RV-Xplorer: A Way to Navigate Lattice-Based Views over RDF Graphs

Mehwish Alam, Amedeo Napoli, and Matthieu Osmuk

LORIA (CNRS – Inria Nancy Grand Est – Université de Lorraine)
BP 239, Vandoeuvre-lès-Nancy, F-54506, France
{mehwish.alam,amedeo.napoli,matthieu.osmuk@loria.fr}

Abstract. More and more data are being published in the form of machine readable RDF graphs over Linked Open Data (LOD) Cloud accessible through SPARQL queries. This study provides interactive navigation of RDF graphs obtained by SPARQL queries using Formal Concept Analysis. With the help of this `View By` clause a concept lattice is created as an answer to the SPARQL query which can then be visualized and navigated using RV-Xplorer (Rdf View eXplorer). Accordingly, this paper discusses the support provided to the expert for answering certain questions through the navigation strategies provided by RV-Xplorer. Moreover, the paper also provides a comparison of existing state of the art approaches.

Keywords: RV-Xplorer, Lattice Navigation, SPARQL Query Views, Formal Concept Analysis

1 Introduction

Recently, Web Data is turning into "Web of Data" which contains the meta data about the web documents present in HTML and textual format. The goal behind this "Web of Data" is to make already existing data to be usable by not only human agents but also by machine agents. With the effort of Semantic Web community, an emerging source of meta data is published on-line called as Linked Open Data (LOD) in the form of RDF data graphs. There has been a huge explosion in LOD in recent past and is still growing. Up until 2014, LOD contains billions of triples. SPARQL[1] is the standard query language for accessing RDF graphs. It integrates several resources to generate the required answers For instance, queries such as *What are the movements of the artists displayed in Musee du Louvre?* can not be answered by standard search engines. Nowadays, Google has introduced a way of answering questions directly such as currency conversion, calculator etc. but such queries are answered based on most frequent queries posed by the experts.

When an expert poses a query to a search engine too many results are retrieved for the expert to navigate through, which may be cumbersome when a

---

[1] http://www.w3.org/TR/rdf-sparql-query/

expert has to go through a number of links to find the interesting ones, hence leading to the problem of *information overload* [3]. Same is the case with the answers obtained by SPARQL query with the `SELECT` [8]. Even if there are hundreds of answers, it becomes harder for the expert to find the interesting patterns. The current study is a continuation of Lattice-Based View Access (LBVA) [1] which provides a view over RDF graphs through SPARQL queries to give complete understanding of a part of RDF graph that expert wants to analyze with the help of Formal Concept Analysis. LBVA takes the SPARQL query and returns a concept lattice called as view instead of the results of the SPARQL query. These views created by LBVA are machine as well as human processable. Accordingly, RV-Xplorer (Rdf View eXplorer) exploits the powerful mathematical structure of these concept lattices thus making it interpretable by human. It also allows human agents to interact with the concept lattice and perform navigation. The expert can answer various questions while navigating the concept lattice. RV-Xplorer provides several ways to guide the expert during this navigation process.

This paper is structured as follows: section 2 gives the motivating example, section 3 introduces the required background knowledge to understand the rest of the paper. Section 4 details the elements of Graphical User Interface while section 5 and section 6 details the navigation operations as well as other functionalities supported by RV-Xplorer. Section 7 briefly discusses the related work. Finally, section 8 concludes the paper and discusses the future work.

2   Motivating Example

Consider a scenario where an expert wants to pose following questions based on articles published in conferences or journals from a team working on data mining. In the current study, we extract the papers published in "Orpailleur Team" in LORIA, Nancy, France. Following are the questions in which an expert may be interested in:

– What are the main research topics in the team and the key researchers w.r.t. these topics, for example, researchers involved in most of the papers in a prominent topic?
– What is the major area of the research of the leader of the team and various key persons?
– Can the diversity of the team leader and key persons be detected?
– Given a paper is it possible to retrieve similar papers published in the team?
– Who are the groups of persons working together?
– What are the research tendencies and possibly the forthcoming and new research topics (for example, single and recent topics which are not in the continuation of the present topics)?

Such kind of questions can not be answered by Google. In this paper we want to answer such kind of questions through lattice navigation supported by RV-Xplorer which is built from an initial query and then is explored by the expert according to her preferences.

3   Preliminaries

Linked Open Data: Linked Open Data (LOD) [2] is the way of publishing structured data in the form of RDF graphs. Given a set of URIs $\mathbf{U}$, blank nodes $\mathbf{B}$ and literals $\mathbf{L}$, an RDF triple is represented as $t = (s, p, o) \in (\mathbf{U} \cup \mathbf{B}) \times \mathbf{U} \times (\mathbf{U} \cup \mathbf{B} \cup \mathbf{L})$, where s is a subject, p is a predicate and o is an object. A finite set of RDF triples is called as RDF Graph $\mathcal{G}$ such that $\mathcal{G} = (V, E)$, where V is a set of vertices and E is a set of labeled edges. Each pair of vertices connected through a labeled edge keeps the information of a statement. Each statement is represented as $\langle subject, predicate, object \rangle$ referred to as an RDF Triple. V includes *subject* and *object* while E includes the *predicate*.

SPARQL: A standard query language for RDF graphs is SPARQL[2] 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). 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 the answers of the query should be constructed.

Let us consider a query from the scenario in section 2, Q = *Who is the team leader of the data mining team in loria.* For answering such questions consider an RDF resource containing all the papers ever published in the data mining team. With the help of SPARQL query the papers published in the last 5 years in English language can be extracted. The SPARQL representation of the query Q is shown in listing 1.1. Lines 1, 2 keep the information about the prefixes used in the rest of the query. Line 5, 6 and 7 retrieve all the papers with their authors and keywords. Line 8 and 9 retrieve the publication year of the paper and filter according to the condition.

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

3 SELECT distinct ?title ?keywords ?author
4 where {
5 ?paper dc:creator ?author .
6 ?paper dc:subject ?keywords .
7 ?paper dc:title ?title .
8 ?paper dcterms:issued ?publicationYear
9  FILTER(xsd:date(?publicationYear) >= '2011-01-01'^^xsd:date) }
```

Listing 1.1: SPARQL for extracting triples.

Lattice-Based View Access: Lattice-Based View Access [1], allows the classification of SPARQL query results into a concept lattice, referred to as a *view*, for data analysis, navigation, knowledge discovery and information retrieval purposes. It introduces a new clause `VIEW BY` which enhances the functionality of already existing `GROUP BY` clause in SPARQL query by adding sophisticated classification and Knowledge Discovery aspects.

---

[2] http://www.w3.org/TR/rdf-sparql-query/

The variable appearing in the `VIEW BY` clause of the SPARQL query is referred to as object variable[3] The rest of the variables are the attribute variables. Then the answer tuples obtained by the query are processed based on object and the attribute variables. The values obtained for the object variable are mapped to the objects in the formal context $\mathcal{K} = (G, M, I)$ and the answers obtained for attribute variables are mapped to the attributes in the context. Consider the query given in listing 1.1 with classification capabilities i.e., containing the clause `VIEW BY ?title` then the set of variables in the SELECT clause can be given as $V = \{$ *?title, ?keyword, ?author*$\}$. The object variable will be *?title* and attribute variable will be *?keyword* and *?author*. After applying LBVA, the objects contain the titles of the paper and the attributes are the set of keywords and authors in the context. From this context, the concept lattice is built which is referred to as a *Lattice-Based View*.

LBVA is oriented towards the classification of SPARQL queries, but we can interpret the present research activity at a more general level, the classification of LOD. Accordingly, what is proposed in the paper is a tool for navigating a classification of LOD.

4   The RV-Xplorer

RV-Xplorer (Rdf View eXplorer) is a tool for navigating concept lattices generated by the answers of SPARQL queries over part of RDF graphs using Lattice-Based View Access. Accordingly, this tool provides navigation to the expert over the classification of SPARQL query answers for analyzing the data, finding hidden regularities and answering several questions. On each navigation step it guides the expert in decision making and performing selection to avoid unnecessary selections. It also allows the user to change her point of view while navigating i.e., navigation by extent. Moreover, it also allows the expert to only focus on the specific and interesting part of the concept lattice by allowing her to hide the part of lattice which is not interesting for her.

RV-Xplorer is a web-based tool for building concept lattices. On the client side it uses D3.js which stands for Data-Driven Documents and is based on Javascript for developing interactive data visualizations in modern web browsers. It also uses model-view-controller (MVC) which separates presentation, data and logical components. On the server side we use PHP and MySQL for computing and storing the data. Generally, data can be a graph or pattern generated by pattern mining algorithms etc. Currently, this tool is not publicly available.

Figure 1 shows the overall interface of RV-Xplorer (Rdf View eXplorer) which consists of three parts: (1) the middle part is called *local view* which shows detailed description of the selected concept allowing interaction, navigation and level-wise navigation, (2) the left panel is referred to as *Spy* showing the global view of the concept lattice and (3) the lower left is the summarization index for guiding the expert in making decision about which node to choose in the next

---

[3] The object here refers to the object in FCA.

level by showing the statistics of the next level. For the running scenario, the concept lattice is also available on-line[4].

4.1   Local View

Each selected node in the concept lattice is shown in the middle part of the interface displaying complete information. Let c be the selected concept such that $c \in C$ where C is the set of concepts in the complete lattice $\mathcal{L} = (C, \leq)$ then a local view shows the complete information about this concept i.e., the extent, intent and the links to the super-concept and the sub-concepts. The set of super and sub-concepts are linked to the selected node where each link represents the partially ordered relation $\leq$. By default, the top node is the selected node and is shown in *local view*.

Figure 1 (below) shows the selected concept, the orange part defines the label of the selected node which is the entry point for the concept, the pink and yellow parts give the labels of the super-concepts and sub-concepts connected to the selected concept respectively. The green and blue part give the information about the intent and the extent respectively.

4.2   Spy

A global view in left panel shows the map of the complete lattice $\mathcal{L} = (C, \leq)$ for a particular SPARQL query over an RDF Graph. It tracks the position of the expert in the concept lattice and the path followed by the expert to reach the current concept. It also helps in several navigation tasks such as direct navigation, changing navigation space and navigation between point-of-views. All of these navigation modes are discussed in section 5.

4.3   Statistics about the next level

The statistics about the next level are computed with the help of a summarization index which depicts the information about the distribution of the objects in the extent of the selected concept in the linked sub-concepts i.e., concepts in the next level of the concept lattice. Let c_i be a concept in the next level where $i \in \{1, \ldots, n\}$ and n is the number of concepts in the next level. $ext(c_i)$ is the extent of the concept then $|ext(c_i)|$ is the size of the extent. Finally, the statistics about the next level are computed with the help of summarization index.

$$summarization\ index = \frac{|ext(c_i)|}{\sum_{j=\{1,\ldots,n\}} |ext(c_j)|} \times 100 \tag{1}$$

Here, $\sum_{j=\{1,\ldots,n\}} |ext(c_j)|$ is the sum of extent size of all the concepts in the next level. The sum of summarization index for all the sub-concept adds to 100%. In Figure 1, the percentages are represented in the form of a pie-chart

---

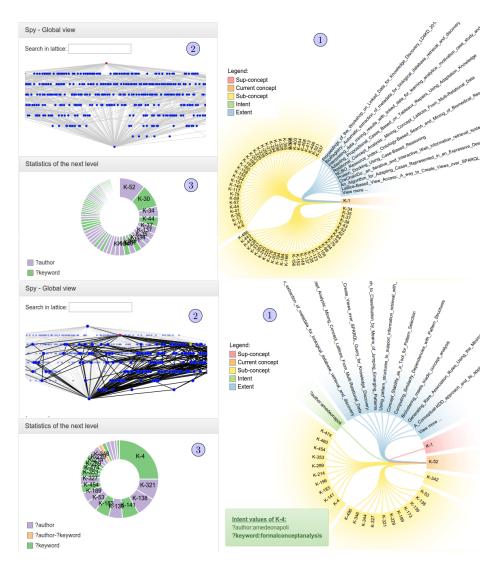[4] http://rv-xplorer.loria.fr/#/graph/orpailleur_paper/1/

Fig. 1: Figure above shows the basic interface of RV-Xplorer displaying the top concept. The Figure below shows the local view of $K\#52$, the concept containing all the papers authored by Amedeo Napoli.

which shows the distribution. The sub-concept containing the most elements in the extent has the highest percentage and hence has the biggest part in the pie chart.

5   Navigation Operations

In this section we detail some of the classical [4] as well as advanced navigation operations that are implemented in RV-Xplorer. Navigation can be done locally with a parallel operation which is shown globally through local and global views. Navigation operations allow the expert to locate particular pieces of information which helps in obtaining several answers of the expert questions as well as analysis of the data at hand. Initially, the selected concept is the top concept which contains all the objects.

5.1   Guided Downward (Drill down)/ Upward Navigation (Roll-up):

The local view provides expert with the drilling down operation which is achieved by selecting the sub-concepts given in yellow part of local view. RV-Xplorer guides the expert in drilling down the concept lattice by showing contents of the sub-concept to the expert before selecting the node on mouse over. Another added guidance provided to the expert is with the help of the summarization index which gives the statistics about the next level. This way the expert can avoid the attributes or the navigation path which may lead to uninteresting results. The local view also allows the expert to roll-up from the specific concept to the general concept. A super-concept can be selected following the link given in the view.

Consider the running scenario discussed in section 2 where the expert wants to know *who are researchers having main influences in the team?* by analyzing the publications of this particular team. Initially, the selected concept in the local view is the top concept (see Figure 1 (above)). Now it can be seen from the summarization index that most of the papers are contained in $K\#52$. On mouse over on $K\#52$ it shows that this concept keeps all the papers published by Amedeo Napoli. From here it can be safely concluded that Amedeo Napoli is the leader of the team. Similarly, several key team members can be identified on the same level such as supervisors etc. If the expert wants to view the papers published by Amedeo Napoli, a downward navigation is performed by selecting concept $K\#52$. With the help of the summarization index another question can be answered i.e., *what are the main research topics of these researchers?*. Again by consulting the index it can be seen that $K\#4$ keeps the largest percentage of papers published by Amedeo Napoli (see Figure 1 (below)) and the keyword in this concept is Formal Concept Analysis meaning that the main area of research of Amedeo Napoli is Formal Concept Analysis. However, there are many other areas of research on which he has worked, which shows the diversity of authors based on the area of research he has published in. Moreover, the sub-lattice connected to this concept keeps information about the community of authors

with who she publishes the most and about which topic and what variants of formal concept analysis. Now, if the expert wants to retrieve all the papers published by Amedeo Napoli then she can go back to $K\#52$.

5.2   Direct Navigation

The spy on the left part of the RV-Xplorer (see Figure 1) allows the expert for direct navigation. If an expert has navigated too deep in the view while performing multiple drill-down operations then the spy, which keeps track of the current position of the expert, shows all the paths from the selected concept to the top concept and allows the expert to directly jump from one concept to another linked concept without performing level-wise navigation. Unlike drill-down and roll-up, direct navigation allows the expert to skip two or more hops and select the more general or specific concept.

These three navigation modes are very common and are repeatedly discussed in many of the navigational tools built for concept lattice such as Camelis [11] and CREDO [5] which may or not may not be for a specific purpose. The main difference between RV-Xplorer and the two approaches and most of the navigational tools is that they use folder-tree display. As a contrast we manage to keep the original structure of a concept lattice. An added advantage of RV-Xplorer is that these navigation modes are guided at each step meaning that the interface shows the expert with what is contained in the next node as well as the statistics about the next level. This way the interface guides the expert in choosing the nodes interesting for her by reducing the chance of performing unnecessary navigation and backtracking to see the details unnecessarily.

5.3   Navigating Across Point-of-Views

The current interface allows the expert to toggle between points-of-view, i.e., at any point an expert can start exploring the lattice with respect to the objects (extent) in the concept lattice. Let c be the selected concept and the expert is interested in $g_1 \in ext(c)$ where $ext(c)$ is the extent of the selected concept. Then if the expert hovers her mouse over this extent in the local view, the Spy highlights all the concepts where this object is present along with the object concept of g_1 which is highlighted in red.

For instance, the selected concept contains keyword *data dependencies* in the intent and she is interested in the paper *Computing Similarity Dependencies with Pattern Structures* and she wants to retrieve all the related or similar papers then on mouse hover it highlights all the concepts containing this paper. Then she selects the concept highlighted in red i.e., the object concept of this paper. The right side of Figure 2 shows the highlighted object concept of *Computing Similarity Dependencies with Pattern Structures* in RV-Xplorer. After this concept is selected. The spy highlights all the paths from this concept until bottom and the top which actually is the sub-lattice associated to this paper. All the objects contained in the extent of the concepts in this sub-lattice are similar to

the paper at hand i.e., papers sharing some properties with the paper *Computing Similarity Dependencies with Pattern Structures.*

If we consider the folder-tree display as discussed in most of the navigational tools such as Camelis [11], CREDO [5] and CEM [7], such kind of navigation is not possible because it only allows navigation w.r.t. intent and extent is considered as the answers of the navigations. In case of RV-Xplorer, it is possible to obtain the sub-lattice related to a certain interesting object and this way the whole sub-lattice connected to the object concept of the object of interest can be navigated to retrieve similar objects i.e., sharing at least one attribute with the object of interest.

5.4   Altering Navigation Space

The navigation space can be changed when the selected concept is deep-down in the concept lattice without the effort to start the navigation all over again from the top concept. Let c be the selected concept such that m_1 and $m_2 \in int(c)$ (int(c) is the intent of the selected concept) and the expert has navigated downwards from the concept whose intent only contains m_1. Now the expert wants to navigate the lattice w.r.t. m_2, on mouse hover the interface highlights all the concepts where the given attribute exists and further highlights the attribute concept in red. The attribute concept of m_2 can be selected. In the running example, if the expert has navigated the lattice w.r.t. the author Amedeo Napoli and she finds some papers on FCA authored by Amedeo Napoli. Now she wants to navigate the concept lattice w.r.t. the keyword *FCA* then she can easily locate the attribute concept of the keyword FCA and navigate to get specific information. The left side of Figure 2 shows the highlighted attribute concept of FCA in RV-Xplorer.

In tree-folder display altering navigation space w.r.t. intent needs the expert to locate the attribute concept by herself by manually checking each of the branches because it represents the concept lattice as a tree. The problem with such a display is that it is not easy to alter the browsing space quickly or change the navigation point of view. Moreover, the sub-lattice connected to a selected concept can not be seen because of the restrictions posed by tree display.

5.5   Area Expansion

Area expansion allows the expert to select several concepts at one time scattered over the concept lattice and gives the overall view of what these concepts contains. These concepts are not necessarily a part of navigation path that the expert is following. It allows the expert to have an overall view of other concepts without starting the navigation process again.

This idea was first put-forth in [14], where they allow the expert to move from one concept lattice to another concept lattice based on the granularity level w.r.t. a taxonomy and a similarity threshold. The concepts in the concept lattice with higher threshold contains more detailed information as compared to the concept lattice built using lesser threshold. One drawback of such kind
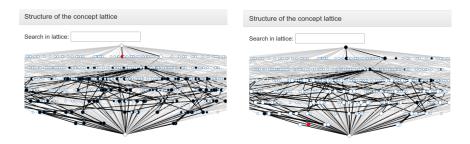
Fig. 2: Left Figure shows the Attribute Concept of FCA and Right Figure shows the Object Concept of *Computing Similarity Dependencies with Pattern Structures*.

of zooming operation is that it requires the computation of several concept lattices. In case of RV-Xplorer, we are dealing with simple concept lattice instead of the one created after using hierarchies meaning that all such kind of information needs to be scaled to obtain a binary context. As we are dealing with concept lattices built from binary contexts, we bend this functionality to suit the needs. It does not require computation of many concept lattices as well as no re-computation is required.

6   Hiding Non-Interesting Parts of the View

One of the most interesting characteristic of RV-Xplorer is that it allows the expert to hide the non-interesting part of the lattice. Let us consider that expert selects a concept c and it contains an attribute which is not interesting for her. She can at any point right click on the concept and select hide sub-lattice. One of the most interesting characteristic of a concept lattice is that if one concept contains some attribute in an intent then all the sub-concepts inherit this attribute. This way if the expert considers one concept as un-interesting then the whole sub-lattice will be considered as uninteresting and hence will be hidden from the expert while navigation. Such kind of functionality enables expert to reduce her navigational space and at the end the concept lattice contains only those concepts which are interesting for the expert.

Similar functionality was first introduced in CreChainDo system [15]. Similar to CREDO [5], CreChainDo allows the expert to pose a query against the standard search engine which returns some results. These results are then organized in the form of a concept lattice and displayed to the expert in the form of folder-tree display. An added advantage of CreChainDo over CREDO is that the former allows expert interaction i.e., the expert can mark the concepts as relevant or irrelevant based on her priorities. After the expert has marked the concept irrelevant the sub-lattice linked to that concept is deleted. Meaning that, it reduces the context based on this feedback and the concept lattice is computed again using the reduced context. In case of RV-Xplorer, the concept lattice is built on top of RDF graphs. Moreover, we do not recompute the lattice or remove anything from the concept lattice. We only hide the non-interesting part

of the lattice to reduce the navigation space of the expert. This way a reduction in the navigation space is performed without re-computing a concept lattice.

7 Related Tools

There have already been many efforts for providing expert the facilities to interact with the concept lattice applied to different domains. In [13], the authors discuss a query-based faceted search for Semantic Web, as a contrast we are mostly dealing with navigational capabilities that can be provided by utilizing the powerful structure introduced by Hasse Diagram. [10] proposes another interesting way of navigating the concept lattice which allows the novice user to navigate through the concept lattice without having to know the structure of the concept lattice. Same is the case with SPARKLIS [12], where user can perform selections and the tool acts as a query builder. As a contrast, RV-Xplorer provides exploration/navigational capabilities over SPARQL query answers with the help of view i.e., a concept lattice for data analysis and information retrieval purposes. Conexp[5] is another tool for visualizing small lattices. As a contrast, RV-Xplorer allows area expansion and also provides guided navigation. [1] discusses that the views generated are easily navigable by machine as well as human agents. Machine agents may access the datasets through SPARQL queries for application development purposes through generic SPARQL queries generating huge number of answers and consequently large number of concepts are provided by `View By` clause. However, when human agents want to access the information through SPARQL query they run specialized queries which do not generate huge number of answers. In the current study we are focusing on manageable number of answers to be visualized by human agents using our visualization software.

An added advantage over these approaches is that RV-Xplorer provides guidance to the expert at each step for making the decision about concept selection. This guidance is provided by showing the user at each step, the contents of the intent of next level, by showing the distribution of the extent with the help of summarization index and finally with the help of global view many other ways of guidance are provided.

8 Discussion

In this study we introduce a new navigational tool for concept lattices called as RV-Xplorer which provides exploration over SPARQL query answers. With the help of guided navigation implemented in RV-Xplorer we were able to answer all the questions posed initially in the scenario. However, this tool is not designed for only specific purpose any kind of concept lattice can be visualized and data from any domain can be analyzed using this tool. The RV-Xplorer tool is still in development and other functionalities should be added such as incremental visualization (w.r.t. a set of given objects and attributes), iceberg visualization

---

[5] `http://conexp.sourceforge.net/`

(given a set of attributes and objects, and a frequency threshold), integration of quality measures, visualization of implications and Duquenne-Guigues basis... We believe that visualization tools, as many other researchers do (see the tools discussed in [6]) are of main importance, not only for FCA but for data mining in general. Accordingly, a new generation of visualization tools should be studied and designed, and RV-Xplorer is an example of this new tools and what can be imagined for supporting the analyst in the mining activity. We also want to perform human evaluation of the tool as discussed in [10] and [13].

References

1. Mehwish Alam and Amedeo Napoli. Defining views with formal concept analysis for understanding SPARQL query results. In *Proceedings of the Eleventh International Conference on Concept Lattices and Their Applications.*, 2014.
2. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
3. Claudio Carpineto, Stanislaw Osiński, Giovanni Romano, and Dawid Weiss. A survey of web clustering engines. *ACM Comput. Surv.*, 41(3):17:1–17:38, 2009.
4. Claudio Carpineto and Giovanni Romano. A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning*, 24(2):95–122, 1996.
5. Claudio Carpineto and Giovanni Romano. Exploiting the potential of concept lattices for information retrieval with CREDO. *J. UCS*, 10(8):985–1013, 2004.
6. Víctor Codocedo and Amedeo Napoli. Formal concept analysis and information retrieval - A survey. In *Formal Concept Analysis - 13th International Conference, ICFCA 2015, Nerja, Spain, June 23-26, 2015, Proceedings*, pages 61–77, 2015.
7. Richard Cole and Gerd Stumme. CEM - A conceptual email manager. In *8th International Conference on Conceptual Structures, ICCS 2000, Darmstadt, Germany, August 14-18, 2000, Proceedings*, pages 438–452, 2000.
8. Claudia d'Amato, Nicola Fanizzi, and Agnieszka Lawrynowicz. Categorize by: Deductive aggregation of semantic web query results. In *ESWC (1)*, 2010.
9. Peter W. Eklund, editor. *Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings*, Lecture Notes in Computer Science. Springer, 2004.
10. Peter W. Eklund, Jon Ducrou, and Peter Brawn. Concept lattices for information visualization: Can novices read line-diagrams? In Eklund [9], pages 57–73.
11. Sébastien Ferré. Camelis: a logical information system to organise and browse a collection of documents. *Int. J. General Systems*, 38(4):379–403, 2009.
12. Sébastien Ferré. Expressive and scalable query-based faceted search over SPARQL endpoints. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II*, 2014.
13. Sébastien Ferré and Alice Hermann. Reconciling faceted search and query languages for the semantic web. *IJMSO*, 7(1):37–54, 2012.
14. Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, and Malika Smaïl-Tabbone. Using domain knowledge to guide lattice-based complex data exploration. In *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, pages 847–852, 2010.
15. Emmanuel Nauer and Yannick Toussaint. Dynamical modification of context for an iterative and interactive information retrieval process on the web. In *Proceedings of the Fifth International Conference on Concept Lattices and Their Applications, CLA 2007, Montpellier, France, October 24-26, 2007*, 2007.