Providing Alternative Declarative Descriptions for Entity Sets using Parallel Concept Lattices

Thomas Gottron[1], Ansgar Scherp[2], and Stefan Scheglmann[1]

[1]WeST – Institute for Web Science and Technologies
University of Koblenz-Landau, Koblenz, Germany
{gottron,schegi}@uni-koblenz.de
[2]Kiel University, Kiel, Germany
Leibniz Information Center for Economics, Kiel, Germany
mail@ansgarscherp.net

Abstract We propose an approach for modifying a declarative description of a set of entities (e.g., a SPARQL query) for the purpose of finding alternative declarative descriptions for the entities. Such a shift in representation can help to get new insights into the data, to discover related attributes, or to find a more concise description of the entities of interest. Allowing the alternative descriptions furthermore to be close approximations of the original entity set leads to more flexibility in finding such insights. Our approach is based on the construction of parallel formal concept lattices over different sets of attributes for the same entities. Between the formal concepts in the parallel lattices, we define mappings which constitute approximations of the extent of the concepts. In this paper, we formalise the idea of two types of mappings between parallel concept lattices, provide an implementation of these mappings and evaluate their ability to find alternative descriptions in a scenario of several real-world RDF data sets. In this scenario, we use descriptions for entities based on RDF classes and seek for alternative representations based on properties associated with the entities.

1 Introduction

Declarative descriptions of sets of entities are used in many scenarios. For instance, when querying a data backend using declarative query languages or in faceted browsing when exploring a data set. In such scenarios it is commonly assumed that a user is aware of all the declarative descriptions he may use. Quite often, however, finding an appropriate description itself is an exploratory task. This is the case in particular when dealing with data which is managed in a de-centralised manner and for which there is no fixed and pre-defined schema.

In such a case, the task of seeking a suitable declarative description for an intended data set is difficult. As the user does not know for sure what data model and vocabulary the data engineers have used to model their data, he might encounter difficulties to formulate an adequate declarative description for the data he is interested in. Even when succeeding to find an initially successful entry point for the description of the desired set of entities, users might not be able to find the best description, i. e. a brief, concise and exhaustive description.

Existing approaches for finding alternative descriptions so far operate locally, i.e. they iteratively add or remove single declarative constraints [9]. While providing some support, these approaches cannot help in breaking out of a local optimum. Furthermore, they do not provide new inspirations to the users, which enable them to think out of the box and get new ideas for how to describe the set of data they are interested in. In traditional document search systems such problems have been encountered already and addressed with methods such as automatic result set expansion, relevance feedback and query reformulation. Similar approaches have recently been investigated for semantic web data. For example, the LOD search engine LODatio provides services to generate related, alternative SPARQL queries [9]. Other approaches aim at finding clusters of related entities [19] or refining graph-based queries [21].

In this paper, we present a generic method for finding alternative declarative descriptions for a given set of entities. It is based on building parallel formal concept lattices [22] over different sets of attributes of the data at hand and providing mappings between these lattices. These mappings allow to find alternative descriptions while preserving the set of entities as far as possible. Given the structure of formal concept lattices, we restrict ourself in this paper on conjunctive forms of declarative descriptions. However, the method is generic as the lattices can be built over arbitrary attributes of the data. A mapping between these lattices can make use of the set of described entities (i. e. the *extent* of the formal concepts) to find alternative descriptions (i. e. the *intent* of the formal concepts) for close approximations of the entity set. We present two such mappings and analyse their behaviour and quality in finding alternative descriptions of formal concepts from different lattices.

The rest of the paper is structured as follows: We provide a high level overview of the idea of suggesting alternative declarative descriptions using formal concept lattices in Section 2. In Section 3, we briefly review formal concept analysis [22] which provides the foundation for our work before we present a thorough formalisation of our idea in Section 4. In Section 5, we implement our approach and investigate its performance for a particular use case of finding alternative representations based on properties for sets of entities which are initially described on the basis of RDF type classes. We review related work in Section 6, before we conclude the paper in 7.

2 Overview to our Approach

The idea of our approach for finding alternative declarative descriptions for a set of entities is based on two assumptions:

1. There are different sets of attributes which can be used to describe the data. In the context of RDF such different attributes can be the class types of entities, the properties used to describe them, the objects they are linked to, the vocabularies used to model them or the data sources providing information about them.
2. The user has not found an ideal declarative description in the sense that the described data set either contains too many or too few entities. Accordingly, an alternative description may extend the data set with additional entities (as long as none of the original entities is lost) or may restrict the data set by removing some entities (as long as no new entities are added).

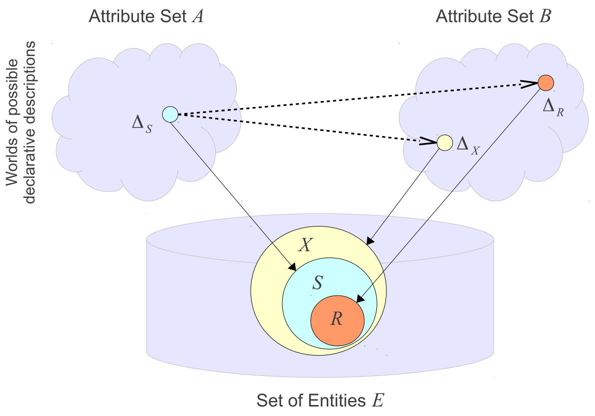


Figure 1. Our approach is based on the idea of finding declarative descriptions $\mathit{\Delta}_X$ or $\mathit{\Delta}_R$ using an alternative set of attributes which approximates a set of entities S defined by declarative description $\mathit{\Delta}_S$ as close as possible.

Based on these assumptions, we build parallel formal concept lattices using different attribute sets. The obtained lattices structure the data set under different conjunctive combinations of the available attributes and their observed combinations. For a given node in one lattice we then define mappings which look for alternative descriptions in other lattices while trying to preserve the set of described entities as far as possible. The result is a set of entities which extends or restricts the original set as much as required to find a concise declarative description using the alternative attribute set.

Figure 1 illustrates the approach. Assume, we can alternatively use attribute sets A or B to describe entities in a set E. The two sets of attributes give rise to two worlds of possible declarative descriptions for sets of entities. Figure 1 depicts one element $\mathit{\Delta}_S$ of the world of descriptions using the attribute set A. This description corresponds to a subset S of entities in E. The idea is to look for alternative descriptions using the set of attributes B. Such descriptions might correspond to extensions X of S (as in the case of $\mathit{\Delta}_X$) or to reductions R of S (as in the case of $\mathit{\Delta}_S$).

The advantage of using a lattice structure in the world of possible declarative descriptions is that we can easily navigate in the hierarchy of sets and their subsets and supersets while having at the same time the descriptions of these sets readily available. Thus, we can efficiently explore the space of possible alternative descriptions.

3 A Review of Formal Concept Analysis

Formal concept analysis has been introduced as a mathematical framework for structuring data and deriving concepts based on the objects belonging to a concept and their common attributes [22]. Therefore, the foundation is a formal context of objects and their attributes.

Definition 1 (Formal Context, Derivative). *Let G and M be sets and $I \subseteq G \times M$ a relation. The elements in G are usually interpreted as objects, the elements in M as attributes and the reading of $(g, m) \in I$ is that object g has attribute m. Then (G, M, I) provides a* formal context*.*

Let $A \subseteq G$ be a set of objects in G. Then the derivative *A' of A is defined as $A' := \{m \in M : (g, m) \in I, \forall g \in A\} \subseteq M$. Likewise, for a subset B of attributes (i. e. $B \subseteq M$), the* derivative *B' is defined as $B' := \{g \in G : (g, m) \in I, \forall m \in B\} \subseteq G$.*

Thus, A' is the set of attributes which is common to all objects in A and B' corresponds to the set of all objects which exhibit all the attributes in B. The definition of formal concepts is based on formal contexts and the notion of derivatives.

Definition 2 (Formal Concept, Extent, Intent). *A* formal concept *(A, B) is defined to consist of a subset $A \subseteq G$ and a subset $B \subseteq M$, for which $A' = B$ and $B' = A$. For such a formal concept, the set of objects belonging to the concept (so A) is the* extent *and the set of attributes (so B) is the* intent *of the concept. The set of all formal concepts in a formal context is denoted with $\mathfrak{B}(G, M, I)$.*

According to this definition, we can use the derivative operator to shift between the two representations for a formal concept: its extent and its intent. Furthermore, we always have two particular formal concepts: the top concept $\top = (G, \emptyset)$ containing all objects and the bottom concept $\bot = (\emptyset, M)$ containing all attributes.

Definition 3 (Formal Concept Lattice). *A* formal concept lattice *$\underline{\mathfrak{B}}(G, M, I)$ is defined as the set of all formal concepts together with the partial order $\leq$ induced by the set inclusion, i. e. $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2$ (which is equivalent to $B_1 \supseteq B_2$).*

Corollary 1 (Top and Bottom Concepts in a Formal Concept Lattice). *From Definition 3, we can directly deduce that $C \leq \top, \forall C \in \mathfrak{B}$ and $\bot \leq C, \forall C \in \mathfrak{B}$.*

Example 1. We consider a set G of ten objects which for the sake of simplicity we simply enumerate from 1 to 10. The set M of attributes shall be $\{a, b, c\}$ and the left side in Table 1 visualises the relation I of which object has which attributes.

The tuple $(\{1, 4, 6, 9, 10\}, \{a, b\})$ constitutes a formal concept. The derivative of $\{1, 4, 6, 9, 10\}$ is $\{a, b\}$, as the objects $1, 4, 6, 9$ and 10 have the attributes a and b in common. Inversely, the derivative of $\{a, b\}$ is $\{1, 4, 6, 9, 10\}$ as these are the only objects exhibiting these properties.

When constructing a formal concept lattice over the formal context from Table 1, we obtain a structure as shown on the left hand side in Figure 2. The visualisation arranges concepts from the top concept above to the bottom concept below and connects two concepts with a line, if there is no other concept between them w.r.t $\leq$.

4 Using Parallel Lattices to Derive Alternative Descriptions

We now present our idea of building parallel concept lattices and how to exploit mappings between theses lattices for finding alternative descriptions.

Table 1. Example of two formal contexts over two different attribute sets.

Object	a	b	c
1	×	×	×
2	×		
3		×	
4	×	×	
5	×		×
6	×	×	
7			×
8		×	×
9	×	×	
10	×	×	×

Object	x	y	z
1		×	×
2	×		
3		×	
4	×	×	×
5		×	
6	×	×	×
7		×	
8	×		
9		×	×
10	×	×	

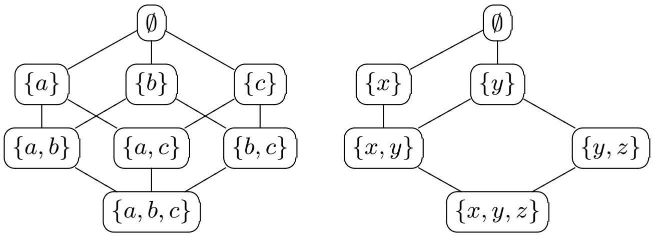


Figure 2. Formal concept lattice structures based on the relations in Table 1. The concepts are represented by their intent—which provides a better overview.

4.1 Parallel Formal Concept Lattices

Assume we have two sets M_1 and M_2 which can serve as attributes to describe the objects in G. Accordingly, there are two relations I_1 and I_2. Then, we can construct two parallel formal concept lattices $\underline{\mathfrak{B}}(G, M_1, I_1)$ and $\underline{\mathfrak{B}}(G, M_2, I_2)$. Note, that while the intent of the concepts in parallel lattices is defined over two different sets of attributes, the extent of the concepts are always based on the same set G. The idea of parallel lattices can easily be extended to an arbitrary number of attribute sets.

Example 2 (Parallel Concept Lattice). In Table 1, we have listed a second relation I_2 over the set of attributes $M_2 = \{x, y, z\}$. In I_2 the same objects are related to a different set of attributes. If we construct a formal concept lattice over this relation we obtain the lattice on the right hand side in Figure 2.

4.2 Extension and Reduction Mappings Between Parallel Concept Lattices

We now introduce two mappings between parallel lattices which are defined over the extent of the formal concepts in the lattices. Such mappings will allow for the approximation of the extent of a concept from a base lattice using the extent of a concept in an alternative lattice. The concept in an alternative lattice provides an alternative representation via its intent composed over a different set of attributes.

In this section, we use $\underline{\mathfrak{B}}(G, M_1, I_1)$ and $\underline{\mathfrak{B}}(G, M_2, I_2)$ as two formal concept lattices defined over the same set G and different sets M_1 and M_2. $\underline{\mathfrak{B}}(G, M_1, I_1)$ will serve as the *base lattice* for which we seek descriptions of its concepts in the *alternative lattice* $\underline{\mathfrak{B}}(G, M_2, I_2)$. For short notation we will refer to them as $\underline{\mathfrak{B}}_i := \underline{\mathfrak{B}}(G, M_i, I_i)$ and to the set of formal concepts by $\mathfrak{B}_i := \mathfrak{B}(G, M_i, I_i)$.

Definition 4 (Maximum Reduction). *Let $C_1 = (A_1, B_1)$ be a formal concept in $\mathfrak{B}_1$. We define the set of reductions of C_1 on an alternative lattice $\mathfrak{B}_2$ as $\operatorname{red}(C_1) \in \mathcal{P}(\mathfrak{B}_2)$ by:*

$$\operatorname{red}(C_1) := \{(A_2, B_2) \in \mathfrak{B}_2 : A_1 \supseteq A_2\} \tag{1}$$

Technically, the set $\operatorname{red}(C_1)$ contains all formal concepts in $\mathfrak{B}_2$ where the extent is a subset of A_1. We then define the maximum reduction *set $\operatorname{max-red}(C_1)$ of a given concept C_1 as:*

$$\operatorname{max-red}(C_1) := \{C_2 \in \operatorname{red}(C_1) : (\nexists C_2' \in \operatorname{red}(C_1) : C_2 \leq C_2')\} \tag{2}$$

In other words: the maximum reduction contains formal concepts in the alternative lattice for which the extent is an as large as possible reduced (i. e. subset) approximation of A_1. This means, there is no other formal concept which is larger (under the partial order $\leq$) and which still has an extent that is a subset of A_1. If no larger reduction is found, $\operatorname{max-red}$ will contain the bottom concept as trivial solution.

Theorem 1 (Perfect Approximation in $\operatorname{max-red}$). *For a perfect approximation the maximum reduction set is of size 1, i. e. if $\exists B_2 \in M_2 : (A_1, B_2) \in \operatorname{max-red}(A_1, B_1)$, then $|\operatorname{max-red}(A_1, B_1)| = 1$.*

Proof: *Trivial, as the perfect approximation is a superset of all reductions. Thus, there cannot be any other concept in* $\operatorname{max-red}$.

Definition 5 (Minimum Extension). *Let $C_1 = (A_1, B_1)$ be a formal concept in $\mathfrak{B}_1$. We define the set of extensions of C_1 on an alternative lattice $\mathfrak{B}_2$ as $\operatorname{ext}(C_1) \in \mathcal{P}(\mathfrak{B}_2)$ by:*

$$\operatorname{ext}(C_1) := \{(A_2, B_2) \in \mathfrak{B}_2 : A_1 \subseteq A_2\} \tag{3}$$

We then define the minimum extension *set $\operatorname{min-ext}(C_1)$ of a given concept C_1 as:*

$$\operatorname{min-ext}(C_1) := \{C_2 \in \operatorname{ext}(C_1) : (\nexists C_2' \in \operatorname{ext}(C_1) : C_2' \leq C_2)\} \tag{4}$$

In words again: the minimum extension set contains formal concepts in the alternative lattice for which the extent is an as small as possible extension of A_1. If no smaller extension is found, $\operatorname{min-ext}$ will contain the top concept as trivial solution.

Theorem 2 (Size of $\operatorname{min-ext}$). *There is only one concept in the minimum extension, i. e. $|\operatorname{min-ext}(C_1)| = 1$ for all C_1.*

Proof: *Assume we have $C_2 = (A_2, B_2) \in \operatorname{min-ext}(C_1)$ and $\tilde{C}_2 = (\tilde{A}_2, \tilde{B}_2) \in \operatorname{min-ext}(C_1)$. Now, let $C_1 = (A_1, B_1)$, then we have $A_2 \cap \tilde{A}_2 \subset A_1$. Thus, we would*

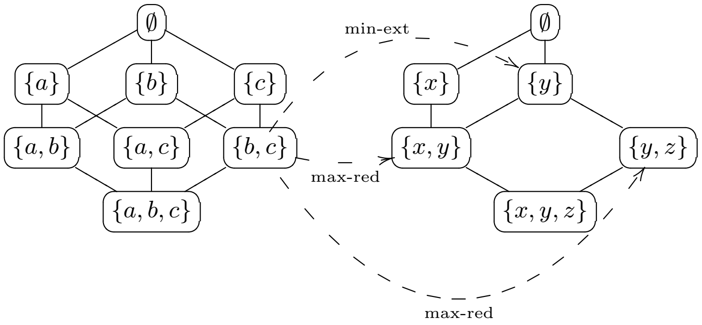


Figure 3. max-red and min-ext mapping between two parallel formal concept lattice structures.

have a formal concept $\hat{C}_2 = (A_2 \cap \tilde{A}_2, B_2 \cup \tilde{B}_2)$ *which is in* $\text{ext}(C_1)$ *and for which* $\hat{C}_2 \leq C_2$ *and* $\hat{C}_2 \leq \tilde{C}_2$*. This is a contradiction to the assumption that* C_2 *and* $\tilde{C}_2$ *are in* $\text{min-ext}(C_1)$*. Thus, there cannot be to two elements in* $\text{min-ext}(C_1)$.

To find alternative declarative descriptions for a given concept C_1 in the base lattice $\underline{\mathfrak{B}}_1$ we map this concept onto the sets $\text{max-red}(C_1)$ and $\text{min-ext}(C_1)$ in the alternative lattice $\underline{\mathfrak{B}}_2$.

Example 3 (Maximum Reduction and Minimum Extension). We use once more the two lattices depicted in Figure 2. Let $\mathfrak{B}_{abc} = \mathfrak{B}(G, \{a,b,c\}, I_1)$ and $\mathfrak{B}_{xyz} = \mathfrak{B}(G, \{x,y,z\}, I_2)$, where I_1 and I_2 are defined as in Table 1. We now pick the formal concept $(\{1,4,6,9,10\}, \{b,c\})$ and compute the min-ext mapping for this concept. In a first step we compute the set $\text{ext}(\{1,4,6,9,10\}, \{b,c\})$. The only two concepts in $\mathfrak{B}_{xyz}$ which satisfy that their extent is a superset of $\{1,4,6,9,10\}$ are the top concept $(G, \emptyset)$ and $(\{1,3,4,5,6,7,9,10\}, \{y\})$. As the concept with intent y is smaller than the top concept, the minimum extension is:

$$\text{min-ext}(\{1,4,6,9,10\}, \{b,c\}) = \{(\{1,3,4,5,6,7,9,10\}, \{y\})\}$$

This is visualised via a dashed arrow in Figure 3 labelled min-ext. The maximum reduction in this case consists of the two concepts $(\{4,6,10\}, \{x,y\})$ and $(\{1,4,6,9\}, \{y,z\})$. The corresponding mapping is marked via dashed arrows labelled max-red.

5 Experiments

As exemplary use case and evaluation scenario, we consider the task of approximating a set of Linked Data entities described through RDF type statements via a description based on properties. This task is of importance for SPARQL query recommendations in search engines [9], for deriving programmable interfaces on RDF data or when computing recommendations which vocabularies to use when modelling Linked Data [16]. Furthermore, it has been observed that sets of properties can provide good descriptors for sets of types [8]. As such the two sets of features seem promising for an approximation setting via parallel lattices.

5.1 Implementation

We used the Colibri library[1] for computing formal concept lattices [11]. The library provides a simple interface to iteratively add tuples of entities and associated attributes in order to define a formal context. These tuples can easily be extracted from RDF triples for our use case. To this end, it is sufficient to distinguish between triples using rdf:type as predicate and triples using any other URI as predicate. From the rdf:type predicate triples, we pass the subject and object, i. e. the class type URI, of the triple as entity-attribute tuple to the library for constructing a type based concept lattice. For all other triples, we pass the subject and the predicate of the triple as entity-attribute tuple to a second Colibri instance for constructing the property-based lattice.

To implement the max-red and min-ext mappings, we employ a traversal of the lattices. For $\text{max-red}(C)$, we start from the bottom concept in the alternative concept lattice structure and iteratively seek upper neighbour concepts as long as they fulfil the $\text{ext}(C)$ criteria. For computing min-ext, we seek a suitable concept approximation in a similar fashion starting from the top concept and moving downwards.

5.2 Quality Metrics for the Approximations

First of all, we consider how often it is actually possible to find a non-trivial approximation in the sense that for the best match we found a concept different from top (for min-ext) and bottom (for max-red).

Furthermore, we use information retrieval metrics to evaluate how accurate are the approximative sets compared to the original set of entities. Assume, we have formal concepts $C_1 = (A_1, B_1)$ in the base lattice and a concept $C_2 = (A_2, B_2)$ in the alternative lattice. We can measure the quality of the approximation of C_1 through C_2 by means of recall (*r*) and precision (*p*) on the extent of the two concepts:

$$r(C_1, C_2) = \frac{|A_1 \cap A_2|}{|A_1|},\ p(C_1, C_2) = \frac{|A_1 \cap A_2|}{|A_2|}$$

Example 4 (Recall and Precision). To continue our example of the mappings from Figure 3: We approximate the concept $(\{1, 4, 6, 9, 10\}, \{b, c\})$ with a concept from the maximum reduction: $(\{1, 4, 6, 9\}, \{y, z\})$. In this case, we observe a recall of:

$$r = \frac{|\{1, 4, 6, 9\}|}{|\{1, 4, 6, 9, 10\}|} = \frac{4}{5} = 0.8$$

The precision for this approximation is $p = 1$.

Please keep in mind that for a given concept there might be multiple maximum reductions. Therefore, we select the best recall and precision values which can be achieved for the approximative description in order to judge the quality of the best proposed description. This means, we compute for each concept the best recall and precision values achieved over the sets of all concepts provided as maximum reductions and select the highest score. Formally this corresponds to:

[1] https://code.google.com/p/colibri-java/, accessed: 12 Jan, 2014

$$
\begin{aligned}
\textit{r-max}_{\mathrm{max\text{-}red}}(C) &= \max_{C' \in \mathrm{max\text{-}red}(C)} (r(C, C')) \\
\textit{p-max}_{\mathrm{max\text{-}red}}(C) &= \max_{C' \in \mathrm{max\text{-}red}(C)} (p(C, C'))
\end{aligned}
$$

To assess the global quality, we aggregate the macro average of these values over all concepts in the base lattice:

$$
\begin{aligned}
\textit{Avg r-max}_{\mathrm{max\text{-}red}} &= \frac{1}{|\mathfrak{B}_1|} \sum_{C \in \mathfrak{B}_1} \textit{r-max}_{\mathrm{max\text{-}red}}(C) \\
\textit{Avg p-max}_{\mathrm{max\text{-}red}} &= \frac{1}{|\mathfrak{B}_1|} \sum_{C \in \mathfrak{B}_1} \textit{p-max}_{\mathrm{max\text{-}red}}(C)
\end{aligned}
$$

Likewise, we define the aggregated metrics for $\mathrm{min\text{-}ext}$. The only difference is that there is no need for identifying a maximum value, as there is only one candidate.

5.3 Data Sets

For our experiments, we have worked with the Billion Triple Challenge[2] (BTC) from 2012. The BTC data set was crawled from the web using a linked data spider. Thus, it represents a real-world data set of mixed quality from various application domains. While the entire BTC data set is too large to be processed with a standard implementation for formal concept analysis, the data set served as a rich resource for sampling smaller data sets. We grouped the data by the pay level domain (PLD) of the servers from which the data has been crawled originally. This lead to a total of 840 smaller data sets, each of which can be considered to be controlled by an individual data provider [4]. We selected 20 of these smaller data sets, which each contained approximately between 1 and 40 million triples. Processing data sets of this size with the Colibri implementation took between a few minutes and up to 12 hours to compute all $\mathrm{max\text{-}red}$ and $\mathrm{min\text{-}ext}$ mappings for all concepts in a pair of lattices.

For these 20 data sets, we identified the number of modelled entities, computed the base lattices using the class type definitions and the alternative lattices over the properties of the entities. Furthermore, we computed the normalised mutual information I_0 between type and property definitions. This normalised mutual information is a measure of redundancy, i. e., how well one set of attributes can explain the respective other[3]. Table 2 lists the data sets we finally used for our experiments, their size and the degree of redundancy.

[2] BTC 2012 data set: http://km.aifb.kit.edu/projects/btc-2012/, accessed: 12 Jan, 2014

[3] For details we refer to [8].

Table 2. Data sets used for evaluation and their characteristics

Data set (PLD)	Triples	Entities	I_0
bbc.co.uk	1,895,817	345,087	0.717
concordia.ca	1,705,287	359,215	0.799
europa.eu	7,362,172	579,497	0.973
fao.org	1,065,538	44,095	0.954
geovocab.org	938,434	260,427	0.989
identi.ca	36,969,163	4,004,911	0.972
kasabi.com	6,170,661	974,307	0.997
legislation.gov.uk	39,200,538	4,850,236	0.897
lexvo.org	3,753,070	751,022	1.000
loc.gov	7,605,348	1,714,943	0.764
neuinfo.org	1,268,368	333,061	0.587
nytimes.com	900,892	57,072	1.000
ontologycentral.com	29,447,217	3,773,117	0.879
opera.com	44,331,144	3,547,299	0.867
ordnancesurvey.co.uk	5,765,802	589,165	0.845
oreilly.com	5,447,983	6,307	0.894
pokepedia.fr	1,043,818	25,190	0.659
rdfize.com	14,949,592	766,905	0.902
semanticweb.org	1,888,030	137,742	0.817
soton.ac.uk	2,813,256	356,701	0.690

5.4 Results and Discussion

On the base and alternative lattice structures obtained for each of the PLD data sets, we evaluated the ability of our $\mathrm{max\text{-}red}$ and $\mathrm{min\text{-}ext}$ mappings to find alternative declarative descriptions. To this end, we iterated over all the concepts in the base lattice except top and bottom and computed for each of them approximations using $\mathrm{max\text{-}red}$ and $\mathrm{min\text{-}ext}$ in the alternative lattice defined over properties. For these approximations, we computed the average recall and precision and the number of concepts for which we could not find a better match than the top or bottom concept in the property lattice.

Table 3 gives an overview of how many concepts could be approximated successfully with a concept which was not the trivial match of top (for $\mathrm{min\text{-}ext}$) or bottom (for $\mathrm{max\text{-}red}$). We can see that for some data sets it is more difficult to find approximations than for others. The lowest performance is observed on pokepedia.fr, where only 2% of the concepts lead to a non-trivial approximation. This can be explained by two reasons: The number of alternative concepts is much lower than the number of concepts in the base lattice $(78 \ll 7556)$. Accordingly, there is much less potential to find a good match. Moreover, as can be seen when looking at the I_0 value of this data set in Table 2, the type and property sets are not correlated very strong. Conversely, high values of I_0 and a larger number of alternative concepts are a good indicator that the mapping will find a match. Good examples for this case are identi.ca, kasabi.com and legislation.gov.uk. Moreover we see, that see that $\mathrm{min\text{-}ext}$ tendentially finds more approximations then $\mathrm{max\text{-}red}$.

Table 4 shows the performance for successfully finding approximated declarative descriptions w.r.t. the precision and recall metrics. We can see that on average the values

Table 3. Number of concepts which could be approximated by max-red and min-ext.

Data set (PLD)	max-red		min-ext		Base Concepts	Alternative Concepts
bbc.co.uk	27	(38.57%)	63	(90.00%)	70	2396
concordia.ca	9	(45.00%)	15	(75.00%)	20	94
europa.eu	24	(63.16%)	35	(92.11%)	38	19046
fao.org	23	(39.66%)	45	(77.59%)	58	18007
geovocab.org	16	(64.00%)	20	(80.00%)	25	117
identi.ca	14	(93.33%)	13	(86.67%)	15	349
kasabi.com	8	(80.00%)	10	(100.00%)	10	52
legislation.gov.uk	10	(100.00%)	10	(100.00%)	10	3168
lexvo.org	6	(100.00%)	6	(100.00%)	6	327
loc.gov	49	(70.00%)	60	(85.71%)	70	1637
neuinfo.org	9	(47.37%)	9	(47.37%)	19	927
nytimes.com	4	(100.00%)	4	(100.00%)	4	69
ontologycentral.com	10	(47.62%)	10	(47.62%)	21	68739
opera.com	4	(80.00%)	4	(80.00%)	5	3073
ordnancesurvey.co.uk	27	(43.55%)	49	(79.03%)	62	171
oreilly.com	21	(80.77%)	24	(92.31%)	26	142
pokepedia.fr	152	(2.01%)	7553	(99.96%)	7556	78
rdfize.com	8	(100.00%)	8	(100.00%)	8	22
semanticweb.org	402	(39.68%)	997	(98.42%)	1013	8331
soton.ac.uk	49	(47.57%)	91	(88.35%)	103	4072

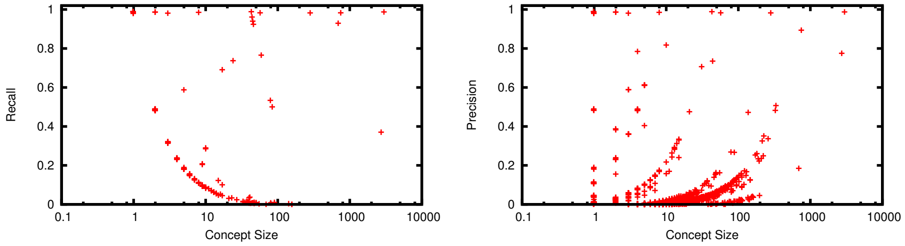
are quite high indicating a good capability of our approach for approximating the sets of entities. However, for a few data sets the quality of the approximations is lower than for the others. We can again point out pokepedia.fr which shows the lowest performance under both approximation mappings. Again, the explanation is the low correlation between the attribute sets as well as low number of alternative concepts. Also the data sets which behave good are consistent. For the data sets mentioned above (identi.ca, kasabi.com, and legislation.gov.uk) we get very good approximations of high quality. When comparing the two mapping methods, we see that the values for max-red are tendentially higher than the ones for min-ext. Combining this with the observation made above, we can say, that max-red might find less approximations but of higher quality, while min-ext finds more approximations which are of slightly lower average quality.

Also when looking into the obtained alternative descriptions we observed a plausible behaviour. In the identi.ca data set, for example, entities of type rss:Item and sioc:MicroblogPost were described as having the properties such as foaf:maker, sioc:has_discussion, rss:link and dcterms:date, which suits the semantics of the RDF types.

To obtain deeper insights into the behaviour of our mapping functions, we compared the quality of their approximations to other indicators. Visual inspection revealed that the size of the extent of a concept seems to play an important role. In Figure 4, we see a scatter plot of the size of the concepts and the quality of approximation for max-red and min-ext. The plot has been generated over the difficult pokepedia.fr data set, but demonstrates quite nicely a behaviour which we observed also for other datasets. We can see a general trend for max-red to achieve lower recall values for larger concepts. This is plausible as for higher concepts it will be difficult to get a common alternative

Table 4. Results on different data sets when approximating type descriptions by properties

Data set (PLD)	max-red		min-ext	
	Avg. r	Avg. p	Avg. r	Avg. p
bbc.co.uk	0.686	1.000	1.000	0.265
concordia.ca	0.977	1.000	1.000	0.539
europa.eu	0.943	1.000	1.000	0.636
fao.org	0.808	1.000	1.000	0.510
geovocab.org	0.693	1.000	1.000	0.512
identi.ca	0.938	1.000	1.000	0.909
kasabi.com	1.000	1.000	1.000	0.807
legislation.gov.uk	0.963	1.000	1.000	0.907
lexvo.org	0.987	1.000	1.000	0.534
loc.gov	0.688	1.000	1.000	0.473
neuinfo.org	0.444	1.000	1.000	0.210
nytimes.com	1.000	1.000	1.000	1.000
ontologycentral.com	0.856	1.000	1.000	0.859
opera.com	1.000	1.000	1.000	0.500
ordnancesurvey.co.uk	0.770	1.000	1.000	0.517
oreilly.com	0.831	1.000	1.000	0.677
pokepedia.fr	0.294	1.000	1.000	0.017
rdfize.com	0.874	1.000	1.000	0.929
semanticweb.org	0.465	1.000	1.000	0.141
soton.ac.uk	0.708	1.000	1.000	0.401



(a) max-red

(b) min-ext

Figure 4. Comparison of the size of a concept and the quality of the achieved approximations on the pokepedia.fr data set.

description which does not introduce other objects. On the other hand, for min-ext we observe an increase in precision for larger concepts. Also this behaviour is plausible: if a larger concept is extended by a few additional elements, the overall precision remains quite high. Vice versa adding a few elements to a small concept drastically decreases precision.

As consequence, we propose to operate in practical applications in a mixed mode combining both methods. For small concepts it seems more fruitful to rather use minimal extension, while using for large concepts a maximal reduction. In this way, we can expect a good behaviour in general.

6 Related Work

Formal concept analysis emerged in the 80s from restructuring lattice theory in order to widen for its adoption in practice [23,22]. Various efficient algorithms have been proposed to compute formal concepts and construct a lattice from it such as [13,18]. Formal concept lattices and their creation has been successfully applied in the past in the Semantic Web such as for analysing Linked Data [10,1] or semi-automatically constructing OWL DL ontologies [2,20]. Ferré et al. [6] describe an approach to build arbitrary relations in a formal concept lattice for the purpose of navigation. They compute a set of navigation links for a query q in order to refine the query. As concrete application scenario, they implemented a navigable UNIX file system that allows for exploring neighbouring concepts. While this allows to find *related* concepts, the approach does not aim at suggesting alternative concepts (and their attribute sets). Other works use formal concept analysis to compute mappings between the concepts of two (or more) ontologies [17,3,5]. In a first step, a lattice is constructed by analyzing a set of entities such as documents w.r.t. to the concepts defined in the ontologies. Subsequently, the lattice is used to find, e. g., class subsumption relations and class equivalence relations. Thus, the ontology alignment approaches apply a single lattice for computing the mappings between ontologies that are provided by different independent organisations. In contrast, we compute mappings between two lattices that are constructed over two kinds of intents (namely the RDF properties and the RDF types) taken from a linked data set and ontology that is curated from a single organisation and pay-level domain, respectively. Although not using formal concept analysis, we like to mention the ontology mapping work by Parundekar et al. [14,15] that computes concept mappings between two independent sources of Linked Data by considering conjunctions and disjunctions of restriction classes.

Finding alternative declarative descriptions for a set of entities can also be related to query recommendation approaches. For example, Meij et al. [12] align query logs with DBpedia concepts. They use different features for query recommendation including the history of previous queries and suggesting concepts related to the current candidate concept based on the number of concepts pointing to it (using the DBpedia property dbpprop:redirect) or concepts linked from it (count of skos:subject) [12]. Hermes [21] guides the users through the query refinement process by providing simple means such as a travel history, navigation panels, and result list panel. However, it does not proactively provide query suggestions. Based on an initial keyword query, Than et al. [19] propose a system that computes k clusters of RDF entities and presents them to the users. The clusters are computed based on their matches to the keyword query and consist of m property-value pairs. The users select relevant clusters and refine the search. By this, the users implicitly contribute to an improved mapping of class and property combinations observed with entities that match the same keyword. Recommending related queries is also part of the LODatio system [9]. Here, Google-style modifications of SPARQL queries are provided in terms of monotonic generalisations and refinements, i. e., removing query patterns or adding new query patterns. Finally it is worth mentioning that there is also work on exact query reformulation on OWL-DL ontologies [7]. In summary, many approaches for query recommendation developed so far require the availability of a proper query log for extracting ranking information. Only a few approaches can conduct a query recommendation without such history knowledge. Among those, none have used formal concept analysis.

7 Conclusions

We presented an approach for finding alternative declarative descriptions of sets of entities which allow for a certain flexibility with respect to the entities belonging to the set. Our approach is based on using parallel formal concept lattices over the same set of objects based on different sets of descriptive attributes. We defined mappings $\mathrm{max\text{-}red}$ and $\mathrm{min\text{-}ext}$ on a base lattice, which approximate the extent of a given formal concept in an alternative concept lattice and use the intent of the approximated concept as alternative declarative description. We implemented the approach and performed experiments on 20 different data sets using formal concept lattices constructed over class types and properties of entities. The experiments showed the potential of the approach and its applicability for the presented use case.

In the future we, we plan to extend the work to implement a disjunction operator. Finally, we want to implement the approach in an application system with end users to perform a user evaluation of the alternative declarative descriptions. This will be done in the context of faceted browsing or an LOD search system.

Acknowledgements We thank our student Tobias Schmidt for his support in implementing the parallel concept lattices. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013), REVEAL (Grant agree number 610928).

References

1. Beck, N., Scheglmann, S., Gottron, T.: Linda: A service infrastructure for linked data analysis and provision of data statistics. In: Cimiano, P., Fernández, M., Lopez, V., Schlobach, S., Völker, J. (eds.) The Semantic Web: ESWC 2013 Satellite Events, Lecture Notes in Computer Science, vol. 7955, pp. 225–230. Springer Berlin Heidelberg (2013)
2. Cimiano, P., Hotho, A., Stumme, G., Tane, J.: Conceptual knowledge processing with formal concept analysis and ontologies. In: Concept Lattices, pp. 189–207. Springer (2004)
3. Curé, O., Jeansoulin, R.: An fca-based solution for ontology mediation. In: Proceedings of the 2Nd International Workshop on Ontologies and Information Systems for the Semantic Web. pp. 39–46. ONISW '08, ACM, New York, NY, USA (2008)
4. Ding, L., Finin, T.: Characterizing the Semantic Web on the web. In: ISWC. pp. 242–257. Springer (2006)
5. Fan, L., Xiao, T.: An automatic method for ontology mapping. In: Apolloni, B., Howlett, R.J., Jain, L.C. (eds.) Knowledge-Based Intelligent Information and Engineering Systems, 11th International Conference, KES 2007, XVII Italian Workshop on Neural Networks, Vietri sul Mare, Italy, September 12-14, 2007, Proceedings, Part III. Lecture Notes in Computer Science, vol. 4694, pp. 661–669. Springer (2007)
6. Ferré, S., Ridoux, O., Sigonneau, B.: Arbitrary relations in formal concept analysis and logical information systems. In: ICCS. pp. 166–180. Springer (2005)
7. Franconi, E., Kerhet, V., Ngo, N.: Exact query reformulation with first-order ontologies and databases. In: Logics in Artificial Intelligence. pp. 202–214. Springer (2012)

8. Gottron, T., Knauf, M., Scheglmann, S., Scherp, A.: A systematic investigation of explicit and implicit schema information on the linked open data cloud. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) The Semantic Web: Semantics and Big Data. Lecture Notes in Computer Science, vol. 7882, pp. 228–242. Springer Berlin Heidelberg (2013)
9. Gottron, T., Scherp, A., Krayer, B., Peters, A.: Lodatio: Using a schema-level index to support users in finding relevant sources of linked data. In: KCAP. pp. 105–108. ACM (2013)
10. Kirchberg, M., Leonardi, E., Tan, Y.S., Link, S., Ko, R.K.L., Lee, B.S.: Formal concept discovery in semantic web data. In: Formal Concept Analysis - 10th International Conference, ICFCA 2012, Leuven, Belgium, May 7-10, 2012. Proceedings. pp. 164–179. Springer (2012)
11. Lindig, C.: Mining patterns and violations using concept analysis. Tech. rep., Saarland University, Software Engineering Chair (2007)
12. Meij, E., Bron, M., Hollink, L., Huurnink, B., de Rijke, M.: Learning semantic query suggestions. In: ISWC. pp. 424–440. Springer (2009)
13. Nourine, L., Raynaud, O.: A fast algorithm for building lattices. Inf. Process. Lett. 71(5-6), 199–204 (1999)
14. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Linking and building ontologies of linked data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) International Semantic Web Conference (1). Lecture Notes in Computer Science, vol. 6496, pp. 598–614. Springer (2010)
15. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Discovering concept coverings in ontologies of linked data sources. In: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (eds.) International Semantic Web Conference (1). Lecture Notes in Computer Science, vol. 7649, pp. 427–443. Springer (2012)
16. Schaible, J., Gottron, T., Scheglmann, S., Scherp, A.: Lover: support for modeling data using linked open vocabularies. In: Guerrini, G. (ed.) EDBT/ICDT Workshops. pp. 89–92. ACM (2013)
17. Stumme, G., Maedche, A.: Fca-merge: Bottom-up merging of ontologies. In: Nebel, B. (ed.) Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001. pp. 225–234. Morgan Kaufmann (2001)
18. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with titanic. Data & Knowledge Engineering 42(2), 189 – 222 (2002)
19. Tran, T., Ma, Y., Cheng, G.: Pay-less entity consolidation: exploiting entity search user feedbacks for pay-as-you-go entity data integration. In: Web Science. pp. 317–325. ACM (2012)
20. Völker, J., Rudolph, S.: Lexico-logical acquisition of owl dl axioms. In: Formal Concept Analysis, 6th International Conference, ICFCA 2008, Montreal, Canada, February 25-28, 2008, Proceedings. pp. 62–77. Springer (2008)
21. Wang, H., Penin, T., Xu, K., Chen, J., Sun, X., Fu, L., Liu, Q., Yu, Y., Tran, T., Haase, P., Studer, R.: Hermes: a travel through semantics on the data web. In: SIGMOD. pp. 1135–1138. ACM (2009)
22. Wille, R.: Restructuring lattice theory: An approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets. pp. 445–470 (1982)
23. Wille, R.: Conceptual graphs and formal concept analysis. In: Lukose, D., Delugach, H., Keeler, M., Searle, L., Sowa, J. (eds.) Conceptual Structures: Fulfilling Peirce's Dream, pp. 290–303. Springer (1997)