User-driven Ontology Learning from Structured Data

Carlos Jacinto and Cláudia Antunes
Department of Computer Science and Engineering
Instituto Superior Técnico
Lisbon, Portugal
{carlos.jacinto, claudia.antunes}@ist.utl.pt

Abstract— **The automatic acquisition of models to represent existing domain knowledge is a key step to further develop domain driven data mining. Ontology Learning has been mostly focused on unstructured data sources, as text, leaving structured data almost ignored. This is probably due to the existence of a model behind that kind of data, that without being an ontology, reveals some data semantics. This paper extends the work by Borgida [1], giving to the user the possibility to choose the level of detail of a domain ontology learnt from a relational database. Beside the full exploration of relational model premises, we apply association rules mining to discover basic axioms, which describe the hidden assertions underlying the domain.**

Keywords: Ontology learning, Pattern mining, Relational databases

I. Introduction

Knowledge discovery techniques have been applied in a large range of applications and domains, trying to acquire hidden information, that may help in the decision making process. Ontology learning is one of those applications, with its goal centered on the automatic design of domain ontologies, through the exploration of existing data.

In the context of computer and information sciences, the term *ontology* denotes "an explicit specification of a conceptualization" or in other words "an abstract view of the world we are modeling" [2]. Formally, an ontology is a triple O=(C, R, A^o), where C is a set of *concepts*, R a set of *relations* from concepts and A^o a set of *axioms*. Usually, concepts represent entities, described by both attributes and relations among concepts that define R. An *attribute* is just a unary predicate, and a relation a binary one. *Axioms* are assumptions regarding the intended meaning of concepts and relations, describing additional constraints on the ontology.

The techniques developed in the area of ontology learning mostly approach the bottleneck problem of knowledge acquisition, aiming for reducing the learning time. However, and despite the advances in the area, there are several open issues. In particular, there are few works on learning non-taxonomical relations and axioms. For example, despite the works described in references [3], [4] and [5], the results on the identification of PART-WHOLE relations are just first steps. The second issue that deserves our attention is the discussion on the level of human intervention in the process, with some authors arguing that it is necessary to remove the user from the learning process.

Against the tradition on knowledge discovery, first works on ontology learning were based on the exploration of unstructured data, in particular text (see [6] and [7]). Despite the quality of proposed methods, the results achieved present significant limitations, mostly due to the inherent difficulties of automatic knowledge acquisition, but also to the complexity of mining textual sources.

The choice of exploring unstructured data is explained by the richness and abundance of such sources, but the existence of innumerous well-structured sources, like relational databases, opens the opportunity to present new approaches to the problem. Indeed, those databases present several advantages against text, since they can be seen as simple domain models. However, they usually respect *normal forms* [8], which difficult the domain understanding and the process of information discovery. Moreover, they do not allow for representing additional knowledge about the relations among entities, like axioms in ontologies.

In this paper, we describe a methodology *PaM4OL* (*Pattern Mining for Ontology Learning*) for constructing a domain ontology directly from a relational database, overcoming the fragmentation of the relations inherent to *normal forms*. Our proposal is based on the use of a set of rules for converting entities, attributes and relations in the database into concepts, attributes and relations in the ontology. The set of rules to use is chosen by the user, from a few number of strategies proposed in this work. During the translation, axioms are also derived directly from the model behind the data, but also from the association rules discovered among the data stored.

The rest of the paper is organized as follows: next, we describe the process of converting the database model in a simple ontology, and in section 3 we present a case study in the well-known movies database. The paper concludes with an analysis of the process and the results achieved in the case study, giving some clues for future directions.

II. *PaM4OL* Methodology

Relational databases are a large percentage of existing structured data sources, and in the last decades, have been used often by companies and other organizations, for supporting their operational activities. These databases are centered on the notion of relation, and are composed by a set of tables, one for each relation in the database. Usually, these databases are derived from the entity-relationship (E-R) model, but in a significant number of legacy cases, the E-R model is unknown or not available. Since, rules for creating a relational database from an E-R model are almost standard, the reverse engineering process may be also performed, with some degree of consensus.

The main goal of our approach to ontology learning from relational databases is to extend the work by Borgida [1],

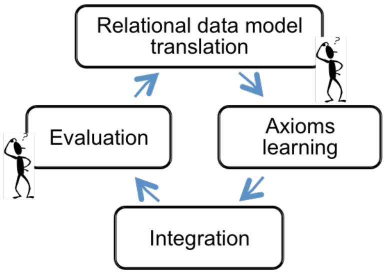

Figure 1 Ontology learning extraction phase

exploring all elements in E-R, and combining it with pattern mining for identifying axioms.

Our methodology, denoted *Pattern Mining for Ontology Learning*, *PAM4OL* for short, corresponds to the cycle depicted in Figure 1. First the relations in the database are translated to a basic ontology, where the concepts, relations and basic axioms are included. This is accomplished automatically, after the user has chosen the most adequate set of rules to translate relational elements into ontological ones.

After the translation, each relational data table is mined in order to discover hidden association rules (rules of the form A➔B). From these, the most interesting ones are used to define a new set of axioms. These axioms are then incorporated in the initial ontology (*integration* step).

The last step closes the cycle; the user is called again, now for evaluating the resulting ontology. User intervention is required for stating if the achieved level of detail is enough for the task. Among other things, the user has to assess if entities are the most adequate to the problem at hands.

A. Relational Data Model Translation

The conversion of relational elements into ontological ones is done through the use of the corresponding E-R model and a set of translation rules, extending the ones proposed in [1]. This conversion is made in two steps: the E-R model design (if needed), and the extraction of the basic ontology elements directly from the model.

In this manner, the first task to accomplish is to warrant the existence of that model; if it is not available then it is necessary to design it automatically through a set of reverse engineering rules that follows.

A table *T* with at least one normal attribute, say A_i, a primary key *T.PK* composed by attributes $B_1 \ldots B_n$ and a primary key of another table (*T'.PK*), creates the following elements in the E-R model: a new weak entity with the name *T* and a new relationship between this weak entity *T* and the entity *T'*. Additionally, $B_1 \ldots B_n$ attributes are mapped into discriminating attributes of the weak entity *T*, and attributes A_i are mapped into normal attributes of *T*.

A similar table *T*, but with no normal attribute, does not create a new weak entity. Instead, $B_1 \ldots B_n$ attributes are mapped into multivalued attributes of the entity *T'*.

A table *R* whose primary key is composed of two attributes corresponding to primary keys of two different tables *T* and *T'*, introduces a new relationship in the E-R model between *T* and *T'*. The type of relationship (one-to-one, one-to-many, many-to-one, many-to-many) and other features like the eventual total participation and cardinality limits may be identified ahead, through pattern mining.

All other tables have a direct mapping into entities. For each table, a new entity with the same name appears with the corresponding primary key and attributes.

The IS-A relationships can be achieved from tables with the same primary key. The way data is arranged between these tables can help to decide the type of generalization to use (*normal*, *total* or *disjoint*).

It is important to notice both role indicators in relationships and derived attributes are not possible to extract from a relation schema.

After having the E-R model for the database, is then possible to create the corresponding ontological elements. However, many questions arise when thinking about these transformations from an E-R model into an ontology. For example, it is not easy to decide if an entity without attributes should be mapped into a concept or an attribute of other concept. We could say that, for example, if that entity is related with more than one entity, probably a mapping into a concept would be more appropriate, but there is no obvious solution for all the other entities without attributes where that condition is not true. It is easy to declare these cases as a concept too, but based on what? Therefore, one must conclude that several types of transformations are needed in order to decide which one is the best. Probably it depends on what answers we are seeking for. However, the questions are not necessarily known before the process, which means that sometimes we do not know exactly what we are looking for.

Since determining the universe of discourse is one of the hardest and controversial decisions, we propose a set of possible strategies for converting the database model into the first version of the ontology. In particular, the different strategies allow for having in consideration the reification problem, letting the user decide what elements should be present in the ontology.

Regardless the strategy followed, some basic rules are always applied

- When an entity in the E-R model has attributes, that entity must be a concept in the ontology. The same rule is valid for relationships too. Since relationships have no direct translation for their attributes in the ontology relations, that relationship must be a concept.

- When an entity or relationship in the E-R model has composite attributes, only the leaves in the trees of attributes can be attributes of concepts in the ontology. It is not necessarily true that those leaves will be attributes of concepts, because that depends on the transformation rules applied, but it is true that all the other E-R attributes of those trees will not be attributes of concepts.

- IS-A relationships are naturally mapped into IS-A relations. Entities where this relation came from must be mapped into concepts.

1) Everything Is Concept

In the first translation strategy, the basic rule is that everything should be a concept. Even simple attributes in the E-R model are mapped into concepts in the ontology with relations to the concept that represents the entity where that attribute was present. Entities without attributes are mapped into concepts with no attributes and relationships without attributes are considered as new concepts related to the concepts that map the entities present in the relationship. Therefore, the resulting ontology has no attributes.

2) *Many Attributes as Possible*

In the second strategy, the opposite approach is followed. Instead of denying the existence of attributes in the ontology, it maps elements into attributes whenever is possible. Attributes in the E-R model are mapped into attributes in the ontology, except non-leaves in trees of composite attributes. Additionally, entities and relationships with attributes are mapped into concepts. However, in this case, relationships without attributes are mapped into relations between different concepts in the ontology. The only exception happens when a relationship links an entity without any other connection to the rest of the E-R model. If this relationship has a single cardinality for this lonely entity, then this entity should be mapped into an attribute.

3) *Weak User Decision*

In the third strategy, called *weak user decision*, the user may decide what must be made for each type of E-R element. For example, the user decides if an E-R entity without attributes should be mapped into a concept or an attribute. However, user has weak power of decision, since he cannot take a different decision for every different entity without attributes.

4) *Strong User Decision*

In the fourth strategy, called *strong user decision*, the user may decide what must be made for each particular E-R element. In this case, the user makes the choice for each different element in the E-R, giving him a strong power of decision. Therefore, every element with possible alternatives in the resulting ontology is picked by user choice.

B. *Learning Axioms*

Besides learning concepts, relations and attributes, there are some axioms that can also be extracted in the translation process. These axioms appear from the E-R model itself. Thus, they are independent of the translation being used.

From the E-R model we are able to identify several kinds of axioms: axioms about concepts, including axioms about inclusion, overlap and disjointness; axioms about attributes; and axioms about relations.

Naturally, axioms about inclusion, overlap and disjointness of concepts come from the normal, total and disjoint generalization of entities in the E-R model, respectively.

Axioms that explicit rules about attributes come from primary keys or discriminating attributes in the E-R model.

Axioms about relations come from the cardinality or eventual total participation of entities in the respective relationships of the E-R model.

A fourth type of axiom, called *general axioms* also can be acquired, not from the E-R model, but through the analysis of stored data. We argue that these axioms may be acquired through association rules discovery.

The same database previously taken as input is then used for pattern mining in order to derive axioms. At first, each table is considered individually. Afterwards, tables are combined (when there is a relevant relationship between the two or more involved entities). The joining continues until all the database is *denormalized*, which means that all the data is stored in just one table. One can join tables by their common columns. The acquired axioms in this process will complement the basic axioms learnt from translation.

The algorithm to use is any of the transactional pattern mining algorithms, like Apriori [9]. After the generation of all possible association rules, one need a filter to denote which ones are interesting for the ontology. Their support and confidence help to understand their value, but axioms do not keep similar information. Rules with 100% of confidence should be used, since they minimize the probability of finding *false positive* axioms, axioms supported by strong rules that appear by pure coincidence.

It is important to note, that axioms are independent of the strategy followed. However, some axioms are easier expressed when using some strategies, such as the “as many attributes as possible”. This is because the extra concepts from the first transformation give a large power of expression about the world (since everything is a concept), but also bring a handicap when trying to write axioms making their representation longer. Thus, one can say that the translation process is based on trade-offs.

C. *Integration and Evaluation*

From the first step concepts, attributes and relations are created. In this manner, C and R are complete, with R accumulating the relations and the attributes derived.

A^o will join the axioms derived from the translation process and from the association rules discovered.

With the three sets defined, the ontology is then complete and ready for evaluation.

The important thing to assess is the adequacy of the ontological elements to express the entities relevant in the domain in analysis. None of the strategies is better than other for a specific domain. Indeed, what determines the strategy to use is the task in hands: depending on the task it is adequate or not to “talk about” a specific element. Whenever this happens, the element should be considered a concept.

In our opinion, only users are able to evaluate which are those elements, and consequently, the centre of the evaluation step.

III. Case Study

In our case study, the domain consists on basic information about movies, like actors, directors, producers, awards, award organizations and studios. The database follows the E-R diagram depicted in Figure 2.

A. *Relational Data Model Translation*

Since E-R model is given, in the first step it is only required to choose one of the strategies to follow.

1) *Everything Is Concept*

The rules following the "Everything Is Concept" strategy make a direct translation from all E-R elements into concepts in the ontology. In this manner, there is no attribute in the new ontology, which increases the number of concepts.

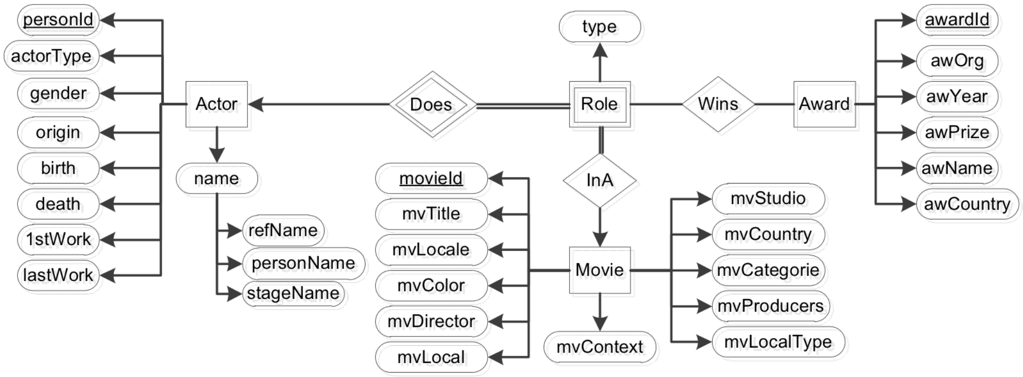

Figure 2 The E-R model of the movies problem.

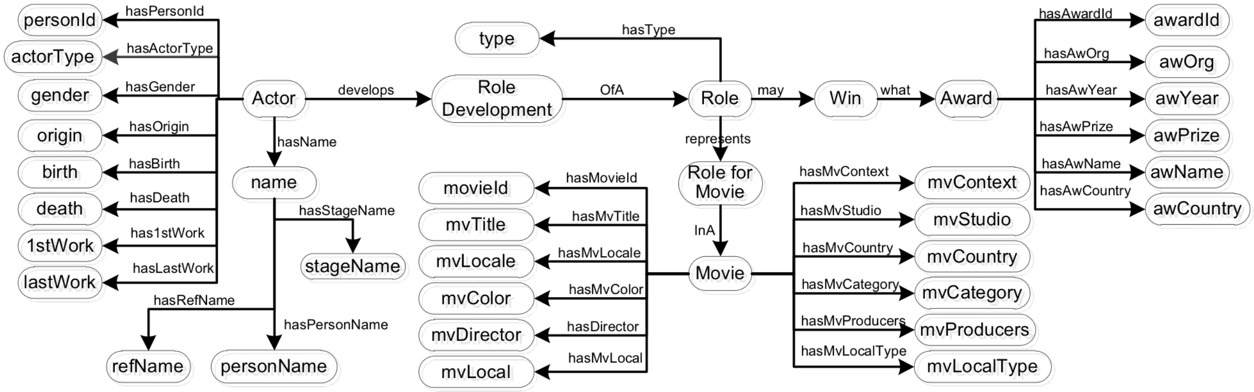

Figure 3 Ontology learnt with *Everything Is Concept* strategy

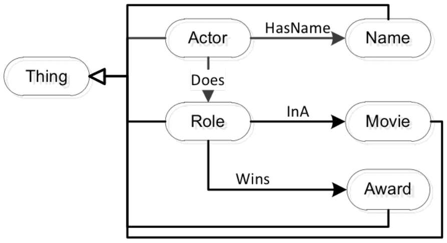

Figure 4 Ontology learnt with *Many Attributes As Possible*

Figure 3 shows the complexity of the resulting ontology. To simplify this figure, the IS-A relations that link every concept with the *Thing* concept are hidden.

As explained in the methodology, relationships in the E-R model are mapped into three elements in the ontology. The *InA* relationship is a good example of this event, being represented by two relations (*Represents* and *InA*) and a new concept *RoleForMovie*. It is possible to see that new names had to be assigned. This is accomplished through user intervention.

In the same way, attributes in the E-R model are mapped into concepts, and new relations appear in the ontology. For example, attribute *Gender* for entity *Actor* in E-R are translated to two concepts, with a new relation defined among them (relation *hasGender*). In these cases, the name is automatically assigned: since an entity can not have multiple attributes with the same name, the concatenation of *has* with the attribute's name is a unique name for a relation.

2) *Many Attributes As Possible*

The second strategy creates a significantly smaller ontology. In this case entities' attributes remain attributes, and relationships' attributes are mapped into concepts. Moreover, entities with a single relationship are mapped into attributes if the relationship has cardinality 1 for that entity. In this case, there is no need of user intervention on assigning names to new elements.

Table 1 Selected axioms learnt for “Everything is concept” (left) and “Many attributes as possible” strategies (right)

<table>
<tr><th colspan="2">Axioms about primary keys</th></tr>
<tr><td colspan="2"><i>1: An Actor must have a personId, since the personId is the primary key of the entity Actor.</i></td></tr>
<tr><td>∀x∈Actor ∃1y ∈personId: HaspersonId(x, y)</td><td>∀(x∈Actor): x.personId ≠ NULL</td></tr>
<tr><td colspan="2"><i>4: Different Actors have different personIds, since the personId is the primary key of the entity Actor.</i></td></tr>
<tr><td>∀w∈Actor, x∈Actor, y∈personId, z∈personId: HaspersonId(w,y) ∧ HaspersonId(x,z) ∧ w≠x ➔ y≠z</td><td>∀x∈Actor, y∈Actor: x≠y ➔ x.personId≠y.personId</td></tr>
<tr><td colspan="2"><i>5: Different personIds must be from different Actors, since the personId is the primary key of the entity Actor.</i></td></tr>
<tr><td>∀w∈Actor, x∈Actor, y∈personId, z∈personId: HaspersonId(w,y) ∧ HaspersonId(x,z) ∧ y≠z ➔ w≠x</td><td>∀x∈Actor, y∈Actor: x.personId≠y.personId ➔ x≠y</td></tr>
<tr><th colspan="2">Axioms derived from total participation in relatioships</th></tr>
<tr><td colspan="2"><i>10: Every Role must be developed by an Actor, since the Role has total participation in the relationship Does.</i></td></tr>
<tr><td>∀x∈Role ∃y∈RoleDevelopment, z∈Actor: Develops(z,y) ∧ Of_A(y,x)</td><td>∀x∈Role ∃y∈Actor: Does(y,x)</td></tr>
<tr><th colspan="2">Axioms from relationships’ cardinality</th></tr>
<tr><td colspan="2"><i>12: Different Actors develop different Roles, since the Does relationship's cardinality is one-to-many.</i></td></tr>
<tr><td>∀x1∈Actor, x2∈Actor, y1∈RoleDevelopment, y2∈RoleDevelopment, z1∈Role, z2∈Role: Develops(x1,y1) ∧ Develops(x2,y2) ∧ Of_A(y1,z1) ∧ Of_A(y2,z2) ∧ x1≠x2 ➔ y1≠y2 ∧ z1≠z2</td><td>∀x∈Actor,y∈Actor,z∈Role: x≠y ∧ Does(x,z) ➔ ~Does(y,z)</td></tr>
</table>

Figure 4 shows the resulting ontology. Attributes are not represented in order to simplify, but every concept has attributes with the same names as the attributes present in the E-R model.

B. Learning Axioms

As said above, a few axioms are also acquired automatically during the translation step. Table 1 presents an example of each kind of axiom learnt, expressed in first order logic. Consider that *R(A,B)* is true in the ontology if and only if exists a relation *R* between the concepts *A* and *B* in the ontology. Similar axioms are found for *Movie*, *Award* and *Role* concepts, totalizing 13 axioms.

After translation, follows the definition of axioms from the discovery of association rules. This extraction from data is independent of the translation used. However, the way each axiom is represented differs, since the same name may refer an attribute, a concept or a relation in the ontology.

It is important to notice that the list of acquired axioms is too extensive. Thus, only a few are presented. The first axiom in Table 2 is discovered from mining the *Awards* table alone. From this, one can say that all the awards from the *Hollywood Academy* are handed out in the *United States of America*. Obviously, anyone knows that, but this information was captured automatically here, in order to complete the ontology.

From the *Actors* table alone, among many other axioms, 2, 3, 4 and 5 appear to reveal information that can be useful. Therefore, one can say that only actresses have the "beauty" *actorType*, and only male actors have the "burly", "cowboy" or "hero" *actorType*.

However, in this same table a false positive (axiom 6) is found. Apparently, all the actors that died in 1936 were male. This is certainly a coincidence and may not be relevant, but the axiom extraction does not take this into account. It is a blind process that selects patterns of data independently of its semantics. In this sense, the user interaction is also required in order to decide what is relevant to the problem.

Regarding the *Movies* table alone, the process found axiom 7 and its symmetric. Thus, one can say that "Schomburgk" produced all the movies he directed and vice-versa. This happens with many other directors.

Other good examples of axioms extracted from the *Movies* table are axioms 8 to 11, stating that with "Daugherty" directing, the movie is "black & white"; with "Crosland", it is a "suspense and/or thriller" movie; with "G. Thomas", it is a comedy movie.

Joining *Actors* and *Casts* tables allows for finding the first axioms relating two different elements (axiom 12). This axiom states that all roles performed by “Sidney Toler” correspond to a "detective" character. This axiom demonstrates the relevance of the application of pattern mining on learning axioms. Indeed, it is the only way to detect these cases, only stored in data, and completely independent of the model behind the data.

C. Evaluation and Comparison

The direct comparison of both learnt ontologies is the best evaluation that can be made. The first ontology contains 38 concepts, no attributes and 37 non-taxonomic relations. On the other hand, the second strategy achieves 6 concepts, 30 attributes and only 4 non-taxonomical relations.

It is clear from this case study, that making everything a concept is an extreme solution, since sometimes there are irrelevant data for the problem at hand. Since entities are always mapped into concepts, the best solution is to make other concepts only from the relevant attributes. On the other hand, the axioms are simpler when they refer attributes instead of concepts, as it is easily noticeable above.

In this sense, the best solution appears to be a trade-off. The user interaction options are probably the most desired, making this methodology enough agile to focus on the relevant information for the learnt ontologies.

In terms of axioms learning, tests were made for three different levels of minimum confidence: 100, 98 and 95%. The minimum support used was always the inverse of the

Table 2 Selected axioms learnt from association rules

```
1.  awardOrganization=Hollywood Academy of Motion
    Picture Arts and Sciences ➔ awardCountry=USA
2.  actorType=beauty ➔ gender=F
3.  actorType=burly ➔ gender=M
4.  actorType=cowboy ➔ gender=M
5.  actorType=hero ➔ gender=M
6.  dateOfDeath=1936 ➔ gender=M
7.  movieDirector=Schomburgk➔movieProducers=Schomburgk
8.  movieDirector=Daugherty➔ movieColor=bnw
9.  movieDirector=CroslandJr➔movieCategory=S&T
10. movieDirector=GThomas➔movieProducers=PeterRogers
11. movieProducers=PeterRogers➔movieCategory=Comedy
12. actorStageName=Sidney Toler➔type=detective
```

number of rows for each table (for example 0.01 for a table with 100 entries). Among the axioms discovered, some are false positives, and others are repetitive, by adding more propositions to the antecedent or consequent, which make them irrelevant.

As we can see, decreasing the level of minimum confidence offers the possibility to increase the total number of acquired association rules. But, unfortunately, most of the ones found in this case, are irrelevant and the increase of good ones was not significant.

IV. Conclusions

Ontologies gained a considerable attention in the area of knowledge management and in information systems, since they are a manageable way to represent domain knowledge, making it useful for many purposes. However, the manual construction of ontologies by knowledge engineers, suffers from several difficulties, which had given raise to the field of ontology learning.

From the works on this area, several approaches deal with non-structured data, but just a few were dedicated to other types of sources. While keeping this idea in mind, a new approach for ontology learning from structured data is introduced here.

The main goal of our approach is to explore the underlying rules of E-R models, converting them directly to ontological elements, namely concepts, attributes, taxonomical and non-taxonomical relations, but also basic axioms. This exploration is then combined with association rules mined from stored data, in order to define a set of axioms for completing the ontology.

On the other hand, our approach considers user intervention as a key point in the learning process, requiring it both for choosing the translation strategy to apply and for evaluating the results achieved.

The case study reported in this paper shows that the process can be accomplished effectively, but also denotes that more work should be done in the discovery of data hidden axioms.

In particular, it would be of great interest to explore more advanced pattern mining techniques that should be used to generalize similar axioms, like the ones created by translation (Table 1), but also the ones discovered by association rules. To our knowledge, such techniques do not exist yet, but may be effectively created in a near future with the advances in the exploration of the semantic aspects on data mining, in particular with the use of constraints.

Acknowledgment

This work was partially supported by Fundação para a Ciência e Tecnologia under research project D2PM (PTDC/EIA-EIA/110074/2009).

References

[1] Borgida, A., Lenzerini, M., Rosati, R.: Description Logics for Databases. In : The Description Logic Handbook - Theory, Implementation and Applications. (2003)

[2] Gruber, T.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Technical, Stanford University (1993)

[3] Berland, M., Charniak, E.: Finding parts in very large corpora. In : Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, pp.57-64 (1999)

[4] Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S.: Learning to extract symbolic knowledge from the World Wide Web. In : National Conference on Artificial Intelligence, pp.509-516 (1998)

[5] Hearst, M. A.: Automatic acquisition of Hyponyms from large text corpora. In : International Conference on Computational Linguistic, pp.539-545 (1992)

[6] Maedche, A.: Ontology Learning For The Semantic Web. Kluwer Academic Publishers (2001)

[7] Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer (2006)

[8] Chen, P.: The E-R Model - Toward a Unified View of Data. ACM Transactions on Database Systems, (1976)

[9] Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In : Int'l Conf on Very Large Data Bases, Chile, pp.487-499 (1994)