

Extracting Relevant Questions to an RDF Dataset Using Formal Concept Analysis

Mathieu d'Aquin
Knowledge Media Institute
The Open University, Milton Keynes, UK
m.daquin@open.ac.uk

Enrico Motta
Knowledge Media Institute
The Open University, Milton Keynes, UK
e.motta@open.ac.uk

ABSTRACT

With the rise of linked data, more and more semantically described information is being published online according to the principles and technologies of the Semantic Web (especially, RDF and SPARQL). The use of such standard technologies means that this data should be exploitable, integrable and reusable straight away. However, once a potentially interesting dataset has been discovered, significant efforts are currently required in order to understand its schema, its content, the way to query it and what it can answer. In this paper, we propose a method and a tool to automatically discover questions that can be answered by an RDF dataset. We use formal concept analysis to build a hierarchy of meaningful sets of entities from a dataset. These sets of entities represent answers, which common characteristics represent the clauses of the corresponding questions. This hierarchy can then be used as a querying interface, proposing questions of varying levels of granularity and specificity to the user. A major issue is however that thousands of questions can be included in this hierarchy. Based on an empirical analysis and using metrics inspired both from formal concept analysis and from ontology summarisation, we devise an approach for identifying relevant questions to act as a starting point to the navigation in the question hierarchy.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

General Terms

Algorithms, Design, Human Factors, Measurement

Keywords

Semantic data, semantic web, RDF, navigation, question, formal concept analysis

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

K-CAP'11, June 26–29, 2011, Banff, Alberta, Canada.

Copyright 2011 ACM 978-1-4503-0396-5/11/06 ...\$10.00.

1. INTRODUCTION

The idea of a Semantic Web is quickly gaining momentum as more and more organisations are exposing their data in structured, semantically described datasets following the principles of linked data [2]. When coming across such a dataset, a significant effort is generally required before it can be exploited. A variety of approaches can be envisaged to become familiar with the content and the structure of such a dataset, including inspecting its schema (i.e., the ontology) with an ontology editor (such as Protégé¹ or the NeOn Toolkit²), using a graph representation, a faceted browser, or sending test queries in a trial and error approach.

To simplify this process, example queries are often used as a way to characterise a dataset (see, e.g., the use of competency questions in ontology engineering [11]). By providing a simple representation of the kind of answers a dataset can provide, they help in better understanding what is the scope of the dataset, and how it can be used. In addition, as questions can be formulated in a way close to natural language, such an approach has the advantage of supporting users unfamiliar with the underlying technologies (e.g., the RDF³ and OWL⁴ representation languages, and the SPARQL⁵ query language), providing easy access points to the dataset.

In this paper, we propose an approach based on automatically extracting a set of questions that can be answered by a dataset. We use formal concept analysis (FCA) to identify sets of objects from a dataset that share common properties. Each of these sets represents the answers to a particular question, which is characterised by the properties shared by the elements of the set. One of the advantages of using FCA is that these sets are organised in a hierarchy (a lattice), relating any extracted question with more general and more specific ones. This hierarchy is used to generate a navigational query interface, allowing the user to browse the set of possible questions to a dataset, together with their answers.

However, one of the main drawbacks of this method is that the application of FCA can generate thousands of questions, making browsing the hierarchy cumbersome. We therefore also study a set of measures, inspired both from ontology summarisation and from FCA, to identify the questions which are more likely to be close to the ones of interest to the user.

¹<http://protege.stanford.edu/>

²<http://neon-toolkit.org>

³<http://www.w3.org/RDF/>

⁴<http://www.w3.org/TR/owl-ref/>

⁵<http://www.w3.org/TR/rdf-sparql-query/>

Through studying questions proposed by human users of three test datasets, we propose a combination of measures which help identifying a reasonable entry point into the generated question hierarchy, and so to the dataset.

2. FORMAL CONCEPT ANALYSIS

FCA [6, 15] is a formal, generic framework, generally associated with the fields of data mining and knowledge discovery. In broad terms, it is concerned with identifying from raw data, patterns of objects’ characteristics that form formal *concepts*.⁶ Such a concept is characterised both by an intent – i.e., a set of attributes, and an extent – i.e., the set of objects in the data that share these attributes.

More formally, FCA relies on the notion of a formal *context*, which represents the raw data. A formal context $C = (G, M, I)$ is made of a set of objects G , a set of attributes M and a binary relation $I \subseteq G \times M$. In simpler terms, a formal context is a binary matrix where the rows represent objects, and columns represent attributes of these objects. Given O a set of objects of G , we note O' the set of attributes of M which are shared by all the objects of O . In the same way, given $A \subseteq M$, $A' \subseteq G$ is the set of objects that share all the attributes in A . The double application of $(\cdot)'$ is said to represent the closure of a set of objects or attributes. In other terms, O'' and A'' are said to be *closed*.

A formal concept of a context $C = (G, M, I)$ is characterised by a pair (O, A) , where $O \subseteq G$ and $A \subseteq M$. O is called the *extent* and represents the objects that share the attributes of A , i.e., $O = A'$. A is called the *intent* and represents the attributes that are shared by the objects of O , i.e., $A = O'$. Note that this implies that $O = O''$ and $A = A''$, i.e., the concept (O, A) is equivalently defined both by its set of objects, and by its set of attributes.

The set of all concepts that can be derived from a formal context form a lattice, relying on the *subconcept* relation (denoted by \leq). Indeed, we say that a concept (O_1, A_1) is a subconcept of another concept (O_2, A_2) – i.e., $(O_1, A_1) \leq (O_2, A_2)$ – if $O_1 \subseteq O_2$ and (equivalently) $A_2 \subseteq A_1$. This *concept lattice* has an upper-bound and a lower-bound (which are often the concept with an empty extent and the one with an empty intent respectively).

3. BUILDING A CONCEPT LATTICE TO IDENTIFY QUESTIONS IN A DATASET

Our goal here is to extract from a dataset, represented in RDF, a set of questions it can answer. We start by introducing simple notations for describing an RDF dataset. We illustrate these notations, as well as most of the other examples in the article, using the FOAF⁷ profile of Tom Heath on the Knowledge Media Institute website (<http://kmi.open.ac.uk/people/tom/rdf>).

In such a dataset, we essentially focus on instances. Instances represent individual objects that are members of classes. For example, `tom` is an instance of the class `Person`. This is represented in RDF through the use of the property `rdf:type`, but we use here the simplified notation

`Person(tom)`.⁸ Instances such as `tom` can have properties linking them to other instances (e.g., to represent the fact that Tom knows Enrico Motta) or to literal values (e.g., to represent the fact that Tom’s phone number is “+44-(0)1908-653565”). Such assertions are occurrences of binary relations, presented in our simplified notation as `knows(tom, enrico-motta)` and `phone(tom, "+44-(0)1908-653565")` respectively.

The classes such as `Person` and the properties such as `knows` come from the ontology(ies) used in the dataset, where they are part of a taxonomy: for example, `Person` can be a subclass of another class `Agent` and `knows` can be a sub-property of a property `hasMet`. We represent this with the notation `Person \sqsubseteq Agent` and `knows \sqsubseteq hasMet` respectively. Such taxonomic relationships can be either asserted in the dataset, or inferred from the definitions of the classes and properties.

3.1 Assumptions and Requirements

We focus here on questions for which the answers are sets of objects, such as “*Who does Tom know?*”. More precisely, we consider questions corresponding to queries to the dataset for which results are set of instances (e.g., all the people that Tom knows). A question itself is characterised by a set of properties that are common to the elements of the answer. In this sense, it can be related to a conjunctive query – e.g., “*Who knows Tom?*” corresponds to the query `Person(?x) \wedge knows(?x, tom)`.

In addition, we assume that a significant question to a dataset should have more than one answer, as querying for the common characteristics of a unique object does not appear relevant. Also, questions complying with our requirements should be related with each other in a hierarchy. For example, it is natural to consider that “*What are the things that Tom knows?*” is more general than “*What are the people that Tom knows?*” (i.e., “*Who does Tom know?*”), or that “*Who has Tom met?*” is more general than “*Who does Tom know?*”. In other words, a question is more general than another if it includes its set of answers.

3.2 Building the Formal Context

The basic idea underlying the technique presented here is relatively straightforward: We want to build a concept lattice where each concept represents a question, with the extent being a set of instances from the dataset corresponding to answers, and the intent the common characteristics of these instances, forming the clauses of the question. We therefore need to build a formal context $C = (G, M, I)$ where G is the set of all instances of the dataset, and M corresponds to all the possible characteristics of these instances.

We consider three types of attributes that can be applied to an instance o of the dataset. Attributes of the form `Class::C` appear if o is an instance of C (i.e., $C(o)$). Attributes of the form `p:m` appear if o is related through the property p to the instance or the literal value m (i.e., $p(o, m)$). Attributes of the form `p:m` are used if the instance m is related to o through the property p (i.e., $p(m, o)$).

In order to extend this *explicit* set of attributes with inferred statements, we also generate additional attributes substituting the classes, properties, and individuals in existing ones with all the possible combinations of superclasses,

⁶To avoid ambiguities in this paper, we use the term *concept* to refer to the notion of formal concept in FCA, and the term *class* to refer to the corresponding entities in ontologies

⁷<http://www.foaf-project.org/>

⁸In this simplified notation, we use the local ID or label of an entity, instance, property or class, instead of its full URI.

superproperties and types that can be inferred. For example, we extend *Class::Person* and *knows:-Enrico-Motta* with inferred attributes of the form *Class::Agent, hasMet:-Enrico-Motta, knows:-Person* and *hasMet:-Person*.

Having built the set of all attributes for all the instances of the dataset, we can now build the matrix relating these attributes to these instances/objects, as a formal context for FCA.

3.3 Creating the Lattice and Eliminating Redundancies

One parameter of a concept lattice building tool is the minimum *support* for a concept to be included in the lattice, i.e., the minimum cardinality of its extent. In accordance with our assumptions and requirements (Section 3.1), we used 2 as minimum support.

An example lattice for the dataset <http://kmi.open.ac.uk/people/tom/rdf> is presented in Figure 1(a). Five concepts are present in the hierarchy, with the top one representing all the objects of the dataset and therefore, all the instances of the class *Thing*, the class of everything in OWL. It can also be noticed in this example that significant parts of the elements characterising some of the concepts are redundant and therefore not really useful. Indeed, having both the attributes *tom:knows* and *Person:knows*, or *Class::Thing* and *Class::Person* is not useful as one of the attributes can be inferred from the other. Checking such a relationship between attributes, we reduce the definition of the intents of concepts to keep only the non-redundant ones as shown in Figure 1(b).

3.4 Using the Concept Lattice as a Query Interface

As mentioned before, the basic idea of our approach is that each concept of the concept lattice represents a question, with its intent being the components of the question, and its extent the answers. The goal is to use this lattice as the basis for a navigational interface to query the underlying dataset. The first step is therefore to provide a simple representation for each concept/question, which would be reasonably readable by a human user. We derive such a representation from the (non redundant) intent of a concept as a ‘question’ in pseudo-natural language, following the template:

What are the $\{C_1, \dots, C_n\}$ that $\{p_1 m_1, \dots, p_m m_m\}$
and that $\{n_1 q_1, \dots, n_t q_t\}$

where $\{C_1, \dots, C_n\}$ are extracted from attributes of the form *Class::C₁, ..., Class::C_n*, $\{p_1 m_1, \dots, p_m m_m\}$ are extracted from attributes of the form $\{p_1:-m_1, \dots, p_m:-m_m\}$ and $\{n_1 q_1, \dots, n_t q_t\}$ are extracted from attributes of the form $\{q_1:-n_1, \dots, q_t:-n_t\}$. We also adapt this general structure depending on whether or not one of the attribute sets is empty and the names of classes, properties and individuals are reduced to the local fragment of their URI, or to the label of the entity if available. For example, the concept at the bottom left of the lattice in Figure 1(b) is transformed into the question:

What are the (Person) that (tom knows)

Interpreting concepts as questions in this way means that the obtained lattice represents a complete hierarchy of questions that can be presented to the user as a query interface

to the considered dataset (see the example Figure 2 where the question “*What are the (Person)?*” has been selected, showing the sub-questions, the super-questions, alternative questions about Tom’s projects and interests, as well as the answers to the selected question).

However, while on our toy example the results are simple and easy to navigate, on a bigger dataset, this process can result in thousands of questions being generated. In the next section, we therefore investigate measures that can be used to identify a set of questions more likely to be of interest to a user, as a way to generate a reasonable entry point into a large question hierarchy.

4. MEASURING THE RELEVANCE AND INTERESTINGNESS OF A QUESTION

In order to identify approaches to find a set of questions more likely to be of interest to a user, we take inspiration from the works prominent in two areas: ontology summarisation and concept lattice simplification.

4.1 Measures Inspired from Ontology Summarisation

In [12], we presented a work on extracting the key classes of a (possibly populated) ontology, based on a variety of different metrics taking into account in particular the “topology” of the ontology, its structure, and external elements such as the popularity of a concept. We look at three of these criteria which appear specially relevant:

Coverage. In ontology summarisation, this criterion intends to take into account the fact that a good summary should contain elements from all the significant parts of the ontology. This also appears important here, as we would expect any point of the lattice to be reachable from the identified questions, acting as entry points to the hierarchy. We define the set of questions reachable from another question using the notions of filter and ideal from FCA (or more generally, from lattice theory [1]). The ideal of a concept is the set of all its direct or indirect subconcepts, or in other terms, all the concepts linked through the transitive closure of the \leq relation. Similarly, the filter of a concept corresponds to the set of concepts that are reachable through the superconcept relation. We define the coverage of a question in our question hierarchy as the union of the filter and ideal of the corresponding concept, and indicate that a set of questions covers the dataset when the union of the coverages of its elements correspond to the entire lattice.

Level. To extract key classes from an ontology, one of the ideas is that key classes are never too general or too specific, but can be found in the middle layer of the hierarchy. A similar idea can be applied here, as a we can expect very general or very specific questions not to be the most useful to the user. We use a measure of the level of the question/concept of the hierarchy as the distance between the question/concept and the root concept.

Density. In ontology summarisation, one of the assumptions is that the richer the representation of a class is (i.e., the denser it is in terms of the properties attached

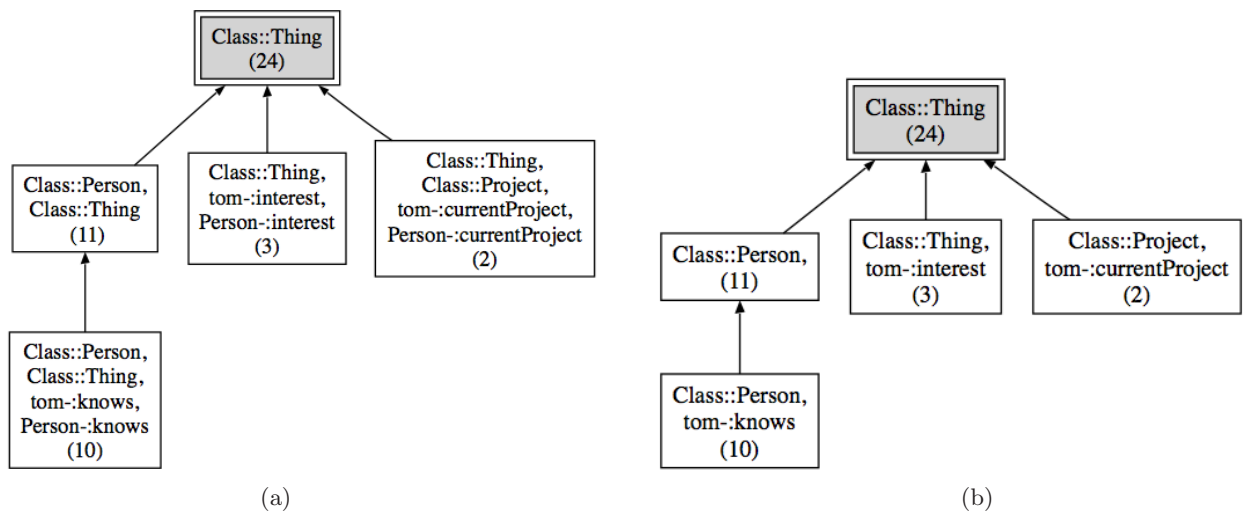


Figure 1: Concept lattice generated from <http://kmi.open.ac.uk/people/tom/rdf> (a); with redundancy eliminated (b). In both figures, the bottom concept of the lattice has been omitted.

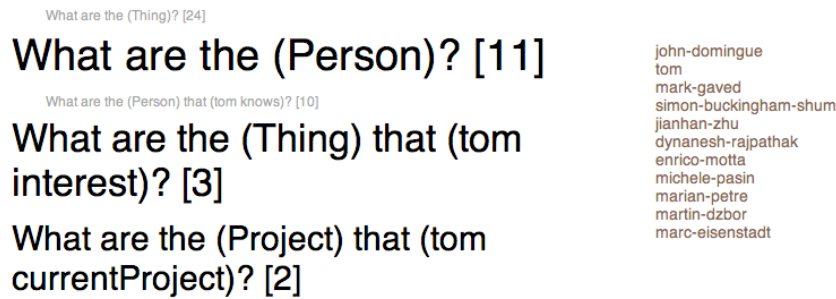


Figure 2: Simple example of the concept lattice-based interface to querying a dataset.

to it) the more likely it is to be important. Here, the situation is slightly different, as the related notion of density (i.e., the number of attributes in the intent of the concept) is closely related to the one of level (the more specific a concept is, the more likely it is to have a large number of attributes). Therefore, we cannot assume density to be used as criterion to maximise, but rather as a metric which should be not too high (useful questions are probably not the most complex ones), but also not too low (useful questions should be sufficiently well defined).

4.2 Measures Inspired from FCA

There are only a few metrics that have been devised in FCA to try to identify the most “interesting” concepts. The most common one is the support, but a notion of stability, applicable both to the intent and the extent of a concept, has been recently discussed as a way to reduce a concept lattice, providing a possible measure to be applied to our problem.

Support. As already mentioned, the support of a concept is the cardinality of its extent, i.e., the number of objects it represents (and so the number of answers to the

question). As noticed in [10] to motivate the stability measure, there are many scenarios where the most interesting concepts might not be the ones representing the largest number of objects. Here as well, an interesting question might be one with few answers, while questions with a large number of answers might be meaningless.

Intensional Stability. Intensional stability as described in [10] intends to define a stable concept as one whose “*intent does not depend much on each particular object of the extent*”. Given a concept (O, A) where O is the extent and A is the intent, the degree of intensional stability σ^i is defined by $\sigma^i(O, A) = \frac{|\{C \subseteq O \mid C' = A\}|}{2^{|A|}}$. However, as explained in [10], computing such a measure is complex. We therefore use an approximation, which corresponds to the ratio between the cardinality of the concept’s extent and the one of the smallest of its direct super-concepts, i.e., $\sigma_{ap}^i(O, A) = \frac{|O|}{\min_{n \in N}(|n|)}$, where N is the set of extents of the direct super-concepts of (O, A) in the lattice.

Extensional Stability. In a similar way as for intensional stability, extensional stability can be defined at the

level to which the extent of a concept depends on a particular attribute of the intent. It is defined as $\sigma^e(O, A) = \frac{|C \subseteq A | C' = O|}{2|O|}$ and we use the following approximation: $\sigma_{ap}^e(O, A) = \frac{|A|}{\min_{b \in B}(|b|)}$ where B is the set of intents of the direct subconcepts of (O, A) in the lattice.

5. EXPERIMENT

While the previous section discusses measures that can be used to assess the potential “interestingness” of a question generated using our FCA-based method, we present here a user-based experiment to find out which of these measures are the most relevant and how to parametrize them. We asked 12 users with various degrees of familiarity with semantic technologies to inspect a reasonably large dataset and express up to 5 questions they believed to be interesting on this dataset.

5.1 Datasets

We used 4 different datasets as testbeds for our experiment. Three of them, called *geography*, *jobs* and *restaurants* were created by the University of Texas, Austin [14], and later transformed into OWL/RDF [7] for the purpose of evaluating a query answering system. The other one (*drama*) concerns modern productions of classical greek drama and was built locally for the needs of a project in the domain of Arts. Two of the evaluators for this dataset are actually domain experts involved with the data, with no background in semantic technologies. For each of the datasets, we constructed the concept lattice as explained earlier in this paper. Information about each dataset and the corresponding lattices is given in Table 1.

Table 1: Summary of the test datasets.

Dataset	Nb. Instances	Nb. Concepts
<i>geography</i>	715	842
<i>jobs</i>	4142	66284
<i>restaurants</i>	9746	6810
<i>drama</i>	19294	10083

5.2 Results

Out of the 44 valid questions we obtained,⁹ we tested that 27 (61%) matched the format of questions produced by our method, and therefore corresponded to questions/concepts in the generated lattices. In the questions that could not be represented we found several reasons why they diverged from our model, which could be considered as possible extensions in the future, including for example the use of disjunctive clauses (e.g., “Which Greek plays have been performed in Kenneth McLeish’s translations or versions?”) or of numerical manipulations/tests on values (e.g., “What are the restaurants that have ratings higher than 2.5?”).

The resulting set of user-generated questions represent a useful sample to analyse the range of values taken by the different measures to be considered. Coverage is not evaluated

⁹Some of the questions given by users could not be answered by a set of instances in the considered dataset. There was no overlap between the questions proposed by different users.

here as it cannot be assessed at the level of an individual question. Since our goal is to obtain a set of questions as an entry points to the hierarchy, we consider coverage as a fundamental criterion to be enforced while generating the initial question set.

Level. Amongst all the valid and representable questions given by evaluators, the average level of a question in its lattice is 4.46. This is slightly higher than the average level of all the concepts in the lattices and it is also worth mentioning that none of the questions were at a level lower than 3 or higher than 7. This validates our hypothesis that questions of interest are generally not the most general, or the most specific, but are located within a small range around the “centre” of the lattice, which corresponds to the average level. We therefore define a normalised metric m_l for a concept of our question hierarchy which is computed as the distance between the concept’s level and the average level in the lattice.

Density. In the case of the density measure, our initial hypothesis was also verified that interesting questions tend to be defined simply, but with sufficient elements to represent a distinct set of answers. Indeed the average density of the valid, representable questions is 2.14, and the measure is always included in the range [1.3] (most of the questions being of density 2, such as “What are the restaurants in San Francisco?”). We can argue that there is a strong relationship between the level of a concept and the density of the question. However, this highly depends on the structure of the original dataset, as for example, “What are the restaurants in a city?” is more general than the previous question, while also being of density 2. We therefore define the normalised metric m_d based on the difference between the density of a given concept and 2, which seems to be the “standard” for simple, but sufficiently defined questions.

Support. Depending on the dataset, the support of the provided questions can vary a lot. For example, questions in the *jobs* dataset tend to have a lot of answers (up to 3402), while in the *drama* dataset, they are generally smaller (from 2 to 38).

Intensional Stability. The expectation related to intensional stability is that the more stable a concept is, the more it is supposed to represent a significant and distinct set of individuals. However from our experiment, there does not seem to be any correlation between a question being identified by evaluators as interesting, and its intensional stability. Values can vary from very low (0.0008) to high (0.82) even for questions provided by a single user, regarding a single dataset.

Extensional Stability. Surprisingly, contrary to intensional stability, the values of extensional stability appear very stable, especially within one dataset, and always high (between 0.75 and 1.0), in particular if compared with the average in the dataset (around 0.4 for all of them). Indeed, it appears that the definition of the question as being a significant subset of the elements of more specific questions is an important criterion to identify interesting questions.

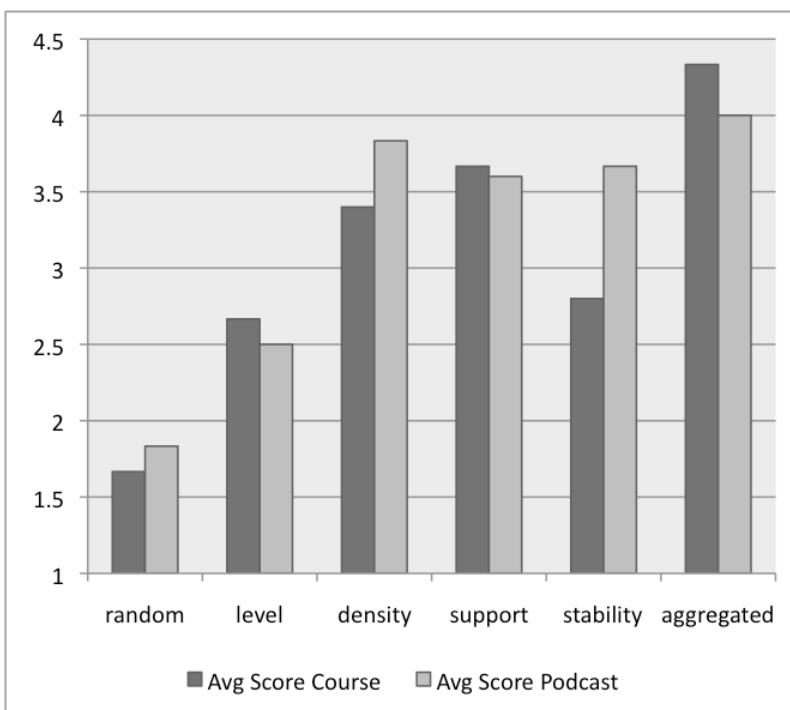


Figure 3: Average scores for the six tested measures on the *Course* and *Podcast* datasets.

Considering the results discussed above, we define an aggregated metric to rank questions in a question hierarchy as the linear combination of the 3 metrics m_l , m_d and σ_{ap}^e on concepts:

$$m(O, A) = w_1.m_l(O, A) + w_2.m_d(O, A) + w_3.\sigma_{ap}^e(O, A)$$

What constitutes an interesting question to a dataset is very dependent on the user and the context in which the answer would be used, but based on the results obtained above, we expect such a metric to be adequate in supporting the identification of reasonable entry points to the question hierarchy.

6. IMPLEMENTATION AND VALIDATION

The overall method presented here can be divided in two separate processes: 1- the creation of the non-redundant hierarchy of question, and 2- the generation of the user interface based on this hierarchy and using the metric defined above. For the first process, we developed a program that generates a formal context in the input format of the lattice generation tool from a SPARQL endpoint, according to the method described in Section 3.2. We use the OWLIM triple store,¹⁰ which partially supports OWL/RDF inferences. We use the implementation of the CHARM algorithm provided by the CORON tool [8] to generate the concept lattice.

We devised a simple algorithm in order to identify in the generated lattice a set of questions that both rank high according to a chosen metric¹¹ and which, all together, maximally cover the lattice (by choosing questions which are

¹⁰<http://www.ontotext.com/owlim/>

¹¹A special case is considered for the support measure regarding the top – i.e., the root – of the hierarchy. It is indeed always the concept with the highest support, and is therefore excluded from the ranking.

not in the same branch as an already chosen question). In order to validate the choice of the aggregated measure defined above, we tested it together with 5 other measures in 2 different datasets. We used as datasets the collection of 614 course descriptions and 1706 video podcasts from <http://data.open.ac.uk>. We generated the list of entry questions for each of these datasets using each of the following metrics: a random metric, m_l (level) alone, m_d (density) alone, support, σ_{ap}^e (stability) alone and the aggregated measure of level, density and stability, using a naive distribution of weights (i.e., $w_1 = w_2 = w_3 = \frac{1}{3}$). We then asked six different users to give a score between 1 and 5 to each of the sets of questions (presented in a random order), 5 corresponding to the highest level of interest.

The results are presented in Figure 3. As can be seen, the aggregated measure appears to provide significantly better results than all of the other measures on both datasets, especially compared to the random measure. As expected, the level, density and stability of questions all contribute to identifying interesting questions (to different extents), but are more appropriate when used in combination. More surprisingly, the support (i.e., number of answers) provide better results than could have been expected from our experiment. A possible explanation is that, in datasets where objects are described homogeneously (i.e., they all have more or less the same structure), support is highly correlated with the measures related to the question’s level and density.

The application of the querying interface has already been shown in Figure 2 on our toy example. Figure 4 gives another example, based on the *restaurant* dataset. It is generated from the concept lattice using as initial questions the set computed using the aggregated measure. The value of this measure is represented in the interface by the font size

What are the (Restaurant) that (ratingString good)? [2761]

What are the (Restaurant) that (ratingString good,isInCity City)? [2759]

What are the (Restaurant) that (foodType indian,ratingString good)? [61]

What are the (Restaurant) that (rating 3.6,ratingString good)? [84]

What are the (Thing) that (isIn Region)? [187]

What are the (Thing) that (City isIn)? [29]

What are the (Restaurant) that (isInCity City)? [9547]

What are the (Restaurant) that (foodType indian)? [104]

What are the (Thing) that (label el sobrante)? [3]

What are the (Thing) that (label monterey)? [2]

```
ID_theAcornRestaurant6437
ID_durantGardenRestaurantClassical1612
ID_tonySSeafoodRestaurant5599
ID_springGardenChineseRestaura6466
ID_thaiGardenRestaurant1908
ID_mainStCoffeeRoasting8201
ID_prima4773
ID_paloAltoCoffeeRoastingCo2795
ID_yuenYung3294
ID_villaCoffeeShop4698
ID_warehouseCafe3897
ID_specialtySCafeAndBakery9428
ID_pizzaRustaciCafeLtd7258
ID_tora-YaRestaurant1630
ID_rosita1468
ID_buckeyeRoadhouse4338
ID_beppo5555
ID_amiciSEastCoastPizzeria1910
ID_tajMahalIndianCuisine1750
ID_sushiMainStreet5114
ID_originalJoeSNo2Restaurant5725
ID_doubleRainbowGourmetIceCream1744
ID_leftAtAlbuquerque7968
ID_apolloPizza2497
ID_emeraldGardenRestaurant4290
ID_jackSSteakhouse7951
ID_villaRomanoRestaurant1565
ID_mediterraneanCafe2751
ID_guadalajarasSuperBurrito5299
ID_okawa669
ID_kitahama9021
```

Figure 4: Example of application of the lattice-based query interface on the *restaurant* dataset.

used for the question. The first question is also attached to the questions directly more general and directly more specific. Any question displayed can be selected, and will then be re-displayed at the top of the list, with more general and more specific questions, as well as the set of its answers (for example, in Figure 4, the question “*What are the (Restaurant) that (ratingString good)*” has been selected).

7. RELATED WORK

As discussed previously, our approach relates to ontology summarisation, where an abstract summary of a supposedly complex ontology is being produced, for example in the form of a set of important concepts [12]. In [16], the authors propose a technique to extract a sub-graph of an RDF graph to act as a summary. Similar ideas have also been recently applied to large RDF datasets, including for example the ExpLOD tool [9] which produces a visual representation of a dataset, clustering (and therefore abstracting) elements together to produce an overview graph of the dataset. As can be seen from these initial works, the idea of summarising datasets and ontologies is only starting to gain attention. While providing example queries is generally seen as an efficient way for somebody to quickly understand a dataset or an ontology,¹² to the best of our knowledge, there have not been any attempt before at summarising a dataset by providing sets of automatically extracted questions.

FCA, and especially concept lattices, have been used in several approaches to support the task of browsing structured datasets. For example, in the context of image search, [3] makes use of several lattices, representing different aspects of images (shape, luminance and “semantic content”, with the “semantic content” aspect being based on an ontology). In

¹²The system SchemaPedia (<http://schemapedia.com/>) for example gives manually created example queries for the ontologies it collects.

this case, the lattices act as support for browsing the results of a search. In [4], the author develops a similar idea, using a concept lattice built from the metadata attached to documents, but makes use of *logical concept analysis*, a variant of FCA where the attributes are logical properties, partially ordered by a relation of subsumption. Other works have been devised that generate views on populated ontologies, which correspond to formal concepts that can be visualised as concept lattices, and be defined by users [13]. A significant difference between these approaches and ours is that we focus on providing an overview of a dataset using the set of questions it can answer, and a navigation mechanism allowing to browse these questions, rather than the data itself. Closer to our work in that sense is the recent paper [5]. Indeed, in this work, the author relies on the principles of FCA to provide a navigation mechanism based on queries to the underlying dataset. While here, we focus on providing a navigation interface to the data (through the questions), targeting users unfamiliar with both the data and the underlying technologies, [5] concentrate on obtaining queries exploiting the high expressivity of the underlying language (close to the SPARQL language). Our approach also includes, as a core mechanism, the ability to identify sets of questions more likely to provide useful entry points to the dataset.

8. CONCLUSION

In this paper, we have presented an approach to generate a hierarchy of questions that could be asked to a dataset using formal concept analysis, and to derive a navigational query interface to this dataset based on this hierarchy. This approach relies on constructing a concept lattice from the description of the instances of the dataset, creating groups (concepts) of instances having common properties. However, for a large dataset, the number of these concepts (and

so of the corresponding questions) can be very large. We also study the measures providing indications of the potential interest and relevance of an extracted question. Our experiment shows that the identified measures provide a good base to select a set of reasonably interesting questions to act as an entry point into the dataset.

One of the most obvious drawbacks of the approach presented here is the complexity of the methods deployed, especially ontological reasoning and concept lattice generation. Since these complex methods only need to be used once per dataset, in an offline process, the few minutes they take on our test datasets cannot be considered a strong limitation. However, some level of approximation and optimisation will have to be applied to use our tools on significantly larger datasets, containing several millions of statements.

There are many extensions that can be considered to the presented approach. Indeed, our experiment also identified some of the most common limitations of our model in terms of the expressiveness of the considered questions. Some of these elements, such as the possibility to add tests on numerical values, could be added to the model. They would however increase significantly the size of the lattice, and therefore the overall complexity of the approach.

Also, while the approach has been shown to provide promising results on self-contained datasets, an interesting future work would be to take into account elements derived from links to external datasets, making it possible to explore the questions that can be answered from integrating multiple sources of data.

Acknowledgment

The authors would like to thank Amedeo Napoli, Mehdi Kaytoue-Uberall and Laszlo Szathmary for their help with the use of the CORON system and for the pointers to elements of formal concept analysis. We also thank Vanessa Lopez for telling us about some of the datasets included in our experiments, as well as all the evaluators who responded to our questionnaire.

9. REFERENCES

- [1] G. Birkhoff. *Lattice Theory, 3rd ed.* Providence, RI: Amer. Math. Soc., 1967.
- [2] C. Bizer, T. Heath, and T. Berners-Lee. Linked data – the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.
- [3] J. Ducrou, P. Eklund, and T. Wilson. An intelligent user interface for browsing and searching MPEG-7 images using concept lattices. In S. Yahia, E. Nguifo, and R. Belohlavek, editors, *Concept Lattices and Their Applications*, volume 4923 of *Lecture Notes in Computer Science*, pages 1–21. Springer Berlin / Heidelberg, 2008.
- [4] S. Ferre. Camelis: a logical information system to organize and browse a collection of documents. *Int. J. General Systems*, (38), 2009.
- [5] S. Ferre. Conceptual navigation in RDF graphs with SPARQL-like queries. In L. Kwuida and B. Sertkaya, editors, *Formal Concept Analysis*, volume 5986 of *Lecture Notes in Computer Science*, pages 193–208. Springer Berlin / Heidelberg, 2010.
- [6] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999.
- [7] E. Kaufmann. Talking to the semantic web - natural language query interfaces for casual end-users. PhD thesis. University of Zurich, Switzerland, 2009.
- [8] M. Kaytoue, F. Marcuola, A. Napoli, L. Szathmary, and J. Villerd. The Coron System. In *8th International Conference on Formal Concept Analysis (ICFCA) - Supplementary Proceedings*, 2010.
- [9] S. Khatchadourian and M. P. Consens. Exploring RDF usage and interlinking in the linked open data cloud using ExpLOD. In *Linked Data On the Web workshop, LDOW*, 2010.
- [10] S. Kuznetsov, S. Obiedkov, and C. Roth. Reducing the representation complexity of lattice-based taxonomies. In *Conceptual Structures: Knowledge Architectures for Smart Applications. 5th International Conference on Conceptual Structures, ICCS*, 2007.
- [11] N. Noy and D. L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880.
- [12] S. Peroni, E. Motta, and M. d’Aquin. Identifying key concepts in an ontology through the integration of cognitive principles with statistical and topological measures. In *Third Asian Semantic Web Conference*, 2009.
- [13] J. Tane, P. Cimiano, and P. Hitzler. Query-based multicontexts for knowledge base browsing: An evaluation. In *Conceptual Structures: Inspiration and Application*, volume 4068 of *Lecture Notes in Computer Science*, pages 413–426. Springer Berlin / Heidelberg, 2006.
- [14] L. R. Tang and R. J. Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In *12th European Conference on Machine Learning, ECML*, 2001.
- [15] R. Wille. Concept lattices and conceptual knowledge systems. *Computers & Mathematics with Applications*, 23, 1992.
- [16] X. Zhang, G. Cheng, and Y. Qu. Ontology summarization based on rdf sentence graph. In *World Wide Web Conference*, 2007.