (2012)
9. Galbrun, E., Miettinen, P.: Redescription Mining. Springer Briefs in Computer Science, Springer (2017)
10. Ganter, B., Kuznetsov, S.O.: Pattern Structures and their Projections. In: ICCS. pp. 129–142. LNCS 2120, Springer (2001)
11. Ganter, B., Wille, R.: Formal Concept Analysis – Mathematical Foundations. Springer (1999)
12. Kaytoue, M., Marcuola, F., Napoli, A., Szathmary, L., Villerd, J.: The Coron System. In: Boumedjout, L., Valtchev, P., Kwuida, L., Sertkaya, B. (eds.) 8th International Conference on Formal Concept Analsis (ICFCA) - Supplementary Proceedings. pp. 55–58 (2010), http://www.loria.fr/~kaytouem/publi/ICFCA10-demo-coron.pdf, (demo paper)
13. Kirchberg, M., Leonardi, E., Tan, Y.S., Link, S., Ko, R.K.L., Lee, B.S.: Formal Concept Discovery in Semantic Web Data. In: Domenach, F., Ignatov, D.I., Poelmans, J. (eds.) Proceedings of ICFCA. pp. 164–179. Springer (2012)
14. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., Verkamo, A.I.: Finding Interesting Rules from Large Sets of Discovered Association Rules. In: CIKM'94. pp. 401–407 (1994)
15. Lajus, J., Suchanek, F.M.: Are All People Married? Determining Obligatory Attributes in Knowledge Bases. In: International Conference WWW. Lyon, France (2018)
16. van Leeuwen, M., Galbrun, E.: Association Discovery in Two-View Data. TKDE **27**(12), 3190–3202 (2015)
17. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web **6**(2), 167–195 (2015)
18. Ramakrishnan, N., Kumar, D., Mishra, B., Potts, M., Helm, R.F.: Turning CARTwheels: an Alternating Algorithm for Mining Redescriptions. In: KDD'04. pp. 266–275 (2004)
19. Reynaud, J., Toussaint, Y., Napoli, A.: Three Approaches for Mining Definitions from Relational Data in the Web of Data. In: Proceedings of the 6th International Workshop FC4AI (co-located with IJCAI/ECAI). pp. 21–32. CEUR Proceedings Vol-2149 (2018)
20. Völker, J., Niepert, M.: Statistical Schema Induction. In: Extended Semantic Web Conference. pp. 124–138 (2011)
21. Zaki, M.J.: Scalable Algorithms for Association Mining. TKDE **12**(3), 372–390 (2000)
22. Zneika, M., Lucchese, C., Vodislav, D., Kotzinos, D.: RDF Graph Summarization Based on Approximate Patterns. In: Springer (ed.) 10th International Workshop on Information Search, Integration, and Personalization (ISIP 2015). pp. 69–87. Communications in Computer and Information Science 622 (2015)