ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220392375

Finding association rules in Semantic Web Data

Article *in* Knowledge-Based Systems · February 2012

DOI: 10.1016/j.knosys.2011.05.009 · Source: DBLP

CITATIONS
34

READS
369

2 authors:

Victoria Nebot
Universitat Jaume I
32 PUBLICATIONS **297** CITATIONS
SEE PROFILE

Rafael Berlanga-Llavori
Universitat Jaume I
141 PUBLICATIONS **1,210** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

SLOD-BI: Development of an open linked data infrastructure for Social Business Intelligence aimed at SMEs View project

All content following this page was uploaded by Victoria Nebot on 25 July 2014.

The user has requested enhancement of the downloaded file. All in-text references underlined in blue are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

Knowledge-Based Systems 25 (2012) 51–62

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Finding association rules in semantic web data

Victoria Nebot *, Rafael Berlanga

Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I, Campus de Riu Sec, 12071 Castellón, Spain

ARTICLE INFO

Article history:
Available online 26 May 2011

Keywords:
Semantic web
Data mining
Association rules
Semantic annotation
Biomedical application

ABSTRACT

The amount of ontologies and semantic annotations available on the Web is constantly growing. This new type of complex and heterogeneous graph-structured data raises new challenges for the data mining community. In this paper, we present a novel method for mining association rules from semantic instance data repositories expressed in RDF/(S) and OWL. We take advantage of the schema-level (i.e. *Tbox*) knowledge encoded in the ontology to derive appropriate transactions which will later feed traditional association rules algorithms. This process is guided by the analyst requirements, expressed in the form of query patterns. Initial experiments performed on semantic data of a biomedical application show the usefulness and efficiency of the approach.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

In the past few years, there has been an increasing interest in combining the two research areas semantic web (SW) and data mining (DM) [1,33]. Thanks to the standardization of the ontology languages RDF/(S)[1] and OWL,[2] the SW has been realized and the amount of available semantic annotations is ever increasing. This is due in part to the active research concerned about learning knowledge structures from textual data, usually referred as Ontology Learning [9]. On the other hand, background knowledge has been also used to improve the results of Web mining. However, little work has been directed towards mining SW data itself, that is, mining the SW. We strongly believe that mining SW data will bring much benefit to many domain-specific research communities where relevant data are often complex and heterogeneous, and a large body of knowledge is available in the form of ontologies and semantic annotations. This is the case of the clinical and biomedical scenarios, where applications often have to deal with large volumes of complex data sets with different structure and semantics. In this paper, we investigate how ontological instances expressed in OWL can be combined into transactions in order to be processed by traditional association rules algorithms [2], and how we can exploit the rich knowledge encoded in the respective ontologies to reduce the search space.

Machine learning algorithms have been successfully applied to large data sets to extract useful knowledge by searching for interesting patterns (e.g., association rules). However, the nature of semantic data is quite different from that of traditional data sets treated by these algorithms. Thus, the main challenges we face in this work are the following ones:

- Traditional DM algorithms deal with homogeneous data sets composed by transactions, where each transaction is represented by a subset of items. In contrast, in a repository of semantic annotations expressed in OWL we keep ontology axioms, describing the conceptual domain, and the semantic annotations are represented as assertions relating instances through properties that are consistent with the ontology. The usual way to represent these assertions is as triples (*subject*,*predicate*,*object*). In this scenario, the identification of transactions and items is not trivial. Items may correspond to either instances or literals, and a transaction is defined according to the user requirements as a subset of items semantically related in the repository.
- OWL is sustained by description logics (DLs) [6], which are knowledge representation formalisms with well-understood formal properties and semantics. Therefore, annotated data does not follow a rigid structure. That is, instances belonging to the same OWL class may have different structures, giving place to structural heterogeneity issues.
- Since DLs are defined with formal semantics, reasoning capabilities must be applied in order to handle the implicit knowledge.

As far as we know, the presented approach is the first attempt to find association rules directly from SW data. Previous work on mining SW data has been mainly focused on clustering and classifying ontology instances [8,12]. However, the representation techniques required in association mining are quite different from clustering and classification tasks. Association rules are based on

* Corresponding author. Tel.: +34 964 72 83 67; fax: +34 964 72 84 35.
E-mail addresses: romerom@lsi.uji.es (V. Nebot), berlanga@lsi.uji.es (R. Berlanga).

[1] RDF/(S): http://www.w3.org/TR/rdf-concepts/,rdf-schema/, '04.
[2] OWL: http://www.w3.org/TR/owl-features/, '04.

0950-7051/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2011.05.009

the notion of transaction, which is an observation of the co-occurrence of a set of items. This is basically a set-based representation of the world which contrasts with the numerical vector-based representations used in clustering and classification. When dealing with SW data, the main challenge consists in identifying interesting transactions and items from the semi-structured and heterogeneous nature of these data. In this way, it becomes crucial to use as much as possible the knowledge provided by the ontologies so that transactions can be easily defined and mined. As we will show along this paper, there is a great variety of ways to generate items and transactions from semantic data, which depend on the detail level and structural semantics the analyst wants to consider.

The main contribution of this paper is twofold: (1) we define the problem of transaction generation and mining association rules from SW data, and (2) we propose a method to efficiently extract items and transactions suited for traditional association rules mining algorithms. This work extends that presented in [28] in the following aspects: a more elaborate and complete formalism is presented, we have updated the related work, and an exhaustive evaluation over a real scenario has been carried out.

The rest of the paper is organized as follows. Section 2 gives an overview of the related work. Section 3 explains the basics of the two integrated technologies, association rules mining and OWL DL ontologies and motivates the problem with a running example. Section 4 contains the general methodology and foundations of the approach. Section 5 shows the experimental evaluation and Section 6 gives some conclusions and future work.

2. Related work

Most research on DM for semantic data is based on inductive logic programing (ILP) [26], which exploits the underlying logic encoded in the data to learn new concepts. Some examples are presented in [23,17]. However, there is the inconvenient of rewriting the data sets into logic programming formalisms and most of these approaches are not able to identify some hidden concepts that statistical algorithms would.

Generalized association rules [32] were formerly proposed to consider item taxonomies for mining association rules. The main idea behind this method is to extend itemsets with all the ancestors of each item, then computing the frequent itemsets, and finally filtering out those rules containing an item and some of its ancestors. The resulting rules are expressed at different taxonomy detail levels, avoiding redundant rules present in the taxonomy. Clearly, this method overheads considerably the transactions length and therefore the mining algorithm. Additionally, the number of generated rules is much larger than disregarding the taxonomy. Several optimizations have been proposed to alleviate this overhead. [15] presents two association rules mining algorithms as the core of the Web personalization process. Both algorithms are efficient in the sense that they avoid the costly generation of candidate sets and the over-generalization of rules. A pattern-fragment growth method is adopted along with efficient pruning. Moreover, the special features of Web 2.0 applications are also addressed, where the taxonomies are not predefined but usually described by user-defined tags. In [36] the problem of efficiently updating the discovered generalized association rules when the item taxonomies are modified is addressed. Nevertheless, our work is not concerned with this kind of association rules but with the definition of transactions for SW data. Once the transactions are generated we can mine any kind of association rules, like generalized and quantitative ones.

Other studies extend statistical machine learning algorithms to be able to directly deal with ontologies and their associated instance data. In [8], it is shown how kernel-based machinery can encapsulate the ontology knowledge into the vector-based representation suited for support vector machines. For clustering purposes, a series of similarity (dissimilarity) measures are proposed to mine either semi-structured data [11,14] or ontological instances [12]. In the latter case, knowledge derived from the ontology is used to define the weights of the similarity measures. However, as previously mentioned, these representations are not suited for association mining, which requires to define both the set of items and the transactions from the semantic data.

Another related topic to this work is that of mining tree and graph structured data. In this line, we can find frequent subtree [10] and graph mining [20], whose aim is to identify frequent substructures in complex data sets. Albeit interesting, these algorithms do not serve the purpose of finding interesting content associations in RDF/(S) and OWL graphs because they are concerned with frequent syntactic substructures but not frequent semantically related contents. Indeed, frequent graph substructures usually hide interesting associations that involve contents represented with different detail levels of the ontology. Moreover, although the underlying structure of RDFS and OWL is a graph, reasoning capabilities must be applied to handle implicit knowledge.

The way we define transactions is similar to those proposed for analysing highly heterogeneous XML data sets. More specifically, in [22,38] it is shown that XQuery is not the most suitable query language for data extraction from heterogeneous XML data sources, since the user must be aware of the structure of the underlying documents. The lowest common ancestor (LCA) semantics can be applied to extract meaningful related data in a more flexible way. [22,38] apply some restrictions over the LCA semantics. In particular they propose SLCA [38] and MLCA [22] whose general intuition is that the LCA must be minimal. However, in [27,30] they showed that these approaches still produced undesired combinations between data items in some cases (e.g. when a data item needs to be combined with a data item at a lower level of the document hierarchy). In order to alleviate the previous limitations they propose the SPC (smallest possible context) data strategy, which relies on the notion of closeness of data item occurrences in an XML document. A similar strategy is presented in [34].

Finally, we find some work aimed at integrating knowledge discovery capabilities into SPARQL[3] by extending its grammar. Some examples are [18], which can be plugged with several DM algorithms and [19], which finds complex path relations between resources. Inspired by these works, we have also extended SPARQL grammar to define association rule patterns over the ontological data but in a less restrictive way than the one imposed by SPARQL. These patterns allow the system to focus only on the interesting features, reducing both the number and length of generated transactions.

3. Preliminaries

This section gives some background about the two integrated research areas, the semantic web and association rules, and introduces our mining problem statement in this context.

3.1. Semantic web data

Semantic web technologies are aimed at providing the necessary representation languages and tools to bring semantics to the current web contents. As a result, the W3C consortium has proposed several representation formats, all relying on XML. The resource description language (RDF) was the first language proposed by the W3C to describe semantic meta-data. In RDF there

[3] SPARQL: http://www.w3.org/TR/rdf-sparql-query,'08.

are three kinds of elements: resources, literals, and properties. Resources are web objects (entities) that are identified through a URI, literals are atomic values such as strings, dates, numbers, etc., and properties are binary relationships between resources and literals. Properties are also identified through URIs. The basic building block of RDF is the triple: a binary relationship between two resources or between a resource and a literal. The resulting metadata can be seen as a graph where nodes are resources and literals, and edges are properties connecting them. RDFS extends RDF by allowing triples to be defined over classes and properties. In this way, we can describe the schema that rules our metadata within the same description framework. Later on, the ontology web language (OWL) was proposed to provide well-founded logic inferences over semantic descriptions. Most sublanguages of OWL are supported by decidable subsets of the first order logic, which are known as description logics (DLs) [6]. Additionally, OWL syntax relies on RDF so that technology developed for RDF like triple stores and query languages can be directly applied to OWL. In this work we mainly focus on OWL-DL semantic descriptions.

As application scenario, we have chosen the biomedical field. In this scenario, a huge amount of semantic data are continuously being generated. Specifically, most of the semi-structured and highly heterogeneous data sources (e.g. laboratory test reports, ultrasound scans, images, etc.) are being semantically annotated through comprehensive domain ontologies like UMLS, NCI and Galen. Fig. 1 shows an excerpt of a clinical report whose semantic annotation results in a set of DL axioms (Fig. 2) and assertions (Table 1). The version of the DL used is $\mathcal{ALCHOIQ}(\mathcal{D})$. The axioms in Fig. 2 capture the semantics of all the information related to a patient (i.e. the medical history, reports, laboratory results, etc.) by conceptualizing the domain. Next section explains in detail the OWL-DL constructors used. Table 1 shows the instance data, that is, the triples that describe some patient. The generated data also presents complex relationships that evolve rapidly as new biomedical research methods are applied. Clearly, traditional analytical tools are not appropriate for this kind of data.

3.1.1. Description logics

Description logics allow ontology developers to define the domain of interest in terms of *individuals*, *atomic concepts* (called *classes* in OWL) and *roles* (called *properties* in OWL). Concept constructors allow the definition of *complex concepts* composed of atomic concepts and roles. OWL DL provides for union ($\sqcup$), intersection ($\sqcap$) and complement ($\neg$), as well as enumerated classes (called *oneOf* in OWL) and existential ($\exists$), universal ($\forall$) and cardinality ($\geqslant, \leqslant, =$) restrictions involving an atomic role R or its inverse R^-. In OWL DL it is possible to assert that a concept C is subsumed by D ($C \sqsubseteq D$), or is equivalent to D ($C \equiv D$). Equivalence and subsumption can be also asserted between roles and roles can have special constraints (e.g., transitivity, simmetry, functionality, etc.) Regarding instance axioms, we can specify the class C of an instance a ($C(a)$), or the relations between two instances a and b ($R(a,b)$). A DL ontology consists of a set of axioms describing the knowledge of an application domain. This knowledge ranges over the terminological cognition of the domain (the concepts of interest, its *Tbox*) and its assertions (the instances of the concepts, its *Abox*). Fig. 2 shows an excerpt of the *Tbox* for the running example and Table 1 shows a fragment of the *Abox*.

```
Ultrasonography
     WristExamination
        date '10/10/2006'
        hasUndergoneWrist True
        rightWrist False
        leftWrist True
     WristScore
        wristExamined 'Left'
        PannusAndJointEffusion
           distalRadioUlnarJoint
              result 'No Synovial thikening and no joint effusion'
           radioCarpalJoint
              result 'Only synovial pannus without joint efussion
           midCarpalCMCJ
              result 'None'
        Synovitis
           distalRadioUlnarJoint 'Mild'
           radioCarpalJoint 'None'
           midCarpalCMCJ 'Severe'
        BoneErosion
           distalUlna
              erosionDegree 0
           carpalBones
              erosionDegree 1
...
```

Fig. 1. Fragment of a clinical report of the Rheumatology domain.

In this paper, we separate the *Tbox* from the *Abox* for practical issues. In general, we use the term *ontology* (O) to refer to the *Tbox* and *instance store* (IS) to refer to the *Abox*. The *ontology* is quite static since the terminological axioms describing a domain do not change frequently, while the *instance store* is very dynamic because it is constantly being updated with new assertions. An *instance store* is in fact a triple store consisting of RDF triples whose instances refer to the *ontology*. We assume that the instance store is consistent w.r.t. the associated ontology.

As previously mentioned, the main advantage of adopting OWL is the inference capability. Thus, we are able to infer implicit knowledge derived from both the ontology axioms and the instance store. The main inference tasks we require in our work are combinations of the following ones: concept subsumption, $O \models C \sqsubseteq D$, for any two concepts C and D from O, instance classification, $O \cup IS \models C(a)$ for any concept $C \in O$ and instance $a \in IS$, and instance relationship, $O \cup IS \models R(a,b)$ for any property $R \in O$, and instances a, $b \in IS$. Additionally, we use the function $MSC(i)$ to denote the most specific concept from O to which the instance i belongs (usually the asserted one). All these inference problems have been demonstrated to be decidable in most of the DL families, and for some of them (e.g. OWL 2 profiles) the computational complexity is polynomial.

3.2. Association rules

The problem of discovering association rules was first introduced in [2]. It can be formally described as follows. Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of m literals, called items. Let $D = \{t_1, t_2, \ldots, t_n\}$ be a database of n transactions where each transaction is a subset of I. An itemset is a subset of items. The support of an itemset S, denoted by $sup(X)$, is the percentage of transactions in the database D that contain S. An itemset is called frequent if its support is greater than or equal to a user specified threshold value.

An association rule r is a rule of the form $X \Rightarrow Y$ where both X and Y are non-empty subsets of I and $X \cap Y = \emptyset$. X is the antecedent of r and Y is called its consequent. The support and confidence of the association rule r: $X \Rightarrow Y$ are denoted by $sup(r)$ and $conf(r)$. These measures can be interpreted in terms of estimated probabilities as $sup(r) = P(X \cup Y)$ and $conf(r) = P(Y|X)$ respectively. Another interesting measure is the *lift* or *interest factor* of a rule, denoted by $lift(r)$, which computes the ratio between the rule probability and the joint probability of X and Y assuming they are independent, that is, $P(X \cup Y)/(P(X) \times P(Y))$. If this value is 1, then X and Y are independent. The higher this value, the more likely that the existence of X and Y together in a transaction is not just a random occurrence, but because of some relationship between them.

The basic task of association data mining is to find all association rules with support and confidence greater than user specified minimum support and minimum confidence threshold values [2]. Mining association rules basically consists of two phases: (1) compute the frequent itemsets w.r.t. the minimum support, and (2)

$$
\begin{aligned}
Patient &\sqsubseteq \ =1\ hasAge.string\\
Patient &\sqsubseteq \ =1\ sex.Gender\\
Patient &\sqsubseteq \forall\ hasGeneReport.GeneProfile\\
GeneProfile &\sqsubseteq \forall\ over.Gene \sqcap \forall\ under.Gene\\
Patient &\sqsubseteq \forall\ hasHistory.PatientHistory\\
PatientHistory &\sqsubseteq \exists familyMember.Family\_Group \sqcap \exists\ hasDiagnosis.Disease\_or\_Syndrome\\
Patient &\sqsubseteq \exists\ hasVisit.Visit\\
Visit &\sqsubseteq \ =1\ date.string\\
Visit &\sqsubseteq \forall\ hasReport.(Rheumatology \sqcup Diagnosis \sqcup Treatment \sqcup Laboratory)\\
Rheumatology &\sqsubseteq \exists\ results.(Articular \sqcup ExtraArticular \sqcup Ultrasonography)\\
Rheumatology &\sqsubseteq \ =1\ damageIndex.string\\
Ultrasonography &\sqsubseteq \forall\ hasAbnormality.Disease\_or\_Syndrome\\
Ultrasonography &\sqsubseteq \forall\ location.Body\_Space\_or\_Junction\\
Articular &\sqsubseteq \exists\ affectedJoint.Body\_Space\_or\_Junction\\
Articular &\sqsubseteq \forall\ examObservation.string\\
Diagnosis &\sqsubseteq \exists\ hasDiagnosis.Disease\_or\_Syndrome\\
Treatment &\sqsubseteq \ =1\ duration.string\\
Treatment &\sqsubseteq \exists\ hasTherapy.DrugTherapy\\
DrugTherapy &\sqsubseteq \ =1\ administration.AD\\
DrugTherapy &\sqsubseteq \ =1\ hasDrug.Pharmacologic\_Substance\\
AD &\sqsubseteq \exists\ dosage.string \sqcap \exists\ route.string \sqcap \exists\ timing.string\\
Laboratory &\sqsubseteq \exists\ bloodIndicants.(\exists\ cell.Cell \sqcap \exists\ result.string \sqcap \exists\ test.Lab\_Procedure)\\
Rheumatoid\_Arthritis &\sqsubseteq Autoimmune\_Disease\\
Autoimmune\_Disease &\sqsubseteq Disease\_or\_Syndrome\\
&\ldots
\end{aligned}
$$

Fig. 2. Ontology axioms (Tbox).

Table 1
Semantic annotations (Abox). Subject, predicate and object are URIs pointing to the respective resources. The fourth column shows the types of the related entities (we omit the triple statements asserting the type of the resources for space issues).

Subject	Predicate	Object	Related entities
PTNXZ1	hasAge	"10"	Patient and "xsd:Integer"
PTNXZ1	sex	Male	Patient and Sex
VISIT1	date	"06182008"	Visit and "xsd:Date"
VISIT1	hasReport	RHEX1	Visit and Rheumatology
RHEX1	damageIndex	"10"	Rheumatology and "xsd:Float"
RHEX1	results	ULTRA1	Rheumatology and Ultrasonography
ULTRA1	hasAbnormality	"Malformation"	Ultrasonography and "xsd:String"
ULTRA1	hasAbnormality	Knee	Ultrasonography and Body_Space_or_Junction
VISIT1	hasReport	DIAG1	Visit and Diagnosis
DIAG1	hasDiagnosis	Arthritis	Diagnosis and Disease_or_Syndrome
VISIT1	hasReport	TREAT1	Visit and Treatment
TREAT1	hasDrugTherapy	DT1	Treatment and DrugTherapy
DT1	hasDrug	Methotrexate	DrugTherapy and Pharmacologic_Substance
PTNXZ1	hasVisit	VISIT2	Patient and Visit
VISIT2	date	"08202008"	Visit and "xsd:Date"
VISIT2	hasReport	RHEX2	Visit and Rheumatology
RHEX2	damageIndex	"15"	Rheumatology and "xsd:Float"
RHEX2	results	ULTRA2	Rheumatology and Ultrasonography
ULTRA2	hasAbnormality	"Malformation"	Ultrasonography and "xsd:String"
ULTRA2	hasAbnormality	Knee	Ultrasonography and Body_Space_or_Junction
RHEX2	results	ULTRA3	Rheumatology and Ultrasonography
ULTRA3	hasAbnormality	"Rotation 15"	Ultrasonography and "xsd:String"
ULTRA3	hasAbnormality	Right_Wrist	Ultrasonography and Body_Space_or_Junction
VISIT2	hasReport	DIAG2	Visit and Diagnosis
DIAG2	hasDiagnosis	Systemic_Arthritis	Diagnosis and Disease_or_Syndrome
VISIT2	hasReport	TREAT2	Visit and Treatment
TREAT2	hasDrugTherapy	DT2	Treatment and DrugTherapy
DT2	hasDrug	Methotrexate	DrugTherapy and Pharmacologic_Substance
TREAT2	hasDrugTherapy	DT3	Treatment and DrugTherapy
DT3	hasDrug	Corticosteroids	DrugTherapy and Pharmacologic_Substance
...	...	...	...

generate rules from the frequent itemsets w.r.t. the minimum confidence. As the number of frequent sets can be very large, closed and maximal itemsets can be generated instead. There are also several types of association rules depending on the cardinalities of X and Y (e.g. horn-like rules), and the kind of itemsets that can participate in the rules (e.g. generalized rules [32], quantitative association rules [21,25]), as well as time-dependent rules [31].

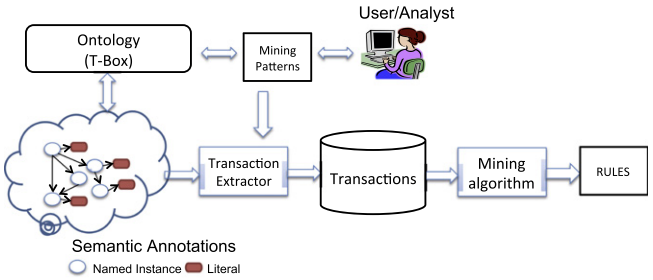

Fig. 3. Arquitecture of our proposal for mining semantic annotations.

```
CREATE MINING MODEL <http://krono.act.uji.es/patients_repository>
   { ?patient   RESOURCE TARGET
     ?drug      RESOURCE
     ?jadi      LITERAL
     ?disease   RESOURCE PREDICT
     ?report    RESOURCE CONTEXT
   }
 WHERE
   { ?patient    rdf:type Patient .
     ?drug       rdf:type Drug .
     ?disease    rdf:type Disease .
     ?report     rdf:type Report .
     ?report     damageIndex ?jadi .
   }
 USING apriori  (SUPPORT = 0.05, CONFIDENCE = 0.7)
 ADD MEASURE lift, leverage
```

Fig. 5. Example of extended SPARQL query with CREATE MINING MODEL statement.

3.3. Instance store association rules

Both the *ontology* and the *instance store* can be encoded into triples of the form "subject-predicate-object", which basically form a graph where nodes are resources and literals, and edges are properties connecting them. This dynamic and graph-based structure contrasts with the well-structured and homogeneous datasets that feed traditional association rules algorithms. Therefore, in order to obtain items and transactions from the *IS*, users must state the target concept of analysis and their involved features. The features must have some kind of relation with the target concept, that is, they will be extracted from the subgraph around each instance belonging to the target concept of analysis. The following definitions introduce the basic concepts of association rules mining in the context of SW data.

Definition 1. A property chain $(r_1 \circ \cdots \circ r_n)$ is an *aggregation path* from concept C to concept C' iff $O \models C \sqsubseteq \exists r_1 \circ \cdots \circ r_n . C'$. The set of all aggregation paths from C to C' is denoted $Path(C, C')$. Aggregation paths can be of length one.

Property chains will allow us to face the structural heterogeneity present in the instance store.

Definition 2 (*Ontology data mining pattern*). An ontology mining pattern Q is a triple $(C_T, C_{ctxt}, Features)$, where C_T is the target concept to which all the concepts in Q must be related in the ontology O through property chains, C_{ctxt} is the context concept from which the transactions are built, and *Features* is a set of concepts from which transaction items are derived from. We will use the functions $target(Q)$, $context(Q)$, and $features(Q)$ to refer to Q's components respectively.

In the definition of a mining pattern, we can indicate that an item can be derived from a concept through a data type property by using the DL expression $C \sqcap \exists DT$, where DT is the data type property used for generating the item.

Definition 3. Transaction generation problem. Given an instance store *IS* consistent with the ontology O, and a mining pattern Q, find co-ocurrences between items derived from instances belonging to $features(Q)$, which must happen within instances belonging to $context(Q)$ associated to instances of $target(Q)$.

In other words, the items co-occurring together under the same context instance are grouped into a transaction. In this new scenario the set of items $I = \{i_1, i_2, \ldots, i_m\}$ is not a set of literal values defined a priori (as opposed to the traditional scenario). Instead, items are complex literals dependent on the context specified by the user. Therefore, they are dynamically derived from the user DM pattern. It is worth mentioning that the generated transactions are binary.

Definition 4 (*Data mining problem*). Given a set of transactions generated from an instance store *IS* consistent with the ontology O and a mining pattern Q, a data mining problem consists in finding association rules with minimum support and confidence threshold values. Support and confidence measures are calculated in the traditional way by taking into account the frequent itemsets.

The previous DM problem can be thought as a classic DM problem where the input data must be derived from the ontology in order to generate transactions according to the user specification.

4. Methodology

In this section we present a detailed view of our method along with the definitions that sustain it. Fig. 3 depicts a schematic overview of the whole process. The user specifies a mining pattern following an extended SPARQL syntax. Then, the transaction extractor is able to identify and construct transactions according to the mining pattern previously specified. Finally, the set of transactions obtained are processed by a traditional pattern mining algorithm, which finds association rules of the form specified in the mining pattern with support and confidence greater than user's specified ones.

4.1. Mining pattern specification

The user has to specify the kind of patterns (s) he is interested in obtaining from the repository. Since semantic annotations are encoded in RDF/(S) and OWL, we have extended SPARQL with a new statement that allows to specify a mining pattern. The syntax is in-

```
Query          ::=   Prologue( SelectQuery | ConstructQuery | DescribeQuery | AskQuery | MiningQuery )
MiningQuery    ::=   CREATE MINING MODEL' Source  '{'
                        Var 'RESOURCE'  'TARGET'
                         ( Var ( 'RESOURCE' | 'LITERAL' )
                           'MAXCARD1'? 'PREDICT'? 'CONTEXT'? )+ '}'
                        DatasetClause* WhereClause UsingClause MeasuresClause
UsingClause    ::=  'USING'  SourceSelector BrackettedExpression
MeasuresClause ::=  'ADD MEASURE' measure +
```

Fig. 4. Extended SPARQL grammar for the CREATE MINING MODEL statement.

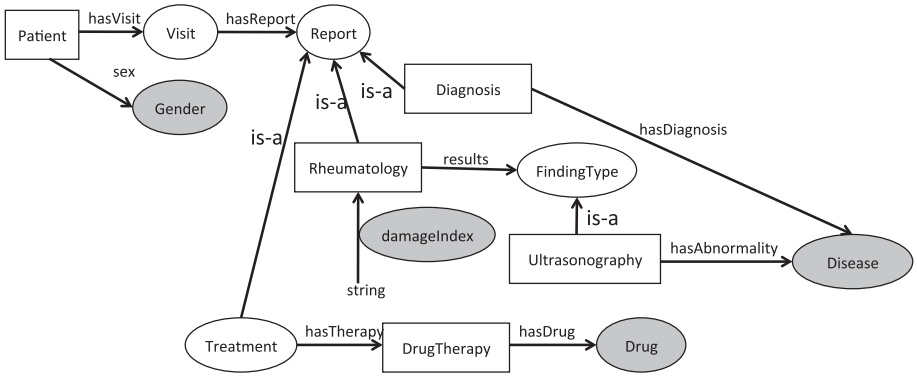

Fig. 6. Example of ontology graph fragment.

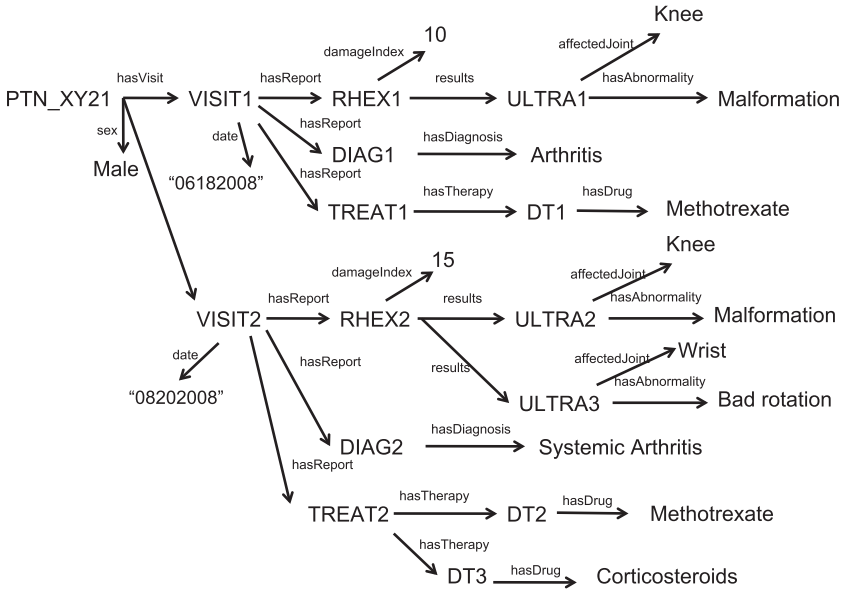

Fig. 7. Example of instance store fragment consistent with ontology fragment of Fig. 6.

Table 2
Example of composition triples.

Subject	Composition path	Object	Feature
PTN_XY21	(VISIT1, RHEX1, ULTRA1)	Malformation	Disease
PTN_XY21	(VISIT1, RHEX1)	RHEX1	*Report* ⊓ ∃ *damageIndex*
PTN_XY21	(VISIT1, TREAT1, DT1)	Methotrexate	Drug
PTN_XY21	(VISIT2, RHEX2, ULTRA2)	Malformation	Disease
PTN_XY21	(VISIT2, RHEX2)	RHEX2	*Report* ⊓ ∃ *damageIndex*
PTN_XY21	(VISIT2, RHEX2, ULTRA3)	Bad rotation	Disease
PTN_XY21	(VISIT2, TREAT2, DT2)	Methotrexate	Drug
PTN_XY21	(VISIT2, TREAT2, DT3)	Corticosteroids	Drug
...	...	...	...

spired by the Microsoft Data Mining Extension (DMX), which is an SQL extension to work with DM models in Microsoft SQL Server Analysis Services.[4]

The extended SPARQL grammar is depicted in Figs. 4 and 5 shows the SPARQL specification for the following mining pattern:

$$Q = (Patient, Report, \{Disease, Drug, Report \sqcap \exists\ damageIndex\})$$

The user has selected *Patient* as target concept of analysis. The set of features that will populate the transactions include diagnosed *diseases*, prescribed *drugs* and the *damage index*[5] measure of the patient. Finally, the transaction will be built at the granularity of *report*, that is, transactions will not include features across reports but only features within each report. The previous mining pattern

[4] DMX Reference: http://technet.microsoft.com/en-us/library/ms132058.aspx.

[5] the damage index is a numerical score that measures the damage of the articulations of rheumatic patients.

Table 3
Item transactions generated for the running example.

#	Transaction
1	{*RheuExam.Disease.Malformation*, *RheuExam.RheuExam. damageIndex* → 10}
2	{*Treatment.Drug.Methotrexate*}
3	{*RheuExam.Disease.Malformation*, *RheuExam.Disease.BadRotation*, *RheuExam.RheuExam.damageIndex* → 15}
4	{*Treatment.Drug.Methotrexate*, *Treatment.Drug.Corticosteroids*}

Table 4
Generated transactions using different contexts for the HeC instance store.

Context	# Trans.	Avg. length
Patient	588	29.57
Visit	1458	12.84
Report	3608	5.24

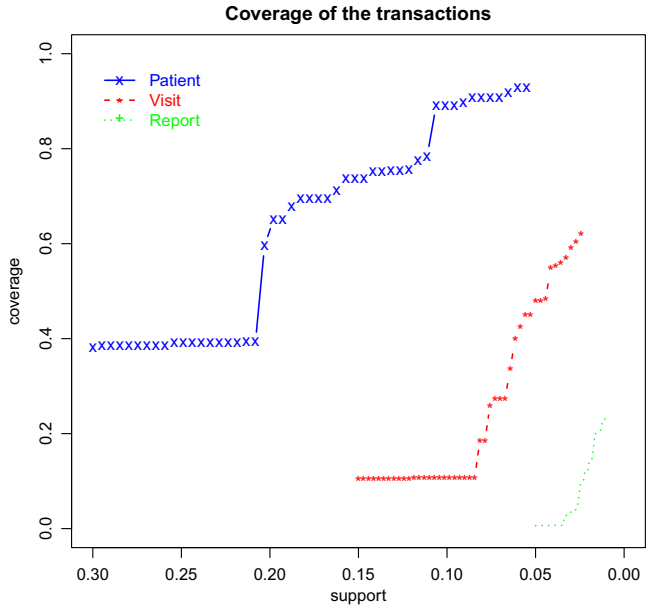

Fig. 8. Coverage of the transactions achieved by the generated rules with different minimum support thresholds.

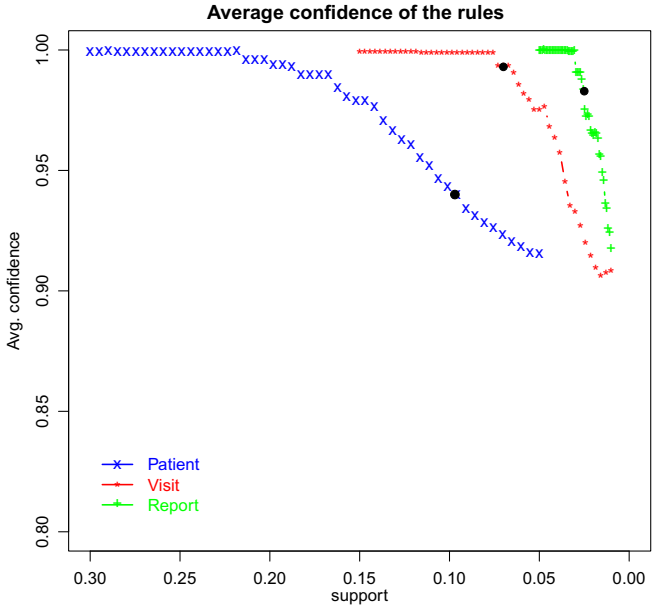

Fig. 9. Average confidence of the generated rules with different minimum support thresholds.

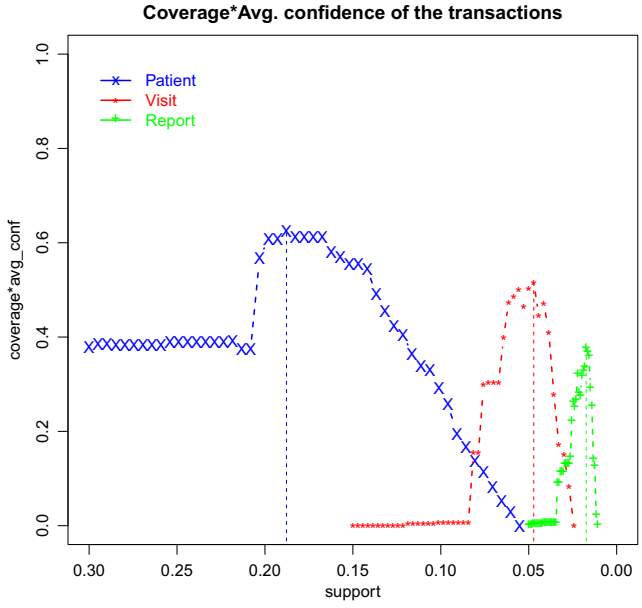

Fig. 10. Coverage multiplied by the average confidence of the generated rules with different minimum support thresholds.

can be specified in a SPARQL-fashion as follows. We extend the SPARQL grammar rule *Query* by adding a new symbol, named *MiningQuery*. This symbol expands to the keywords CREATE MINING MODEL followed by the *Source*, which identifies the input repository. The body consists of the variables the user is interested in with the purpose of data mining. Next to each variable, we specify its content type: RESOURCE for variables holding RDF resources and LITERAL for variables holding literal values. In case we want to find patterns with just one occurrence of the variable, we attach the keyword MAXCARD1 to the variable. By default, patterns found can contain more than one occurrence of each variable. Moreover, we specify the consequent of the rule by attaching the keyword PREDICT (notice that this is optional). Finally, the keyword TARGET denotes the resource under analysis, which must be an ontology concept. The analysis target determines the set of obtained rules. In the *WhereClause*, we specify the restrictions over the previous variables. The good news is that we do not expect users to have an exact knowledge of the ontology structure. Therefore, users do not have to input the paths relating the pattern variables in SPARQL. Instead, they are asked to specify just the type (i.e. ontology concept) that the variables refer to. In case variables are not resources, users must specify the domain and data type property from which that literal is reachable. In the example, the variable *jadi* refers to the damage index score of a patient joint, which is a literal, so the user specifies *report* and *damageIndex* as the resource and data type property from which the value can be accessed, respectively. The *UsingClause* grammar symbol defines the name and parameters of the learning algorithm. Finally, the *MeasuresClause* is used to specify other interestingness measures that the mining model should provide.

Since we do not ask the user to specify the exact relations, the previous query model introduces some ambiguity regarding the items that form a transaction. That is to say, the same conceptual entities (i.e. selected features) may appear under different contexts in the ontology. For example, *Disease* may refer to the patient's

Table 5
Quality measures of the generated rules for the different transaction sets.

Context	Min. sup.	# Rules	Avg. conf.	Avg. lift	ϕ-coeff	Cov.
Patient (588)	0.187	109	0.993	2.944	0.796	0.678
Visit (1458)	0.047	93	0.976	9.975	0.865	0.480
Report (3608)	0.017	151	0.964	27.69	0.836	0.169

Table 6
Quality measures of the generated rules at the context of patient with minimum support set to 0.1 by selecting only transactions containing certain types of reports.

Transactions	# rules	Avg. conf.	Avg. lift	% Multi-context
All reports (588)	655	0.943	4.125	83
Top-5 excl. (585)	22	0.963	6.966	56
Top-12 excl. (438)	26	0.934	6.652	35

diagnosed disease or to some family member disease. This ambiguity makes it challenging for the system to automatically discover what the users' intentions really are. As previously mentioned, the user could remove this ambiguity by specifying in the SPARQL extended query the exact relation of concepts in the ontology through pattern graph triples in the WHERE clause. However, this task can be cumbersome and not always viable. In order to provide the right sense to the query, the user can select the intended context with the CONTEXT keyword attached to the appropriate concept. Alternatively, the system will build the transactions by taking into account all the possible contexts.

4.2. Transaction extractor foundations

Since our goal is to identify and construct transactions according to the user's mining pattern, in this section we present all the definitions that sustain the method we have developed.

Let O be an ontology, Q a data mining pattern, and IS an instance store consistent w.r.t. O. The following definitions state the necessary elements to define the intended transactions over the IS.

Definition 5. The target instances for Q consist of the set $I_T(Q) = \{i / i \in IS, O \cup IS \models C_T(i) \wedge C_T \sqsubseteq target(Q)\}$.

In the running example, C_T can be any subconcept of *Patient* and $I_T(Q)$ is the set of all instances classified as *Patient*.

Definition 6. We define the contexts of two concepts C_a and C_b under C_T, denoted with *Contexts* (C_a, C_b, C_T), as the set of concepts C' satisfying the following conditions:

(1) $C' \in Contexts(C_a, C_b, C_T)$ if $\exists\ p_1 \in Paths(C_a, C') \wedge \exists p_2 \in Paths(C_b, C') \wedge \exists p_3 \in Paths(C', C_T)$, $C_T \sqsubseteq target(Q)$ (C' is common reachable concept), and
(2) $\nexists E \in Contexts(C_a, C_b, C_T)$ such that $\exists p_z \in Paths(E, C')$ (is least).

In this definition the second condition "is least" means that the concept C' is the nearest common reachable concept to C_a and C_b.

Example 1. In Fig. 6, $Contexts(DrugTherapy, Diagnosis) = \{Visit\}$ because $p_1 = DrugTherapy.hasTherapy^{-} \circ hasReport^{-}.Visit$, $p_2 = Diagnosis.hasReport^{-}.Visit$ and $p_3 = Visit.hasVisit^{-}.Patient$.

Definition 7. Let i and i' be two named instances of the instance store IS. $(r_1 \circ \cdots \circ r_n)$ is an aggregation path from i to i' iff:

1. There exists a list of property assertions $r_j(i_{j-1}, i_j) \in IS$, $1 \leqslant j \leqslant n$.
2. There exists two concepts C, $C' \in O$ such that $O \cup IS \models C(i) \wedge C'(i')$ and $(r_1 \circ \cdots \circ r_n) \in Paths(C, C')$

The set of aggregation paths between i and i' is denoted $Path(i, i')$.

The first condition of the definition states that there must be a property chain between both instances in the IS, and the second one states that such a path must be also derived from the *Tbox*.

Definition 8. Let $i_T \in I_T(Q)$ be a target instance, and i_a, $i_b \in IS$ two named instances. $Contexts(i_a, i_b, i_T)$ represent the set of common reachable instances, which is defined as follows:

1. $i' \in Contexts(i_a, i_b, i_T)$ if $\exists p_1 \in Paths(i_a, i') \wedge \exists p_2 \in Paths(i_b, i') \wedge \exists p_3 \in Paths(i', i_T)$ (i' is common reachable instance), and
2. $\nexists i'' \in Contexts(i_a, i_b, i_T)$ such that $\exists p_z \in Paths(i', i'')$ (is least).

Example 2. Fig. 7 shows an example of IS represented as a graph that is consistent with ontology fragment in Fig. 6. In this example, $Contexts(DT1, DIAG1, PTN\_XY21) = \{VISIT1\}$ because $p_1 = DT1.hasTherapy^{-} \circ hasReport^{-}.VISIT1$, $p_2 = DIAG1.hasReport^{-}.VISIT1$ and $p_3 = VISIT1.hasVisit^{-}.PTN\_XY21$.

Definition 9. An *instance transaction* associated to a query pattern Q and a target instance $i_T \in I_T(Q)$, is the set of instances $I_{trans} \subseteq IS$ such that:

- $\forall i \in I_{trans}$, $\exists C' \in features(Q)$, $O \cup IS \models C(i) \wedge C \sqsubseteq C'$, and
- $\exists i_C \in IS$, $\exists p \in Paths(i, i_C)$, $\exists p' \in Paths(i_C, i_T)$, and $O \cup IS \models MSC(i_C) \sqsubseteq context(Q)$.

Notice that the number of transactions corresponds to the number of connected pairs (i_T, i_C).

Example 3. Given the mining pattern $(Patient, Visit, \{Disease, Drug\})$, the set of valid instance transactions are {Malformation, Methotrexate, Arthritis} and {Malformation, Bad rotation, Systemic Arthritis, Methotrexate, Corticosteroids}.

4.3. Transactions and items generation

Recall that a transaction is composed by a set of items co-occurring within instances belonging to $context(Q)$. The items are in turn dynamically generated from instances belonging to $features(Q)$ and the user specified context. This section illustrates how instance transactions are first generated and how the final item transactions are later derived.

The generation of instance transactions from the instance store is performed by first computing the composition triples involved by the mining pattern Q. Basically, a composition triple connects each instance $i_T \in I_T(Q)$ to instances $i' \in features(Q)$ that are connected through a path p of intermediate instances. The path p must contain just one instance i_C belonging to some concept of $context(Q)$.

Composition triples are easily computed by depth-first traversal, starting from the instances retrieved for $target(Q)$ and stopping when an instance of some concept of $features(Q)$ is found. Then the resulting path is validated.

To generate the final transactions, i.e. those that will feed the association mining algorithm, composition triples must be grouped. More specifically, those composition triples with the same subject and the same path sliced at the position of the context instance i_C, are grouped together. The transaction will be generated from the set of *objects* belonging to each group. Only instance transactions satifying Definition 9 are considered for generating the final *item* transactions.

The last step is to generate the item transactions that will serve as input to the association mining algorithm. This step is described in Algorithm 1. This algorithm first expands those instances having some data type property in the feature's set with the corresponding values asserted in the *IS*. Since the items are dependent on the context, they include it in their definition.

Algorithm 1. Generation of item transactions from instance transactions

Require: An instance transaction T, the query pattern Q, the ontology O, the instance store IS, and the composition triples IS_Q.
Ensure: An item transaction T'.
$T' = \emptyset$
i_C be the instance belonging to $context(Q)$ within the composition path associated to T in IS_Q.
for all $i \in T$ **do**
  if $O \cup IS \vDash MSC(i) \sqsubseteq C_i \wedge C_i \in features(Q)$
$\wedge\, MSC(i) \sqsubseteq (C_i \sqcap \exists DTP)$ **then**
    for all $(i, DTP, value) \in IS$ **do**
      update T' with the item $MSC(i_C).MSC(i).DTP \rightarrow value$
    end for
  else
    update T' with the item $MSC(i_C).MSC(i).i$
  end if
end for
return T'

Table 2 shows the composition triples for the mining pattern:

$$Q = (Patient, Report, \{Disease, Drug, Report \sqcap \exists damageIndex\})$$

and the *IS* fragment shown in Fig. 7. Underlined instances belong to concepts of $context(Q)$. The item transactions derived from these composition triples are shown in Table 3.

5. Experimental results

Several experiments have been carried out in this paper to prove that our method is able to generate semantic-based transactions from SW data, which later translates into high quality association rules. Notice the focus of the work is not on developing new mining algorithms but on transforming and making complex semantic data amenable to existing ones. In the future, we will investigate on new mining algorithms that profit from the semantically-enriched items of the generated transactions.

5.1. Dataset

In order to show the usefulness of our proposal, we test the method over a real-world instance store holding OWL annotations about patient's follow-ups. These annotations have been generated in the context of the Healt-e-Child project,[6] and they are consistent with an ontology similar to the one used as example in Fig. 2. The structure of the semantic annotations is very heterogeneous and holds information about 588 patients classified in three different groups according to their disease: Juvenile Ideopathic Arthritis (*JIAPatient*), heart disease (*CardioPatient*) and neurological disease (*NeuroPatient*). The total number of semantic annotations is 629.000, which gives more than 1000 semantic annotations per patient on average.

[6] http://www.healt-e-child.org/.

5.2. Experimental set-up

The generation of the transactions is guided by the user requirements specified in the form of a mining pattern. In order to avoid manual intervention in our experiments, we have automatically generated query patterns for 12 different feature concepts (e.g. *Disease*, *Procedure*,*Drug*, and so on), and 3 concepts for contexts: *Patient*, *Visit* and *Report*. The target concept is always *Patient*. It is worth mentioning that the concept *Report* has 20 subconcepts pairwise disjoint, which correspond to different clinical reports of the HeC project.

The current implementation of the transaction extractor has been developed on top of the ontology indexing system proposed in [29], which also provides a simple reasoning mechanism over the ontology indexes. To prove the usefulness and quality of the generated transactions we have a wide range of association rules algorithms available. Recently, genetic algorithms (GAs) have been proposed for mining interesting association rules [4,3]. However, since the focus of the work is not on the association rules algorithms, we evaluate the generated transactions with more traditional approaches.

In order to obtain the association rules we make use of several utilities of the package *arules* of the R toolkit [16]. First, we calculate the *closed frequent itemsets* [40], which reduces the total number of itemsets specially when transactions have sparse features. Then, we prune itemsets with *cross support* [37] lower than 0.7 in order to filter out patterns that mix frequent and rare items, since they tend to be spurious. Finally we apply the *ruleInduction* function to derive rules from the previous itemsets with minimum confidence set to 0.8.

5.3. Results

The proposed method is able to generate transactions at different contexts specified by the user. Table 4 shows the three selected contexts for the experiments along with the number of generated transactions and their average length. The number and nature of the transactions obtained at every context is quite different and will therefore influence the generated rules. We observe general contexts tend to generate less and longer transactions, which in turn increases the probability of obtaining more rules. On the contrary, specific contexts generate smaller transactions, which hinders the discovery of rules. This disparity in the nature of the transactions makes it necessary to adequately tune the minimum support threshold of each data set to be able to find interesting association rules.

Fig. 8 analyzes the coverage of the generated rules with different support thresholds as an indicator of their quality. The rules obtained with the *Patient* transaction set achieve good coverage rates with relatively high support thresholds. However, the other obtained rule sets are not able to explain a high percentage of the transactions. The intuition behind this is that the length of the latter transaction sets is shorter, which usually tends to decrease the number of association rules discovered. In the case of the *Report* transaction set, the coverage is even lesser because the transactions stem from different types of reports. Therefore, interesting rules might occur but with very low support thresholds. In these cases it would be interesting to apply more sophisticated mining algorithms that do not rely on the notion of support, as those relying on GAs [4].

Fig. 9 shows the average confidence of the generated rules with different support threshold values. We observe the confidence of the rules for the three transaction sets remains high even for low support thresholds, which reasserts the quality of the obtained rules.

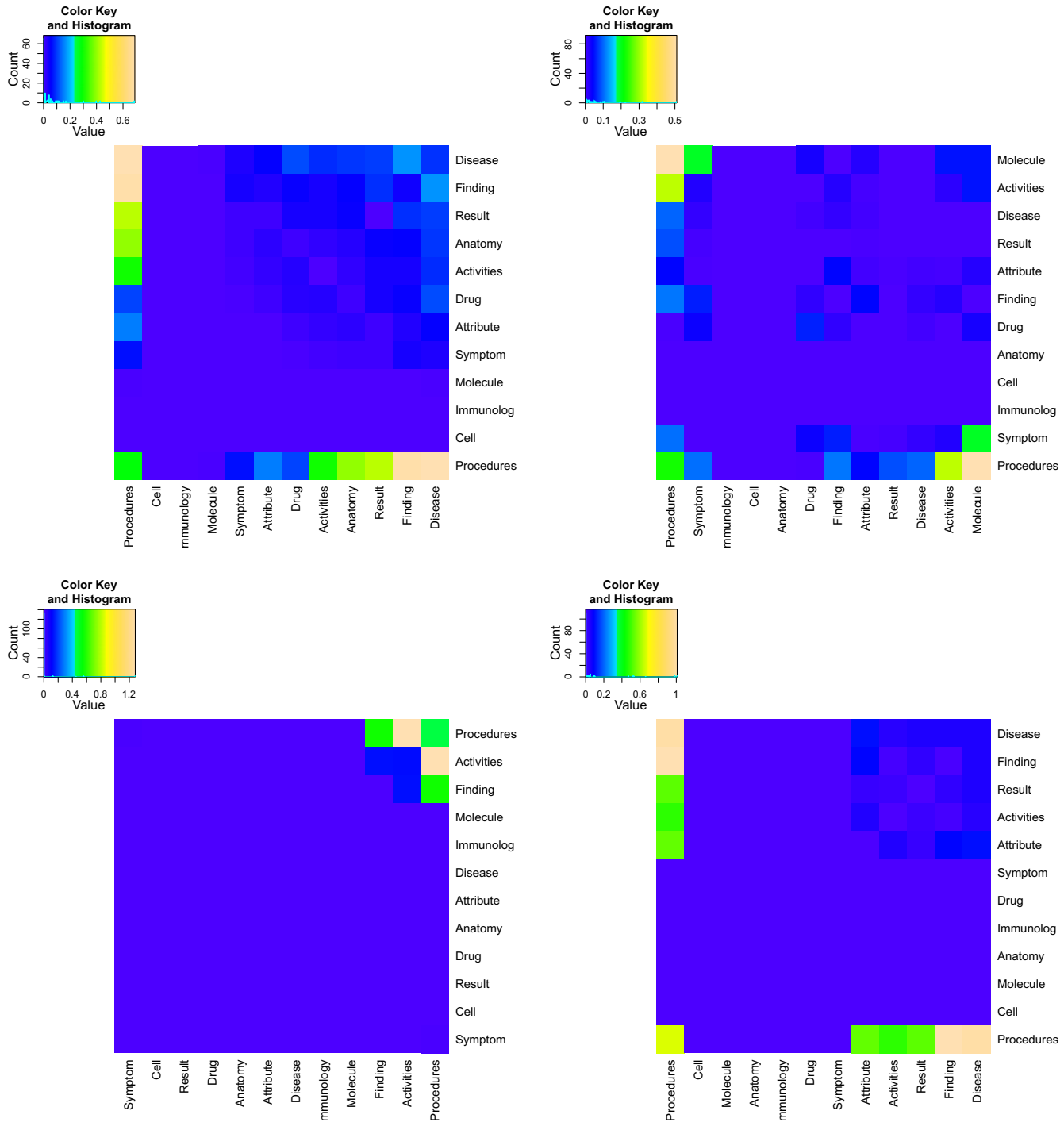

Fig. 11. Heat maps showing the distribution of rules w.r.t. mining pattern features for different context concepts of type *Report*.

Based on the two previous measures (i.e. coverage and average confidence), we want to select a minimum support threshold for each transaction set and further analyze the quality of the obtained rules. Fig. 10 shows the coverage multiplied by the average confidence (re-scaled). For each transaction set, we select the support threshold where both measures are maximized.

Table 5 shows the number of generated rules along with their average confidence, the average lift and the ϕ-coefficient [35] for the three transaction sets. Notice that all generated rules have a high average confidence. Also, the more restricted the context is the stronger are the rules generated. Moreover, the average ϕ-coefficient shows a strong positive correlation in all the cases.

Table 6 shows the impact of applying certain restrictions (i.e. selecting only specific report types) in the mining pattern. Each row shows the resulting transactions and rules regarding all reports, discarding the 5 more frequent reports and finally discarding the 12 more frequent reports at the context of *Patient*. This experiment shows how different patterns can be obtained by selecting the appropriate information through the mining pattern. In this table, we have also included the percentage of rules that have items stemming from different reports. These rules are of special interest

Table 7
Examples of rules with patient as context.

Rule	Supp.	Conf.	Lift
{JIAPatient. (Disease.disease)->oligoarthritis, JIAPatient. (Finding.sacroiliac_tenderness & Anatomy.sacroiliac_joint & MedicalProcedure.presence)} => {JIAPatient. (Symptom.inflammatory_pain)}}	0.26	1.0	18.833
{CardioPatient. (MedicalProcedure.genetic_research & ExternalActivities.genetic_molecular), CardioPatient. (Molecular.tbx5)} => {CardioPatient. (Molecular.5_nkx2)}	0.107	1.0	9.187
{NeuroPatient. (ExternalActivities.endocrinology)} => {NeuroPatient. (MedicalProcedure.resonance_magnetic_material_imaging_contrast &Chemical.metal)}	0.107	0.969	8.380
{JIAPatient. (Symptom.tenderness & Quantity.score)->1} => {JIAPatient. (Finding.pain)->1}	0.144	1.0	6.917
{CardioPatient. (Finding.ejection_systolic_murmur), CardioPatient. (Symptom.cyanosis)} => {CardioPatient. (Anatomy.border_sternal)}	0.108	0.914	5.485

Table 8
Examples of rules with visit as context.

Rule	Supp.	Conf.	Lift
{CardioVisit. (MedicalProcedure.auscultation_lung), CardioVisit. (Quantity.compensation)} => {CardioVisit. (Finding.breath_sounds & Quality.equality)}	0.074	1.0	11.950
{JIAVisit. (Chemical.methotrexate)->Weekly} => {JIAVisit. (Chemical.methotrexate)}	0.08	0.990	7.197
{JIAVisit. (Finding.active_wrist & Disease.arthritis_wrist), JIAVisit. (PatientGroup.parents_legal & MedicalProcedure.data_collection & Result.consent), JIAVisit. (Quality.inclusion_criteria)} => {JIAVisit. (Disease.idiopathic_juvenile_arthritis & ExternalActivities.rheumatology	0.154	1.0	6.451

Table 9
Examples of rules with Rheumatology reports as context.

Rule	Supp.	Conf.	Lift
{JIADiagnosis. (Finding.fever)} => {JIADiagnosis. (Finding.erythematous_rash)}	0.010	0.812	57.439
{Surgery. (Finding.surgery_performed), Surgery. (MedicalProcedure.resonance_magnetic_material_imaging_contrast & Chemical.metal)} => {Surgery. (MedicalProcedure.tumour_removal)}	0.012	0.815	46.137
{DiagnosisCollectingForm. (ExternalActivities. endocrinology), DiagnosisCollectingForm. (Finding. neurological_findings), DiagnosisCollectingForm. (MedicalProcedure. resonance_magnetic_material_imaging_contrast & Chemical.metal)} => {DiagnosisCollectingForm. (MedicalProcedure. tomography_computed)->Abnormal}	0.014	0.808	40.34
{JIALaboratoryOPBG. (Immunology.antibody_antinuclear)} => {JIALaboratoryOPBG. (Immunology.rheumatoid_factor)}	0.0133	0.85	37.951
{CMPPatientGeneralInfo. (Finding.hospital_birth), CMPPatientGeneralInfo. (MedicalProcedure. procedures_therapeutic), procedures_therapeutic), CMPPatientGeneralInfo. (PatientAttribute. pregnancy_term_full), CMPPatientGeneralInfo. (Symptom.cyanosis)} => {CMPPatientGeneralInfo. (Quality.provisional_diagnosis)-> postop_ToF}	0.010	0.838	14.926

for clinicians as they can find associations between clinical variables collected from different departments. Notice that the likelihood of finding multi-report associations decreases with the frequency of the reports.

In order to show the variety of patterns that can be found by selecting different contexts, we have summarized the obtained rules in terms of the features they contain. Summaries are presented in form of heatmaps, where each cell represents a pair of features, and its color indicates the percentage of rules containing that pair. Fig. 11 shows the heatmaps generated for different context concepts of type *Report*. It can be noticed that by considering different contexts we obtain rules of very different nature. For example, the first heatmap shows a high percentage of rules relating procedures with diseases and findings. The second heatmap shows a high percentage of rules relating procedures with molecules. In the third heatmap the prevailing rules are the ones relating procedures with activities. Finally, in the fourth heatmap most of the rules relate procedures with diseases and findings. These maps can also be used to explore and select the rules of interest.

Finally, Tables 7–9 show some interesting rules obtained for the *Patient*, *Visit* and *Report* contexts, respectively. Notice that most of these rules are self-explained and we consider most of them of interest for the clinicians.

6. Conclusions

We have presented a novel method for mining association rules from heterogeneous semantic data repositories expressed in RDF/(S) and OWL. To the best of our knowledge, this problem has only been considered to a minor extent. The intuition under the method

developed is to extract and combine just the interesting instances (i.e. features) from the whole repository and flatten them into traditional transactions while capturing the implicit schema-level knowledge encoded in the ontology. Then, existing association rules algorithms can be applied. We believe this type of learning will become increasingly important in future research both from the machine learning as well as from the SW communities. Initial experiments on real world SW data enjoy promising results and show the usefulness of our approach. As future work, we would like to apply generalized query patterns by using the ontology axioms, as well as to automatically discover interesting contexts and their association rules. Moreover, our method could be applied in a variety of different scenarios such as [34,41,39], where the mining tasks are transaction oriented. An interesting issue for future work is to use the knowledge encoded in the ontology in order to filter and prune discovered rules, and also to express the user goals, similar to the work in [24,7]. Another important direction worth exploring concerns the combination of clustering and association mining algorithms to summarize document collections. This technique was formerly introduced in [13] through the Frequent Itemset based Hierarchical Clustering (FIHC). Basically, the FICH algorithm generates clusters from frequent itemsets, which in turn constitute the cluster descriptors. Several enhancements of this algorithm have been proposed since then (e.g. [41]). Recently, [5] proposed a novel approach also based on frequent item pairs that provides more homogeneous clusters and better descriptions than those obtained with FIHC. Alternative research lines, which are out of the scope of the present work, consist in applying more sophisticated data mining algorithms to the generated transactions [4,3] and study their performance. Equally interesting is to devise new data mining algorithms that take profit from the semantically-enriched items of the generated transactions.

References

[1] Workshop on Mining for and from the Semantic Web (MSW-04), 2004.
[2] R. Agrawal, T. Imielinski, A.N. Swami, Mining association rules between sets of items in large databases, SIGMOD Conference, ACM Press, 1993. pp. 207–216.
[3] B. Alatas, E. Akin, An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules, Soft Comput. 10 (3) (2006) 230–237.
[4] J. Alcala-Fdez, N. Flugy-Pape, A. Bonarini, F. Herrera, Analysis of the effectiveness of the genetic algorithms based on extraction of association rules, Fundam. Inf. 98 (2010) 1–14.
[5] H. Anaya-Sánchez, A. Pons-Porrata, R. Berlanga, A document clustering algorithm for discovering and describing topics, Pattern Recognit. Lett. 31 (6) (2010) 502–510.
[6] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.
[7] K. Becker, M. Vanzin, O3R: Ontology-based mechanism for a human-centered environment targeted at the analysis of navigation patterns, Know.-Based Syst. 23 (2010) 455–470.
[8] S. Bloehdorn, Y. Sure, Kernel methods for mining instance data in ontologies, in: ISWC/ASWC, LNCS, vol. 4825, Springer, 2007, pp. 58–71.
[9] P. Buitelaar, P. Cimiano, B. Magnini (Eds.), Ontology Learning from Text: Methods, Evaluation and Applications, Frontiers in Artificial Intelligence and Applications, vol. 123, IOS Press, Amsterdam, 2005.
[10] Y. Chi, R.R. Muntz, S. Nijssen, J.N. Kok, Frequent subtree mining – an overview, Fundam. Inform. 66 (1–2) (2005) 161–198.
[11] R. Dánger, J. Ruiz-Shulcloper, R.B. Llavori, Objectminer: a new approach for mining complex objects, ICEIS 2 (2004) 42–47.
[12] N. Fanizzi, C. d'Amato, F. Esposito, Metric-based stochastic conceptual clustering for ontologies, Inf. Syst. 34 (8) (2009) 792–806.
[13] B.C.M. Fung, K. Wang, M. Ester, Hierarchical document clustering using frequent itemsets, in: D. Barbará, C. Kamath (Eds.), Proceedings of the Third SIAM International Conference on Data Mining, SIAM, 2003, pp. 59–70.
[14] A. García, R. Berlanga, R. Dánger, A description clustering data mining technique for heterogeneous data, Commun. Comput. Inform. Sci., Softw. Data Technol. 10 (2008) 361–373.
[15] P. Giannikopoulos, I. Varlamis, M. Eirinaki, Mining frequent generalized patterns for web personalization in the presence of taxonomies, IJDWM 6 (1) (2010) 58–76.
[16] M. Hahsler, B. Gruen, K. Hornik, Arules – a computational environment for mining association rules and frequent item sets, J. Stat. Softw. 14 (15) (2005) 1–25.
[17] J. Hartmann, Y. Sure. A knowledge discovery workbench for the semantic web, in: Workshop on Mining for and from the Semantic Web at the ACM SIGKDD, August 2004.
[18] C. Kiefer, A. Bernstein, A. Locher, Adding data mining support to SPARQL via statistical relational learning methods, in: S. Bechhofer, M. Hauswirth, J. Hoffmann, M. Koubarakis (Eds.), ESWC, Lecture Notes in Computer Science, vol. 5021, Springer, 2008, pp. 478–492.
[19] K. Kochut, M. Janik, SPARQLeR: extended SPARQL for semantic association discovery, in: ESWC, LNCS, vol. 4519, Springer, 2007, pp. 145–159.
[20] M. Kuramochi, G. Karypis, Frequent subgraph discovery, in: N. Cercone, T.Y. Lin, X. Wu (Eds.), ICDM, IEEE Computer Society (2001) 313–320.
[21] B. Lent, A. Swami, J. Widom, Clustering association rules, Proc. ICDE'97 (1997) 220–231.
[22] Y. Li, C. Yu, H.V. Jagadish, Schema-free XQuery, in: VLDB '04: Proceedings of the 30th International Conference on Very Large Data Bases, VLDB Endowment, 2004, pp. 72–83.
[23] F.A. Lisi, F. Esposito, Mining the semantic web: a logic-based methodology, in: ISMIS, LNCS, vol. 3488, Springer, 2005, pp. 102–111.
[24] C. Marinica, F. Guillet, Knowledge-based interactive postmining of association rules using ontologies, IEEE Trans. Knowledge Data Eng. 22 (6) (2010) 784–797.
[25] R.J. Miller, Y. Yang, Association rules over interval data, SIGMOD Rec. 26 (2) (1997) 452–461.
[26] S. Muggleton, L.D. Raedt, Inductive logic programming: theory and methods, J. Log. Program. 19 (20) (1994) 629–679.
[27] T. Näppilä, K. Järvelin, T. Niemi, A tool for data cube construction from structurally heterogeneous xml documents, JASIST 59 (3) (2008) 435–449.
[28] V. Nebot, R. Berlanga, Mining association rules from semantic web data, Proc. IEA/AIE (2010) 0–10.
[29] V. Nebot, R.B. Llavori, Efficient retrieval of ontology fragments using an interval labeling scheme, Inf. Sci. 179 (24) (2009) 4151–4173.
[30] T. Niemi, T. Näppilä, K. Järvelin, A relational data harmonization approach to xml, J. Inf. Sci. 35 (5) (2009) 571–601.
[31] S. Ramaswamy, S. Mahajan, A. Silberschatz. On the discovery of interesting patterns in association rules, in: Proceedings of the 24th International Conference on Very Large Data Bases, VLDB '98, San Francisco, CA, USA, 1998, pp. 368–379.
[32] R. Srikant, R. Agrawal, Mining generalized association rules, VLDB (1995) 407–419.
[33] G. Stumme, A. Hotho, B. Berendt, Semantic web mining: state of the art and future directions, Web Semantics: Sci. Services Agents World Wide Web 4 (2) (2006) 124–143.
[34] A. Tagarelli, S. Greco, Semantic clustering of xml documents, ACM Trans. Inf. Syst. 28 (1) (2010).
[35] P.-N. Tan, V. Kumar, J. Srivastava, Selecting the right objective measure for association analysis, Inf. Syst. 29 (2004) 293–313.
[36] M.-C. Tseng, W.-Y. Lin, R. Jeng, Updating generalized association rules with evolving taxonomies, Appl. Intell. 29 (3) (2008) 306–320.
[37] H. Xiong, P.-N. Tan, V. Kumar, Mining strong affinity association patterns in data sets with skewed support distribution, Proceedings of the Third IEEE International Conference on Data Mining, ICDM '03, IEEE Computer Society, Washington, DC, USA, 2003. p. 387.
[38] Y. Xu, Y. Papakonstantinou, Efficient keyword search for smallest LCAs in XML databases, SIGMOD '05: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, ACM, New York, NY, USA, 2005. pp. 527–538.
[39] U. Yun, K.H. Ryu, Approximate weighted frequent pattern mining with/without noisy environments, Know.-Based Syst. 24 (2011) 73–82.
[40] M.J. Zaki, Mining non-redundant association rules, Data Min. Knowl. Discov. 9 (2004) 223–248.
[41] W. Zhang, T. Yoshida, X. Tang, Q. Wang, Text clustering using frequent itemsets, Knowl.-Based Syst. 23 (5) (2010) 379–388.

View publication stats