Interactive Exploration over RDF Data using Formal Concept Analysis

Mehwish Alam
Université de Lorraine, LORIA, UMR 7503,
Vandoeuvre-lès-Nancy,
F-54506, France
Email: mehwish.alam@loria.fr

Amedeo Napoli
CNRS, LORIA, UMR 7503,
Vandoeuvre-lès-Nancy,
F-54506, France
Email: amedeo.napoli@loria.com

***Abstract*—With an increased interest in machine processable data, many datasets are now published in RDF (Resource Description Framework) format in Linked Data Cloud. These data are distributed over independent resources which need to be centralized and explored for domain specific applications. This paper proposes a new approach based on interactive data exploration paradigm using Pattern Structures, an extension of Formal Concept Analysis, to provide exploration and navigation over Linked Data through concept lattices. It takes RDF triples and RDF Schema based on user requirements and provides one navigation space resulting from several RDF resources. This navigation space allows user to navigate and search only the part of data that is interesting for her.**

I. Introduction

With the efforts of Semantic Web community many technologies have been offered for publishing machine-readable data on web. It annotates textual data with meta-data and makes it available in the form of ontologies and RDF graphs. One of the emerging source of data in the form of entity-relationship are published as Linked Open Data (LOD) cloud [1]. As a contrast to textual resources, LOD does not need extensive preprocessing as it is already annotated in the form of entities and relationships. This structured format leads to other kind of challenges. One of the basic characteristics of LOD is that it follows a *decentralized* publication model [2], meaning that the RDF graphs are published in several distributed resources, instead of creating one knowledge-base of statements any one can contribute new statements and make it publicly available. These resources have nothing in common except some shared terms. These decentralized graphs should be integrated through machine/software agents to provide domain specific applications. Moreover, external schemas in the form of ontologies or taxonomies can be linked to these data to make sense based on real world conception. Some of the resources in LOD only contain the schema without the instances such as SWRC ontology [3] and some semantic web documents may only contain RDF triples without the RDF Schema such as DBLP[1]. The problem of how to provide applications which allow guided navigation and exploration over these data sources still persists.

This paper introduces a framework based on interactive data exploration [4] paradigm using Pattern Structures [5] which is an extension of Formal Concept Analysis (FCA) [6]. The goal of exploratory data mining is to provide an expert with an insight into the data. One of the basic principals is to take into account the user-requirements and task-specific information. This way the patterns obtained by pattern mining algorithms are more relevant and interesting. For doing so, the pattern mining algorithms need to be combined with visualization tools for providing human-computer-interaction. Moreover, in order to make these pattern mining algorithms useful, there is a need to apply these algorithms to smaller datasets or only interesting and relevant subsets of the datasets. This enables user to interactively explore the data and identify patterns of interest.

In the current study, we use small datasets/subsets of datasets present in the form of RDF graphs over Linked Open Data and use FCA for giving the user an insight into the RDF datasets with the help of a visualization tool. During this process we directly involve the user in defining the datasets or part of datasets she is interested in. Then FCA is applied to these data and then patterns are obtained. Finally, the patterns computed are explored with the help of visualization tool. A complete iterative process of interactive data exploration on Linked Open Data is shown in Figure 1. This framework takes into account the prior requirements of the user and selects the data sources which are relevant to the user application distributed over several resources over Linked Open Data Cloud in the form of RDF triples and RDF Schema. This new approach called RDF-Pattern Structures, takes RDF triples and the RDF Schema present on distributed locations as an input and integrates them into one *navigation space*. This navigation space provides centralization over distributed RDF resources and keeps a partially ordered organization of RDF triples with respect to RDF Schema. It serves as a navigational as well as an interactive exploratory space for the user where she can identify the samples of the data which are not relevant to her while navigation. These identified samples are then hidden from the user and are recorded as irrelevant. In the second iteration a new navigation space is built on user demand which keeps only the information that is relevant to the user. This navigation space is explored and navigated for the purpose of data analysis and information retrieval over several data sources. To date this is the first attempt to deal with RDF data and exploration techniques with Pattern Structures and FCA.

The paper is structured as follows: section II gives the motivating example, section III introduces the background on the methodology used. Section IV details on creating a

[1]http://dblp.l3s.de/d2r/

navigation space for RDF. Section V explains the process of interactive data exploration over the obtained navigation space. Section VI describes the experimental results of the framework. Section VII discusses the related work while Section VIII concludes the paper.

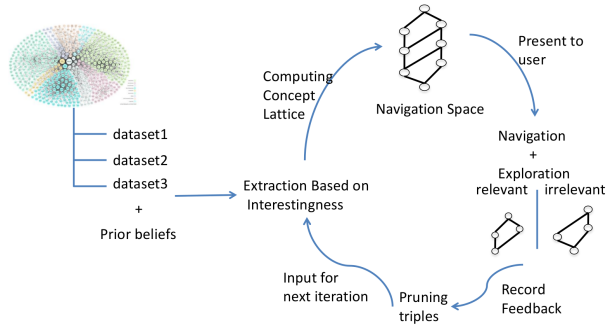


Fig. 1: Interactive Data Exploration over RDF Data through Concept Lattices

II. Motivating Example

Consider the scenario where the user wants to search for the papers published in conferences or journals related to her field of research. The problems faced by her for retrieving such papers are as follows:

- She looks-up DBLP page of the authors working in her field. In that case she has to go through all the publications of each author and then browse through the DBLP pages of the co-authors of this author.
- Moreover, if she is searching for the papers which are targeting more than one problem areas such as information retrieval and World Wide Web then it is not possible to retrieve such papers directly.
- Finally, she will also not be able to detect the communities of the author who often work together to retrieve the relevant papers or establish the collaboration with these authors.

Accordingly we try to guide such kind of research based on a concept lattice which is built from an initial query and then is explored by the user according to her preferences.

III. Preliminaries

A. Linked Open Data

Recently, Linked Open Data (LOD) [1] has become a standard for publishing data on-line in the form of Resource Description Framework (RDF)[2] which can further be linked to other data sources published in the same format. The idea underlying RDF is based on storing statements about a particular resource, where each statement is represented as $\langle subject, predicate, object \rangle$. A set of linked statements is referred to as RDF graph which constitutes an RDF triple store. For instance, Table II keeps the RDF triples for papers with their keywords and authors present on DBLP

[2]http://www.w3.org/RDF/

Index	Term	Index	Term
p_1	http://purl.org/dc/elements/1.1/subject (dc:subject)	p_2	http://purl.org/dc/elements/1.1/creator (dc:creator)
p_3	http://purl.org/dc/elements/1.1/title (dc:title)	p_4	rdfs:subClassOf
C_1	Web Crawling	C_8	Recommender Systems
C_2	Web Indexing	C_9	Clustering and Classification
C_3	Page and site Ranking	C_{10}	Web Search Engines
C_4	RDF	C_{11}	Semantic Web Description
C_5	OWL	C_{12}	World Wide Web
C_6	Similarity Measure	C_{13}	Retrieval Models and Ranking
C_7	Question Answering	C_{14}	Retrieval Tasks and Goals

TABLE I: Prefixes and Abbreviations of the terms used in the rest of the paper.

tid	Subject	Predicate	Object	Dataset
t1	s_1	p_1	o_{11}	DBLP
t2	s_1	p_2	o_{12}	DBLP
t3	s_2	p_1	o_{16}	DBLP
t4	s_2	p_2	o_{22}	DBLP
t5	s_1	rdf:type	Publication	DBLP
t6	o_{12}	rdf:type	Author	DBLP
t7	o_{11}	p_4	C_1	ACCS
t8	o_{12}	p_4	C_1	ACCS
t9	C_1	p_4	C_3	ACCS
⋮	⋮	⋮	⋮	⋮

TABLE II: RDF triples about papers with their authors and keywords from DBLP.

i.e., $t1, t2, t3, t4, t5, t6$. DBLP is provided by University of Mannheim in the form of RDF triples generated by using D2R Server [7] which provides mappings from SQL Databases to RDF. The prefixes and full forms of all the abbreviations used in this paper are shown in Table I. Consider $t1$ i.e., $\langle s_1, p_1, o_{11} \rangle$, here s_1 is subject, p_1 is a predicate and o_{11} is the object. Here, s represent the titles of the paper, p represent the predicates p_1, p_2, p_3, o represent the authors or keywords. Each resource is defined in the form of URI. The subject denotes the *resource* and the predicate denotes properties of the resource and defines relationship between the subject and the object. In the rest of the paper we use dereferenced resources i.e., s_1 instead of complete URI. The information about the background knowledge related to topics related to the keywords of the papers is represented in the ACM Computing Classification System[3](ACCS) and are shown in triples $t7, t8, t9$. For the sake of simplicity here we use only two resources.

B. SPARQL

A standard query language for RDF graphs is SPARQL[4] which mainly focuses on graph matching. A SPARQL query is composed of two parts the head and the body. The body of the query contains the Basic Graph Patterns (present in the WHERE clause of the query). It is composed of complex graph patterns that may include RDF triples with variables, conjunctions, disjunctions and constraints over the values of the variables. These graph patterns are matched against the RDF graph and the matched graph is retrieved and manipulated according to the conditions given in the query. The head of the query is an expression which indicates how

[3]https://www.acm.org/about/class/2012
[4]http://www.w3.org/TR/rdf-sparql-query/

the answers of the query should be constructed. Continuing the scenario in section II, the required data sets will be DBLP. A subset of these triples is selected based on user needs. For instance if user only wants the papers from the field of classification then for extracting this information the SPARQL query is given in Listing 1.

```
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX dc:<http://purl.org/dc/terms/>

SELECT distinct ?title ?keywords ?author
where {
?paper dc:creator ?a .
?a rdfs:label ?author .
?paper dc:subject ?keywords .
?paper dc:title ?title .
FILTER(
   regex(STR(?keywords), "pattern based classification", "i")
|| regex(STR(?keywords), "unsupervised classification", "i"))
```

Listing 1: SPARQL for extracting triples. Prefixes are also defined in Table I.

C. Formal Concept Analysis

Formal Concept Analysis (FCA) [6] is a mathematical framework used for a number of purposes, among which are classification and data analysis, information retrieval and knowledge discovery [8]. This section is based on the definitions from [6]. A formal context $\mathcal{K} = (G, M, I)$, consists of a set of entities[5] G and a set of attributes M and a binary relation I between G and M.

Definition 1 (Formal Context): A formal context $\mathcal{K} := (G, M, I)$, consists of two sets, a set of objects G and a set of attributes M and a binary relation I between G and M. The binary relation $(g, m) \in I$ or gIm is interpreted as "object g is in a relation I with an attribute m".

This formal context is represented as a binary table. Table III presents a formal context related to papers and their authors. The titles of the paper are kept as entities while their authors are kept as attributes in the context. The fact that the paper has an author defines a relationship I and is represented as a cross in the binary context. According to the first row in Table III, the paper s_1 has the author o_{21}.

From this binary context formal concepts are obtained keeping the classes of entities sharing some attributes. These concepts are computed by applying derivation operators. Given $A \subseteq G$ and $B \subseteq M$, two derivation operators, both denoted by $'$, formalize the sharing of attributes for objects. Dually, the sharing of objects for attributes:

$$A' = \{m \in M \mid gIm \ for\ all\ g \in A\} \quad (1)$$

$$B' = \{g \in G \mid gIm \ for\ all\ m \in B\} \quad (2)$$

The two derivation operators $'$ form a *Galois connection* between the powersets $\wp(G)$ and $\wp(M)$. Maximal sets of objects related to maximal set of attributes correspond to closed sets of the composition of both operators $'$ (denoted by $''$).

[5]we name objects in FCA as *entity* to avoid confusion with the object in an RDF triple.

	p_2				
Titles	o_{21}	o_{22}	o_{23}	o_{24}	o_{25}
s_1	×				
s_2		×	×		
s_3		×		×	×
s_4			×		
s_5		×	×		×

TABLE III: Formal Context $\mathcal{K}$.

Definition 2 (Formal Concept): A formal concept of the context $\mathcal{K} := (G, M, I)$ is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. A is the extent and B is the intent of the concept (A, B). $\mathcal{B}(G, M, I)$ denotes the set of all concepts of the context (G, M, I).

Consider the binary context in Table III, the pair $(\{s_2, s_5\}, \{o_{22}, o_{23}\})$ is a formal concept because $\{s_2\}'$ = $\{o_{22}, o_{23}\}$. and $\{o_{22}, o_{23}\}'$ = $\{s_2, s_5\}$, which means that the set of authors which are common to s_2 and s_5 are $\{o_{22}, o_{23}\}$. It is represented as a maximal rectangle, highlighted in grey in Table III.

Let $\mathcal{B}(G, M, I)$ be the set of all formal concepts for $\mathcal{K} = (G, M, I)$. Let $K\#1 = (A_1, B_1)$ and $K\#2 = (A_2, B_2)$ be two concepts, then $K\#1$ is a subconcept of $K\#2$ and $K\#2$ is a superconcept of $K\#1$, denoted by $K\#1 \leqslant K\#2$, iff $A_1 \subseteq A_2$ and $B_2 \subseteq B_1$. An example of $\leqslant$-relation between two concepts with respect to Table III will be: $(\{s_5\}, \{o_{22}, o_{23}, o_{25}\}) \leqslant (\{s_2, s_5\}, \{o_{22}, o_{23}\})$ (see Figure 2). Figure 2 shows a complete lattice for Table III. In this figure the concept lattice is labeled using reduced labeling which means that all the sub-concepts of a concept also contain the attributes present in the intent of that concept by inheritance.

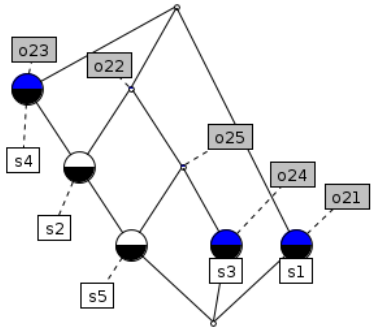


Fig. 2: Concept Lattice for the context in Table III

D. Pattern Structures

FCA [6] can process only binary context, more complex data such as graphs can not be directly processed by FCA. The concept lattice obtained by a binary context mixes between several types of attributes. Pattern structures, an extension of FCA, allows direct processing of such kind of context. The pattern structures were introduced in [5]. A pattern structure is a triple $(G, (D, \sqcap), \delta)$, where G is the set of entities, $(D, \sqcap)$ is a meet-semilattice of descriptions D and $\delta : G \rightarrow D$ maps an entity to a description. More intuitively, a pattern structure is the set of entities with their descriptions with a

	m_1	m_2	m_3
g_1	5	7	6
g_2	6	8	4
g_3	4	8	5

TABLE IV: Context with Numeric Data.

similarity operation $\sqcap$ on them which represents the similarity of entities. This similarity measure is idempotent, commutative and associative. If $(G, (D, \sqcap), \delta)$ is the pattern structures then the derivation operators can be defined as:

$$A^{\square} := \prod_{g \in A} \delta(g) \qquad for\ A \subseteq G$$
$$d^{\square} := \{g \in G | d \sqsubseteq \delta(g)\} \qquad for\ d \in D$$

Each element in D is referred to as a *pattern*. The subsumption order over these patterns is give as follows:

$$c \sqsubseteq d \Leftrightarrow c \sqcap d = c$$

The operator $(.)^{\square}$ makes a Galois connection as described in section III-C. Now the pattern concept can be defined as follows:

Definition 3 (Pattern Concept): A pattern concept of a pattern structure "$G, (D, \sqcap), \delta$" is a pair (A, d) where $A \subseteq G$ and $d \in D$ such that $A^{\square} = d$ and $A = d^{\square}$, where A is called the concept extent and d is called the concept intent.

Let us take an example of numeric data being processed by pattern structures in Table IV, The first record shows that the entity g_1 has the numeric value 5. The mapping $\delta : G \rightarrow D$ is given by $\delta(g_1) = \langle [5,5], [7,7], [6,6] \rangle$ which returns a description of a set of entities as a set of attributes. The lattice operation in the semi-lattice $(\sqcap)$ is called a similarity operation between two descriptions. For example, let $\delta(g_1) = \langle [5,5], [7,7], [6,6] \rangle$ and $\delta(g_2) = \langle [6,6], [8,8], [4,4] \rangle$ then $\delta(g_1) \sqcap \delta(g_2) = \langle [5,6], [7,8], [4,6] \rangle$. Following the definition of a pattern concept in definition 3, $(\{g_1, g_2\}, \langle [5,6], [7,8], [4,6] \rangle)$ is a pattern concept. The obtained concept lattice is called as *pattern concept lattice* (see Figure 3)). The pattern structure dealing with numeric data is called as interval pattern structure [9].

IV. Towards RDF-Pattern Structures

For creating navigation spaces over RDF as well as RDF Schema to provide navigation and exploration, the first operation is to select the RDF data set with no RDF Schema information and a suitable RDF Schema associated to these RDF triples from distributed data sources. This RDF schema will be used for organizing the RDF triples. An RDF Schema which will be used as a reference for comparing objects in the RDF triples is referred to as *reference schema*. This RDF Schema could also be a semantic resource such as Word Net or YAGO [10] or DBpedia ontology. In the running scenario as we are targeting the organization of RDF triples in DBLP, we use ACCS as our reference schema.

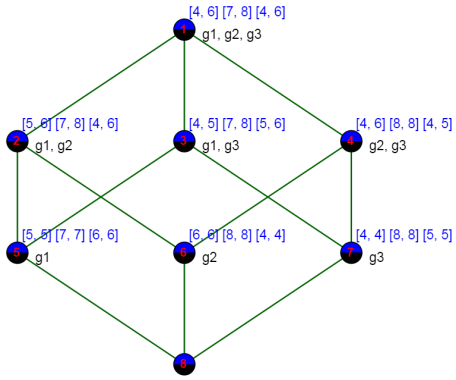


Fig. 3: Pattern Concept lattice for Table IV.

A subset of RDF triples is extracted according to user requirements and these RDF triples are defined in terms of patterns structures i.e., specifying the entities, their descriptions and the mapping from entities to description. After these two operations, we define a similarity measure $\sqcap$ over two descriptions which we generalize to the set of descriptions. After defining the similarity measure, we explain how an RDF-pattern concept lattice is built using this similarity measure. More generally, an organization of RDF triples is built, based on concept lattice, w.r.t background knowledge. Finally, this lattice is used for navigation and interactive exploration purposes.

A. From RDF Triples to RDF-Pattern Structures

Firstly, we represent RDF triples extracted by the SPARQL query in Listing 1 as entities and their descriptions. The pattern structures are given as $(G, (D, \sqcap), \delta)$. A subject in an RDF triple is mapped to an entity g in the set of entities G and predicate object pair (p:o) is mapped to a description $d \in D$. As the set of entities G represent the subjects in triples, we represent it as S. The descriptions D are termed as descriptions and are denoted as D_s.

The mapping of entities to description $\delta : S \rightarrow D_s$ is given as follows: let $s_i \in S$ then $\delta(s_i) = \{d_{i1}, \ldots, d_{iq}\}$ where $i \in \{1, \ldots, n\}$ and $d_{ij} = \{p_j : \{o_1, o_2, \ldots, o_m\}\}$ where $j \in \{1, \ldots, q\}$. In the running scenario, we have $p_1 = dc : subject$ and $p_2 = dc : creator$ as shown in Table I. For p_1 the elements in the range can be compared with the help of reference schema and for p_2 the elements in the range are names of the author which can not be compared.

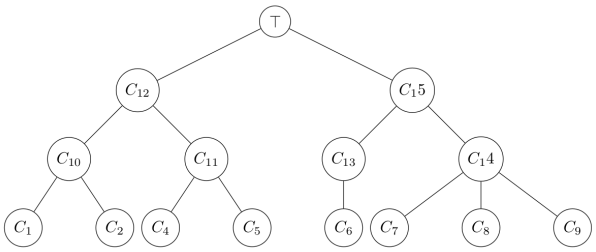


Fig. 4: A small part from ACM Computing Classification System.

Entities S	d_{i1}	d_{i2}
s_1	$(p_1 : \{C_1, C_2, C_7\})$	$(p_2 : \{o_{21}\})$
s_2	$(p_1 : \{C_6, C_8, C_9\})$	$(p_2 : \{o_{22}, o_{23}\})$
s_3	$(p_1 : \{C_4, C_5\})$	$(p_2 : \{o_{22}, o_{24}, o_{25}\})$
s_4	$(p_1 : \{C_4, C_7, C_8\})$	$(p_2 : \{o_{23}\})$
s_5	$(p_1 : \{C_8, C_9\})$	$(p_2 : \{o_{22}, o_{23}, o_{25}\})$

TABLE V: RDF Triples as entities S and semantic descriptions D_s.

After this organization a suitable RDF Schema is selected based on what user needs and the fact that it contains the objects in the triples. RDF Schema contains many constructs such as property, sub-property etc. along with `rdfs: subClassOf` information but in this work we use the RDF Schema predicates such as `rdfs:subClassOf`, `skos:broader` which lead to a tree structure in an RDF graph. An RDF Schema from ACM Computing Classification System is shown in Figure 4 for the set of objects connected to predicate p_1. While the objects connected to the second predicate p_2 do not have any associated schema. The circles represent classes and the lines between these circles represent predicate `rdfs:subClassOf/skos:broader`. Each object is replaced with their classes i.e., $C_1(o_{11})$ meaning that o_{11} is an instance of class C_1. Then, the description $\{(p_1 : \{o_{11}, o_{12}, \dots\})\}$ is given as $\{(p_1 : \{C_1, C_2, \dots\})$ This replacement is only performed for the description for which there is some RDF Schema present. The triples $t1, t2, t3$ in Table II are represented as entities and descriptions where the entity s_1 has the description $\delta(s_1) = \{d_{11}, d_{12}\}$ where $d_{11} = p_1 : \{C_1, C_2, C_4, C_7\}$ and $d_{12} = p_2 : \{o_{21}, o_{22}\}$. Table V shows a final representation of the RDF triples after replacing the objects with their corresponding classes (if reference schema is available) as entity descriptions.

B. Similarity Operation Over Classes

The similarity operation between two different classes is computed w.r.t. the Least Common Subsumer (LCS) of two classes. Definition 4 is the definition of Least Common Subsumer in a partially ordered set. We denote $\sqsubseteq$ as $\preceq$ to avoid confusion between the concept lattice subsumption order.

Definition 4 (Least Common Subsumer): Given a partially ordered set $(S, \preceq)$, a least common subsumer E of two classes C and D (lcs(C,D) for short) in a partially ordered set is a class such that $C \preceq E$ and $D \preceq E$ and E is least i.e., if there is a class E' such that $C \preceq E'$ and $D \preceq E'$ then $E \preceq E'$.

Given a reference schema which in our case is a tree, two elements whose LCS is $\top$ are considered as non-similar. Now we are in the reference schema, we consider classes (these are classes of the objects in triples). We want to compute the similarity of descriptions $p_i : X$ and $p_j : Y$. First, we compute the similarity with the following constraints:

- *Case 1:* $p_i = p_j$.
- *Case 2:* we consider X and Y be the set of objects such that $X = \{o_i\}$, $Y = \{o_j\}$ where $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$. Here we consider the case when there is no reference schema for the elements in X and Y. If $o_i = o_j$ then $X \cap Y = \{o_i\}$ otherwise $\varnothing$.

  For instance, in Table V we have d_{i2} for which there is no reference schema available. Lets choose two sets of objects then accordingly we have, $p_2 : \{\{o_{22}, o_{23}\} \cap \{o_{22}, o_{23}, o_{25}\}\} = p_2 : \{o_{22}, o_{23}\}$.
- *Case 3:* In this case, let $X = \{C\}$ and $Y = \{D\}$ and the elements of X and Y are in the reference schema, we consider the classes of the elements and then the LCS of these elements. More formally, it can be defined as follows: Let us take two descriptions $c = p_1 : C$ and $d = p_1 : D$ then:

$$\begin{aligned} c \sqsubseteq d &= p_1 : C \sqsubseteq p_1 : D \\ \Leftrightarrow p_1 : C \sqcap p_1 : D &= p_1 : C \\ \Leftrightarrow p_1 : lcs(C, D) &= p_1 : C \\ \Leftrightarrow D \preceq C &\quad in\ the\ reference\ schema \end{aligned}$$

  This is the fundamental that should be verified by the similarity operator. The definition $\sqcap$ implies that specific class subsume the general class which is the super class. For example, for descriptions $D_s = \{C_2, C_4, C_5\}$ and the reference schema shown in Figure 4 the meet semi-lattice is given in Figure 5. Here, $C_{11} = C_4 \sqcap C_5$ and is subsumed by the class $C_{11} \sqsubseteq C_4$ and $C_{11} \sqsubseteq C_5$.

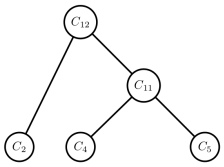


Fig. 5: $(D_s, \sqcap)$

- *Case 4:* In this case we compute the similarity between sets of classes. The similarity between the sets of classes is defined in the same way as Case 3. LCS is computed for every pair of classes in two different sets of description and then only the most specific classes are retained. Let C and D be two sets of classes then we have:

$$\begin{aligned} C \sqsubseteq D &= p_1 : C \sqsubseteq p_1 : D \\ \Leftrightarrow p_1 : C \sqcap p_1 : D &= p_1 : C \\ \Leftrightarrow p_1 : lcs(c_i, d_j) &= p_1 : c_i \\ where\ c_i \in C &\quad and \quad d_j \in D \end{aligned}$$

It means that $\forall c_1 \in C, \exists d_1 \in D, d_1 \preceq c_1$. The process is explained with the help of only one reference schema, multiple schemas can also be considered for several sets of objects.

For instance, lets choose two sets of classes $C = \{C_1, C_2, C_7\}$ and $D = \{C_4, C_7, C_8\}$ then the LCS between each pair will generate the set $\{C_{12}, C_7, C_{14}\}$. As only the specific elements are retained, the final set obtained is $E = \{C_{12}, C_7\}$. Accordingly, $E \sqsubseteq C$ and $E \sqsubseteq D$.

C. Building the RDF Pattern Concept Lattice

A representation of RDF triples as the set of descriptions given in Table V can be formalized as a pattern structure

$K\#ID$	**Extent**	**Intent**
$K\#1$	s_1, s_2, s_4, s_5	$(p_1 : \{C_{14}\})$
$K\#2$	s_1, s_3, s_4	$(p_1 : \{C_{12}\})$
$K\#3$	s_1, s_4	$(p_1 : \{C_7, C_{12}\})$
$K\#4$	s_2, s_4, s_5	$(p_1 : \{C_8\})$
$K\#5$	s_3, s_4	$(p_1 : \{C_4\})$
$K\#6$	s_1	$(p_1 : \{C_1, C_2, C_7\})$
$K\#8$	s_2, s_5	$(p_1 : \{C_8, C_9\})$
$K\#7$	s_4	$(p_1 : \{C_4, C_7, C_8\})$
$K\#9$	s_3	$(p_1 : \{C_4, C_5\})$
$K\#10$	s_2	$(p_1 : \{C_6, C_8, C_9\})$

TABLE VI: Details of *Pattern Concept lattice* in Figure 6.

$K\#ID$	**Extent**	**Intent**
$K\#1$	s_1, s_2, s_4, s_5	$(p_1 : \{C_{14}\})$, $(p_2 : \{\})$
$K\#2$	s_1, s_3, s_4	$(p_1 : \{C_{12}\})$, $(p_2 : \{\})$
$K\#3$	s_2, s_3, s_5	$(p_1 : \{\})$, $(p_2 : \{o_{22}\})$
$K\#4$	s_2, s_4, s_5	$(p_1 : \{C_8\})$, $(p_2 : \{o_{23}\})$
$K\#5$	s_1, s_4	$(p_1 : \{C_{12}, C_7\})$, $(p_2 : \{\})$
$K\#6$	s_3, s_4	$(p_1 : \{C_4\})$, $(p_2 : \{\})$
$K\#8$	s_4	$(p_1 : \{C_4, C_7, C_8\})$, $(p_2 : \{o_{23}\})$
$K\#7$	s_3, s_5	$(p_1 : \{\})$, $(p_2 : \{o_{22}, o_{25}\})$
$K\#9$	s_1	$(p_1 : \{C_1, C_2, C_7\})$, $(p_2 : \{o_{21}\})$
$K\#10$	s_3	$(p_1 : \{C_4, C_5\})$, $(p_2 : \{o_{22}, o_{24}, o_{25}\})$
$K\#11$	s_2, s_5	$(p_1 : \{C_8, C_9\})$, $(p_2 : \{o_{22}, o_{23}\})$
$K\#12$	s_2	$(p_1 : \{C_6, C_8, C_9\})$, $(p_2 : \{o_{22}, o_{23}\})$
$K\#13$	s_5	$(p_1 : \{C_8, C_9\})$, $(p_2 : \{o_{22}, o_{23}, o_{25}\})$

TABLE VII: Details of *Pattern Concept lattice* in Figure 7.

$(S, (D_s, \sqcap), \delta)$. When $A \subseteq S$ is a set of entities and $d \in (D_s, \sqcap)$ is a semantic description containing classes and objects then $A^{\square}$ returns a set of common objects (if reference schema is absent) present in the descriptions of each subject in A and $A^{\diamond}$ returns LCS of the classes (when reference schema is available) in the description of A. On the other hand, $d^{\square\diamond}$ returns the set of subjects which are described by the objects/classes of objects included in d.

Firstly, we explain how to build the lattice for the descriptions having the reference schema. So we build the lattice only with the descriptions d_{i1} in Table V. The similarity between two subjects can be given as:

$$\begin{aligned}
\{s_1, s_3\}^{\diamond} &= \prod_{s \in \{s_1, s_3\}} \delta(s) \\
&= \delta(s_1) \sqcap \delta(s_3) \\
&= \langle (p_1 : \{C_1, C_2, C_7\}) \sqcap (p_1 : \{C_4, C_5\}) \rangle \\
&= \langle (p_1 : \{lcs(C_1, C_4), lcs(C_1, C_5), \dots\}) \rangle \\
&= \langle (p_1 : \{C_{12}\}) \rangle \\
\langle (p_1 : \{C_{12}\}) \rangle^{\diamond} &= \{s \in S | \langle (p_1 : \{C_{12}\}) \rangle \sqsubseteq \delta(s)\} \\
&= \{s_1, s_3, s_4\}
\end{aligned}$$

The pair $(A, d) = (\{s_1, s_3, s_4\}, \langle (p_1 : \{C_{12}\}) \rangle)$ is a pattern concept ($K\#2$ in Table VI) meaning that $A^{\diamond} = d$ and $d^{\diamond} = A$. A set of all pattern concept creates a pattern concept lattice shown in Figure 6. The subsumption order $\sqsubseteq$ between two patterns in pattern concepts (A_1, d_1) and (A_2, d_2) is given as follows: $(A_2, d_2) \sqsubseteq (A_1, d_1) \iff \forall c_2 \in d_2, \exists c_1 \in d_1, c_1 \preceq c_2$ *(in the reference schema)*. Similarly, it can be seen that $(A_2, d_2) \sqsubseteq (A_1, d_1) \iff (A_1, d_1) \sqcap (A_2, d_2) = (A_2, d_2)$.

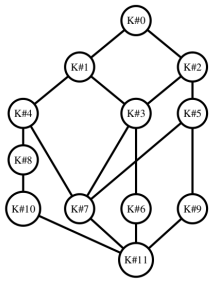


Fig. 6: Pattern Concept lattice for DBLP and ACCS.

Now we consider an example with all the cases described in the previous section. From Table V we have $\delta(s_1) = (p_1 : \{C_1, C_2, C_7\}), (p_2 : \{o_{21}\})$ and $\delta(s_3) = (p_1 : \{C_4, C_5\}), (p_2 : \{o_{22}, o_{24}, o_{25}\})$, where p stands for a predicate, C stands for a class and o stands for an object having no class.

In this case, the first description $(p_1 : \{C_1, C_2, C_7\})$ has an associated reference schema, while the reference schema for second description $(p_2 : \{o_{21}\})$ is absent. Then the similarity between these two subjects is given as follows:

$$\begin{aligned}
\{s_1, s_3\}^{\square\diamond} &= \prod_{s \in \{s_1, s_3\}} \delta(s) \\
&= \delta(s_1) \sqcap \delta(s_3) \\
&= \langle (p_1 : \{C_1, C_2, C_7\})(p_2 : \{o_{21}\}) \\
&\quad \sqcap (p_1 : \{C_4, C_5\})(p_2 : \{o_{22}, o_{24}, o_{25}\}) \rangle \\
&= \langle (p_1 : \{C_1, C_2, C_7\}) \sqcap (p_1 : \{C_4, C_5\}), \\
&\quad (p_2 : \{o_{21}\}) \sqcap (p_2 : \{o_{22}, o_{24}, o_{25}\}) \rangle \\
&= \langle (p_1 : \{C_{12}\})(p_2 : \{\}) \rangle \\
\langle (p_1 : \{C_{12}\})(p_2 : \{\}) \rangle^{\square\diamond} &= \{s \in S | \langle (p_1 : \{C_{12}\})(p_2 : \{\}) \rangle \sqsubseteq \delta(s)\} \\
&= \{s_1, s_3, s_4\}
\end{aligned}$$

The pair $(A, d) = (\{s_1, s_3, s_4\}, \langle (p_1 : \{C_{12}\})(p_2 : \{\}) \rangle)$ is a pattern concept ($K\#2$ in Table VII) meaning that $A^{\square\diamond} = d$ and $d^{\square\diamond} = A$. A set of all pattern concept creates a pattern concept lattice shown in Figure 7 and is termed as *navigation space*. Further details about the algorithm used for computing Least Common Subsumer are discussed in [11]. It discusses how pattern structures is adapted for dealing with structured attribute sets.

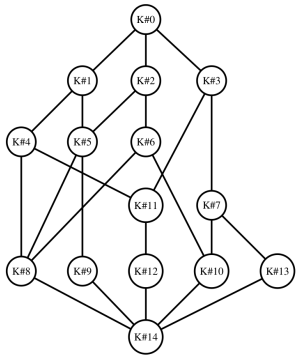


Fig. 7: Pattern Concept lattice for DBLP and ACCS with and without reference schema.

V. Navigation and Interactive Exploration over Pattern Concept Lattice

A. Navigation Operations:

Several navigation operations can be applied over the *navigation space* for obtaining precise information by navigation [12]. Here, each concept contains a group of subjects connected to the classes of the objects. The most general concepts near the top of the pattern concept lattice contain more number of subjects (entities) and lesser number of classes (description) meaning that the descriptions are very general. As the lattice is navigated downward more specific descriptions exist with lesser number of subjects.

Let us consider the scenario discussed in section II. We consider the navigation space shown in Figure 7. If user wants to retrieve the scientific papers on some specific topic such as World Wide Web, she would easily locate the concept containing only the papers about this topic i.e., $K\#2$. The retrieved papers will be s_1, s_3, s_4. For narrowing down her papers which are related to World Wide Web and Question Answering, the lattice can be navigated downwards to obtain $K\#5$ which contain the two papers s_1, s_4. Now the user has the choice for further narrowing down w.r.t. more specified classes such as papers on Question Answering and recommender systems over RDF i.e., $K\#8$.

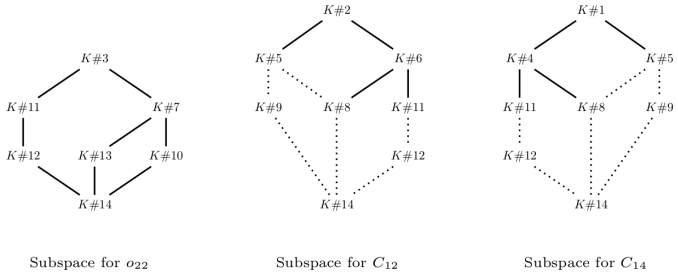


Fig. 8: Sublattices from Figure 7

The obtained concept lattice keeps sub-spaces (sub-lattices) which are interpreted as the space related to some topic or author. Figure 8 shows three example subspaces (Note that these subspaces are not extracted, these are just drawn separately to give a clear look into what the navigation space contains).

Figure 8 shows the sub-space related to an author o_{22} which represents the community of the authors who work with this author. It contains 5 concepts $K\#3, K\#7, K\#10, K\#11, K\#12, K\#13$. $K\#3$ contains all the papers published by the author o_{22}, then this sub lattice can be navigated downwards to obtain specific concepts such as $K\#7$ and $K\#11$. These two concepts show co-authors of o_22 i.e., o_{23} and o_{25}. Based on the support of the concept i.e., It can also be seen that these two concepts represent the community of authors that often work together. The author o_{22} has stronger communication with the authors in this concept. However, when we move downwards in the lattice the communication becomes weaker as the number of papers published together decrease. Based on the concepts $K\#7$ and $K\#10$, recommendation can be given to author o_{25} to work with author o_{23}. This way this navigation space can be used for recommending the social collaborations of the authors working in the similar field. Navigating from $K\#3$ to $K\#11$ the papers of the author o_{22} are filtered with respect to the topic of the paper.

Now let us consider two more sub-spaces w.r.t. the topic of the paper. Figure 8 provide the subspaces w.r.t. the topics World Wide Web and Retrieval tasks and goals respectively. Both the subspaces can navigated from top to bottom to obtain papers on general topics and can be navigated down to get a smaller list of papers based on sub-topics or the list of authors. The important point to notice is that the dotted part in both the subspaces represent the common space of the two topics C_{12} and C_{14} meaning that this common sub-sub-space keep the papers which are common to both the topics. It also keeps the combination of different classes which benefits the user while finding the subjects which related to more than one object or class of object simultaneously. The cases indicated in the navigation and interpretation scenario above is very generic and can be used in any domain.

B. Interactive Data Exploration over Navigation Space:

The navigation space obtained using the RDF-Pattern Structures serves as an interactive exploration space for the user. The user can perform interactive exploration while navigating from any of the dimensions (in our case authors and topic). Each concept act as a sample for exploration. These concepts keep classes of RDF triples as described before. Now the user can mark these samples as irrelevant. If the dimension explored by the user does not have a reference schema then all the sub-concepts of the selected concept are marked irrelevant automatically (i.e., author dimension) because the descriptions in this concept are inherited by its sub-concepts as described ins section III-C. However, if the dimension is organized w.r.t. a reference schema then all the subclasses of the class in the marked concept are also marked irrelevant. So, all the sub-concepts of the marked concept containing the classes as well as its subclasses are marked irrelevant and are hidden from the user.

Suppose that the user is not interested in the papers on the topic of Semantic Web Description Languages i.e., C_{11}. The navigation space shown in Figure 7 works as the exploration space, while navigating the user sees $K\#2$ which contains papers on World Wide Web i.e., C_{12}, she will mark this concept as irrelevant then the sub-concepts $K\#5, K\#6, K\#8, K\#9, K\#10$ are also marked as irrelevant because these concepts either keep the class C_{12} or a sub-class of C_{12} i.e., C_1, C_2, C_4. Consider that the user is exploring the navigation space w.r.t author dimension and she marks $K\#3$ as irrelevant because of the author o_{22} then the sub-lattice obtained by following the links from $K\#3$ until the bottom will be marked as irrelevant i.e., the concepts $K\#7, K\#10, K\#11, K\#12, K\#13$.

VI. Experimentation

In this section we discuss the experimental results for the RDF Pattern Structures. The proposed algorithm was coded in C++ and the experiments were performed using 3GB RAM on Ubuntu version 12.04.

A. DBLP

The dataset used for experimentation was DBLP which keeps bibliographic information about millions of journals, conferences and authors. DBLP is converted to RDF and made available with the help of D2R server[6]. D2R server [7] provides a mapping from SQL database schema to RDF triples. However, in the current experiment the triple store used is the RDF data dump for DBLP is made available at RDF-HDT[7] [13]. RDF-HDT (Header, Dictionary, Triples) is a compact data structure for RDF data which provides efficient storage by compressing big datasets. It also provides search and browse operations without prior decompression. For the experimentation two subsets of datasets were extracted from DBLP. First includes all the papers on Artificial Intelligence (AI) and the second dataset includes all the papers on Machine Learning (ML). The reference schema used for this purpose is ACM Computing Classification System (ACCS) which is available on-line in several formats. The RDF Format used for ACCS uses SKOS[8] vocabulary, an application of RDF, to define the background knowledge about the topics of the papers. For conducting the experiments, titles were considered as entities and keywords and authors were kept as descriptions. ACCS was used as a reference schema for keywords and authors did not have any reference schema.

Datasets	No. of Triples	No. of Subjects	No. of Objects
AI	31045	9986	31140
ML	18141	5571	17633

TABLE VIII: Statistics of two datasets.

After extracting the datasets navigation spaces are built on each of the data sets using RDF Pattern Structures. The statistics regarding both the data sets are shown in Table VIII. The number of triples extracted for AI dataset are 31045 and for ML are 18141. Figure 9(a) depicts the size of the navigation spaces created for AI and ML data sets. It can be seen that the size of navigation space is suitable for exploration purposes, however the interactive data exploration will further reduce its size when the user will mark the concepts as irrelevant. For example, in AI if the user is not interested in the papers about robotics then she can choose the general class as irrelevant. This will further decrease the navigation space of the user. Figure 9(b) illustrates the runtime for creating the navigation space. In the current experiments we extracted the RDF data using SPARQL queries. These SPARQL queries specify the initial user and task- specific requirements and extract only small subsets of data interesting for the user. Following this line it is safe to assume that our approach is well adapted to exploratory data mining as we are using small subsets of data for exploration and visualization using a visualization tool discussed in next section. Finally, the main focus of our approach is the qualitative analysis of the data and allow user with interaction and exploration.

B. Visualization

Another experiment was performed on the papers published by a Data Mining Team in a research lab. For this purpose all the papers from 2010-2014 published in international journals and conferences were selected. The papers chosen for this purpose were all in English Language. A pattern concept lattice (navigation space) was built using the paper titles, their keywords and authors. The results were visualized using the tool RV-Xplorer (Rdf View eXplorer) [14]. It visualizes and allows interaction over the view defined over RDF graph through SPARQL queries by classifying SPARQL query answers in the form of a concept lattice. A dedicated web page to visualize and interact with the navigation space is available *http://webloria.loria.fr/∼alammehw/rdfps/#/.*

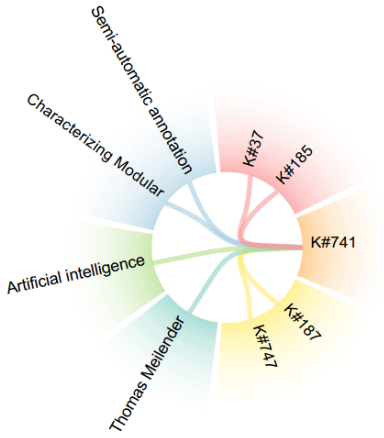


Fig. 10: Concept

Figure 10 shows one concept from the graphical user interface for visualizing the resulting navigation space. The circle represents the selected concept which is the top of the concept lattice by default. It displays the contents of the selected concept i.e., the extent, intents, parent concepts and children concepts. The pink and yellow part in the selected concept $(K\#741)$ show the parent $(K\#37, K\#185)$ and child concepts $(K\#187, K\#747)$ respectively. The green and cyan part show two different types of intent i.e., topics and authors. The blue part shows the extent of the concept i.e., the group of papers sharing some authors and topic. Now let us

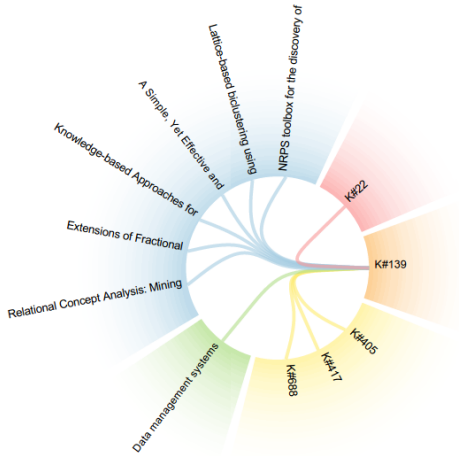
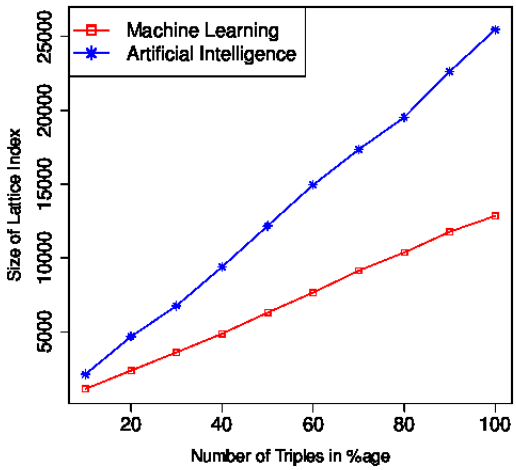


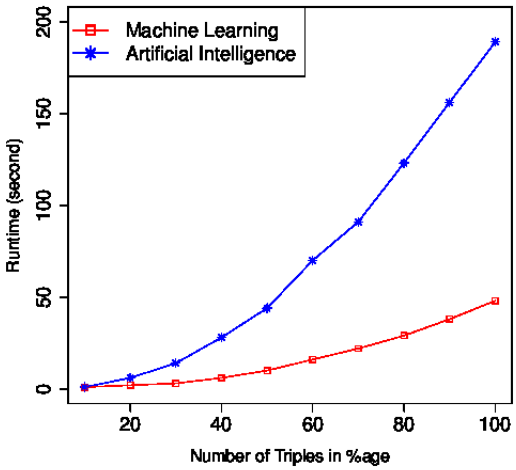
Fig. 11: Papers on Database Management System

consider that the user chooses a concept keeping the papers

[6]http://dblp.l3s.de/d2r/
[7]http://www.rdfhdt.org/datasets/
[8]http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102/



(a) Variation in the size of Navigation Space.



(b) Runtime for Creating the Navigation Space.

Fig. 9: Experimental Results.

about Database Management System. Figure 11 shows the selected concept (K#139). This visualization allows the user to navigate upwards and downwards in the navigation space to access specific as well as general information as discussed in section V-A. If the user wants to navigate downwards in the lattice she should select one of the concepts in the yellow part of the selected concept which keep the sub-concepts (K#405, K#417, K#688). As the sub-concepts can be large in number we display what kind of papers the sub-concept of the lattice contains on mouse hover (see Figure 12). This phenomena guides the user in deciding which path to take while navigating. Now the user wants to narrow down papers about Relational database query languages, she will click on K#688 in the yellow part of the selected concept and then K#688 becomes the selected concept. This navigation is referred to as *level-wise navigation*.

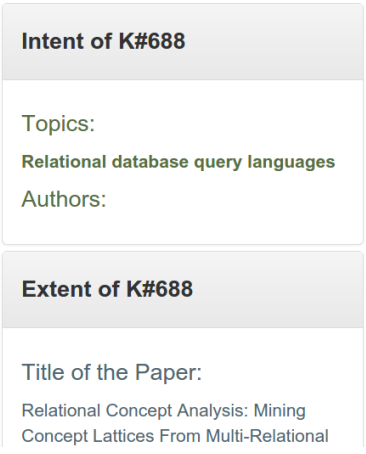


Fig. 12: Details of Subconcept K#688

The navigation space on the left hand side of the visualization shows the complete lattice to track the position of the user. The selected concept is highlighted in red color. Now if the user is on the current concept and she is interested in the papers in this concept and wants to find other papers similarly to this paper then on mouse hover on this paper title. This will highlight all the concepts where this paper is present and the user can then directly navigate to the concept of interest. Toolkit also allows direct navigation from one concept to another without going from one hop to another. Such kind of navigation is referred to as *direct navigation*.

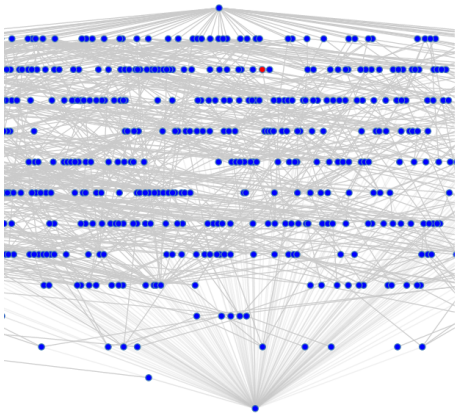


Fig. 13: Selected Sub-concept

Finally, it helps in decreasing the navigation space of the user by enabling her to focus on only the interesting parts in the navigation space and hide the rest of the lattice (see section V-B). If the user right-clicks on a concept in the navigation space, it is marked as irrelevant to the user and is hidden from the user. Once marked irrelevant the hidden part can not be accessed unless marked relevant.

VII. Related Work

There have been several studies which apply FCA to RDF data but to-date this is the first attempt to deal with RDF graph and pattern structures. In [15], the author focuses on allowing conceptual navigation to RDF graphs, where each concept is accessed through SPARQL-like queries. However, in our case several RDF resources can be navigated from one platform based on user requirements. Hiding the non-interesting part of the concept lattice is the feature very unique to our approach. Moreover, [16] introduces ontological pattern structures for enriching raw data with $\mathcal{EL}$ ontologies. But both the approaches consider only one resource at a time, hence not targeting the problem of decentralization. As a contrast to [16], RDF-Pattern Structures provide navigation space over RDF graphs as well as schema level information from several resources allowing user to access information from one platform.

In [17], the authors focus on clustering the SPARQL query answers based on some background knowledge and provide access to these clusters using a tree structure. As a contrast our approach can directly deal with RDF graphs. Moreover, several reference schemas related to different types of objects can be used through RDF Pattern Structures. Also, our approach can handle the objects which do not have any existing reference schema.

VIII. Conclusion

This paper proposes a new approach for navigating semantic web data and targets the capabilities of Pattern Structures to deal with RDF data. It provides navigation space over RDF data by organizing RDF triples with respect to reference schema with the help of RDF Pattern Structures. To deal with such an organization, this paper proposes a new similarity measure. The pattern concepts in the concept lattice are considered as clusters of RDF triples used for information retrieval purposes over RDF data. The proposed framework is very general and can be applied to any RDF data set having heterogeneity i.e., some of the objects containing the reference schema and some of the objects containing no schema. One such application is the RDF triples contained in Drugbank where the objects are considered as drugs and the attributes are side effects, their categories and proteins. There are reference schemas regarding the side effects and categories of the such as MeSH and MedDRA but there is no reference schema defined for protein targeted by these drugs. One of the future directions is to use complete RDF Schema i.e., all the predicates instead of using only those predicates which are defining taxonomical structure. In such a case, the similarity measure can be modified to path between two classes instead of only computing the Least Common Subsumer.

References

[1] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data - the story so far," *Int. J. Semantic Web Inf. Syst.*, vol. 5, no. 3, pp. 1–22, 2009.

[2] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello, "Sindice.com: a document-oriented lookup index for open linked data," *IJMSO*, vol. 3, no. 1, pp. 37–52, 2008. [Online]. Available: http://dx.doi.org/10.1504/IJMSO.2008.021204

[3] Y. Sure, S. Bloehdorn, P. Haase, J. Hartmann, and D. Oberle, "The swrc ontology - semantic web for research communities." in *EPIA*, ser. Lecture Notes in Computer Science, vol. 3808. Springer, 2005, pp. 218–231. [Online]. Available: http://dblp.uni-trier.de/db/conf/epia/epia2005.html#SureBHHO05

[4] M. van Leeuwen, "Interactive data exploration using pattern mining," in *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics - State-of-the-Art and Future Challenges*, ser. Lecture Notes in Computer Science, A. Holzinger and I. Jurisica, Eds. Springer, 2014, vol. 8401, pp. 169–182. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-43968-5_9

[5] B. Ganter and S. O. Kuznetsov, "Pattern structures and their projections," in *ICCS*, ser. Lecture Notes in Computer Science, H. S. Delugach and G. Stumme, Eds., vol. 2120. Springer, 2001, pp. 129–142.

[6] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*. Berlin/Heidelberg: Springer, 1999.

[7] C. Bizer and R. Cyganiak, "D2r server - publishing relational databases on the semantic web," Poster at the 5th International Semantic Web Conference, 2006. [Online]. Available: http://www4.wiwiss.fu-berlin.de/bizer/pub/Bizer-Cyganiak-D2R-Server-ISWC2006.pdf

[8] C. Carpineto and G. Romano, *Concept data analysis - theory and applications.* Wiley, 2005.

[9] M. Kaytoue, S. O. Kuznetsov, and A. Napoli, "Revisiting numerical pattern mining with formal concept analysis," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence.*, 2011, pp. 1342–1347. [Online]. Available: http://ijcai.org/papers11/Papers/IJCAI11-227.pdf

[10] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 697–706. [Online]. Available: http://doi.acm.org/10.1145/1242572.1242667

[11] M. Alam, A. Buzmakov, A. Napoli, and A. Sailanbayev, "Revisiting pattern structures for structured attribute sets," in *Proceedings of the 12th International Conference on Concept Lattices and Their Applications, Clermont-Ferrand, France, October 13-16*, 2015.

[12] V. Codocedo, I. Lykourentzou, and A. Napoli, "A semantic approach to concept lattice-based information retrieval," *Ann. Math. Artif. Intell.*, vol. 72, no. 1-2, pp. 169–195, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10472-014-9403-0

[13] J. D. Fernández, M. A. Martínez-Prieto, C. Gutiérrez, A. Polleres, and M. Arias, "Binary RDF representation for publication and exchange (HDT)," *J. Web Sem.*, vol. 19, pp. 22–41, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.websem.2013.01.002

[14] M. Alam, M. Osmuk, and A. Napoli, "RV-Xplorer: A way to navigate lattice-based views over RDF graphs," in *Proceedings of the 12th International Conference on Concept Lattices and Their Applications, Clermont-Ferrand, France, October 13-16*, 2015.

[15] S. Ferré, "Conceptual navigation in RDF graphs with sparql-like queries," in *Formal Concept Analysis, 8th International Conference, ICFCA 2010, Agadir, Morocco, March 15-18, 2010. Proceedings*, 2010, pp. 193–208. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-11928-6_14

[16] A. Coulet, F. Domenach, M. Kaytoue, and A. Napoli, "Using pattern structures for analyzing ontology-based annotations of biomedical data," in *11th International Conference on Formal Concept Analysis,*, 2013. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38317-5_5

[17] C. d'Amato, N. Fanizzi, and A. Lawrynowicz, "Categorize by: Deductive aggregation of semantic web query results," in *ESWC (1)*, ser. Lecture Notes in Computer Science, L. Aroyo, G. Antoniou, E. Hyvönen, A. ten Teije, H. Stuckenschmidt, L. Cabral, and T. Tudorache, Eds., vol. 6088. Springer, 2010, pp. 91–105.