An ontology engineering approach for knowledge discovery from data in evolving domains

Paulo Gottgtroy, Nik Kasabov, Stephen Macdonell
Knowledge Engineering and Discovery Institute,
Auckaland University of Technology – New Zealand

Abstract

Knowledge discovery in evolving domains presents several challenges in information extraction and knowledge acquisition from heterogeneous, distributed, dynamic data sources. We define an evolving process if the process is developing, changing over time in a continuous manner. Examples of such domains include biological sciences, medical sciences, and social sciences, among others.

This paper describes research in progress on a new methodology for leveraging the semantic content of ontologies to improve knowledge discovery in complex and dynamical domains. We consider in this initial stage the problem of how to acquire previous knowledge from data and then use this information in the context of ontology engineering. The first part of this paper concerns some aspects that help to understand the differences and similarities between ontologies and data models, followed by an analysis of some of the methods and on going researches in the process of building ontology from databases in evolving domains, or ontology learning from databases.

In the second part we describe our approach to build a framework able to enhance ontology learning and discovery from data and present future directions of our research integrating ontology and evolving connectionist systems that is being developed in the Knowledge Engineering & Discovery Research Institute - Kedri.

1 Introduction

There are substantial research challenges in modelling evolving data interactions, extracting valuable knowledge, and building a reusable knowledge base that provides ongoing solutions to new and existing problems. For instance, evolving processes [1] are inherently difficult to model because some of their parameters are unlikely to be known a priori. Unexpected perturbations or changes may happen at certain times in their development, so they are not strictly predictable in the longer term. Thus modelling of such processes is a challenging task with many practical applications in business and in the biological and medical sciences.

In recent years Ontologies [2] have been increasingly used to provide a common framework across disparate systems, especially in bioinformatics, medical decision support systems, and knowledge management. Ontology is defined in artificial intelligence literature as a specification of a conceptualisation [3]. An ontology specifies at a higher level the classes of concepts that are relevant to the application domain and the classes of relations that exist between these classes. The ontology captures the intrinsic conceptual structure of a domain. For any given domain, its ontology forms the heart of the knowledge representation.

This paper describes research in progress on a new methodology for leveraging the semantic content of ontologies to improve knowledge discovery in complex and dynamical domains. The first part of this paper concerns some aspects that help to understand the differences and similarities between ontologies and data models, followed by an analysis of some of the methods and on going researches in the process of building ontology from databases in evolving domains, or ontology learning from databases.

In the second part we describe our approach to build a framework able to enhance ontology learning and discovery from data and present future directions of our research integrating ontology and evolving connectionist systems that is being developed in the Knowledge Engineering & Discovery Research Institute - Kedri.

2 Data Model x Ontology Engineering

The current interest in ontologies is the latest version of Artificial Intelligence's alternation of focus between content theories and mechanism theories [4]. Sometimes, the Artificial Intelligence community gets excited by some mechanism such as rule systems, frame languages, neural nets, fuzzy logic, constraint propagation, or unification. The mechanisms are proposed as the secret of making intelligent machines. At other times, we realize that, however wonderful the mechanism, it cannot do much without a good content theory of the domain on which it is to work. Moreover, we often recognize that once good content theory is available, many different mechanisms might be used equally well to implement effective systems all using essentially the same content.

Ontologies in current computer science language are computer based resources that represent agreed domain semantics. Unlike data models, the fundamental asset of ontologies is their relative independence of particular applications, i.e. an ontology consists of relatively generic knowledge that can be reused by different kinds of applications/tasks. [5]

A data model, on the contrary, represents the structure and integrity of the data elements of the, in principle "single", specific enterprise application(s) by which it will be used. Therefore, the conceptualisation and the vocabulary of a data model are not intended a priori to be shared by other applications.

Furthermore, in the data modelling practice scenario the semantics of data models often constitute an informal agreement between the developers and the users of the data model and, in many cases, the data model is updated on the fly as particular new functional requirements pop up without any significant update in the metadata repository.

On the other hand both ontology model and data model have similarities in terms of scope and task. They are context dependent knowledge representation, that is, there doesn't exist a strict line between generic and specific knowledge when you are building ontology. Moreover, both modelling techniques are knowledge acquisition intensive tasks and, the resulted models represent partial account of conceptualisations.

In spite of the differences, we should consider the similarities and the fact of data models carry a lot of useful hide knowledge about the domain in its data schemas, in order to build ontologies from data and improve the process of KDD.

In the next section we present a non-exhaustive overview of the current research in the field of ontology learning from databases. Ours analysis is limited by few numbers of published work in this area which shows the dimension of this open problem.

3 Ontology learning from data

The fact of data schemas do not have the required semantic knowledge to intelligently guide ontology construction has been presented as a challenge for database and Ontology engineers. In this section we describe different methods and approaches that allow the extraction of Ontologies or semantics from database schemas.

This review is based on the investigation done by the Ontoweb Group [6] and by our research group. This section summarizes the most relevant methods used for ontology learning from relational schemata in alphabetical order. The name of authors is used as reference for the method. At the end we show a table 1 summarizing the main aspects of each approach.

3.1 Johannesson's method

This method [7] aims to translate a relational model into a conceptual model with the objective that the schema produced has the same information capacity as the original schema. The method starts transforming the relational schemas into a form appropriate for identifying object structures. After the initial transformations, the relational model is mapped into a conceptual schema. The iterations with the user are needed during the translation process. For each candidate key, a user must decide whether it corresponds to an object type of its own, and for each inclusion dependency where both sides are keys, a user must decide whether it corresponds to an attribute or a generalization constraint.

The method bases its functionality on four different transformations: candidate key splitting (occurs when a relation scheme in third normal form corresponds to several object types), inclusion dependency splitting (when a single relation corresponds to several objects types), folding (when several relation schemes correspond to a single object type), and schema mapping (to map a relational scheme into an object type)

3.2 Kashyap's method

One of the most important goals for this project is to develop technologies that operate on heterogeneous information sources in a dynamic environment. In their approach the fundamental premise of building domain ontology from database schemas is that the knowledge specific to the domain is embedded in the data and the schemas of the selected databases.

The method [8] uses the database schemas to build an ontology that will then be refined using a collection of queries that are of interest to the database users. The process is interactive, in the sense that the expert is involved in the process of deciding which entities, attributes and relationships are important for the domain ontology. It is iterative in the sense that the process will be repeated as many times as necessary.

The process has two stages. In the first one, the database schemas are analysed in detail to determine keys, foreign keys, and inclusion dependences. As a result of this process a new database schema is created, and by means of reverse engineering techniques, it is content is mapped into the new ontology. In the second stage, the ontology constructed from the database schemas has to be refined to better reflect the information needs of the user and can be used to refine the ontology.

3.3 Phillips and colleagues' approach

This system [9] scans new databases to obtain type and constraint information, which users verify (figure 1). The system then uses this information in the context of a shared ontology to intelligently guide the potentially combinatorial process of feature construction. Further, the system aims to learn each time it is applied, easing the user's verification task on subsequent runs.

The goal of this approach is to exploit the information contained in Ontologies to the help KDD process. Specifically, they hope to:

1. Automatically suggest and generate new attributes based upon semantic and domain information,
2. Capture useful knowledge for reuse, and
3. Reduce the user's workload to interpret new tables.

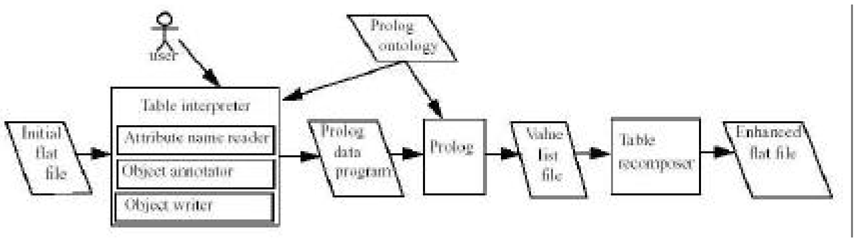

Figure 1. Phillips' process.

3.4 Rubin and colleagues' approach

This approach [10] proposes to automate the process of filling the instances and their attributes' values of an ontology using the data extracted from external relational sources. This method uses a declarative interface between the ontology and the data source, modelled in the ontology and implemented in XML schema. The process allows the automatization of updating the links between the ontology and data acquisition when the ontology changes. The approach needs several components: an ontology, the XML schema (is the interface between data acquisition and the ontology), and an XML translator (to convert external incoming relational data into XML when it is necessary).
The proposed steps are:
1. Create the ontology model for the domain.
2. Creating the XML Schema. Once the ontology is built and the constraints on data values are declared, the XML schema is sufficiently determined, and it can be written directly from the ontology.
3. Data acquisition. Data acquired from external relational data sources must be put into an XML document that uses the syntax specified by XML schema.
4. Ontology evolution and propagating changes.

3.5 Stojanovic and colleagues' approach

This approach [11] tries to build light ontologies from conceptual database schemas using a mapping process. To carry out the process, it is necessary to know the underlying logical database model that will be used as source data.
The approach has the following five steps to perform the migration process.
1. Capture information from a relational schema through reverse engineering.
2. Analyse the obtained information to built ontological entities by applying a set of mapping rules.
3. Schema translation. In this step the ontology is formed.
4. Evaluate, validate and refine the ontology.
5. Data migration. The objective of this step is the creation of ontological instances based on the tuples of the relational database.
The next to approaches were not proposed for ontology learning, but both have a potential methodology that can be applied for ontology learning when combining or extending its approach with some of the techniques used by the previous approaches above.

3.6 Saltz and colleagues' approach

This approach [12] aims to provide limited knowledge awareness to a conventional DBMS (Database Management Systems). This goal is achieved by extending DBMS in such way that it becomes ontology aware. The concept of ontology is used in this approach as a way of formalizing knowledge and relationships among objects in a domain of interest.

The solution is compounded by two main pieces: an external knowledge server and a set of functions to extend the DBMS. The main objective is enhance adhoc queries in such way that both queries and its results are meaningful for the users.

They argue that their solution is both powerful in the sense of supporting knowledge retrieval in the queries, and generic, in the sense that it can be deployed in any DBMS with the support for user-defined functions.

Although this method has not been developed for ontology learning from database, we've selected it because a mapping technique can be applied in such way that it can be used to refine the ontology through the rules generate by the query engine.

3.7 Spyns and colleagues' approach

Spyns' approach [5] is based on ORM (Object Role Modelling). ORM may be classified among the semantical network approaches to knowledge representation that were popular in AI and in database design especially in the 1970s, and later. It is a semantically rich modelling language that was extended to support the data modeling process through a graphical and intuitive representation that translate the ORM model into entity-relationship diagram and its physical implementation.

An Object Role Modelling Mark-up Language has been developed to represent ORM [13] models in an XML-based syntax to facilitate the exchanging of ontology models. The agreed semantical knowledge expressed in ORM is done in much the same way that "classical" databases take data structures out of these applications.

Both graphical representation and declarative textual representation of the ontological commitments are easy to understanding and well established in the database community, thus this methodology is a quite good start point when the ontology engineer has a strong background in data modelling.

Although this method hasn't been proposed for ontology learning from databases, it can be extending through some reverse engineering techniques and be implemented as an alternative for learning from relational databases.

7.3.1 Summary of ontology learning methods from relational schema.

Name	Main goal	Techniques used	Sources used for learning
Johannesson's method	To map a relational schema with a conceptual schema	Mappings	Relational schemas
Kashyap's method	To create and refine an ontology	Mappings and Reverse engineering	Schemas of domain specific databases
Phillips and colleagues' approach	To create and refine an ontology	Induction inference	Flat files
Rubin and colleagues' approach	To create ontological instances	Mappings	Relational schema of a database
Stojanovic and colleagues' approach	To create ontological instances from a database	Mappings and Reverse engineering	Schemas of domain databases
Saltz and colleagues' approach	enhance adhoc queries	Rule generation	Relational databases
Spyns and colleagues' approach	To create an ontology	Graphic Modelling	Relational databases

Table 1 - Summary of ontology learning methods from relational schema.

4 Kedri's Approach

It is already well accepted that Ontologies are useful for data integration and data translation between systems. Although ontology-engineering tools have matured over the last decade, manual ontology acquisition remains the most frequently used approach to knowledge representation. This is, however, a tedious, cumbersome task that can easily result in a knowledge acquisition bottleneck[14], particularly where large volumes of data are concerned. Therefore, in the context of evolving processes, ontologies should be created and refined automatically.

A tool that gradually accumulates knowledge of the data-bases of a domain is appropriate for and applicable to knowledge discovery from data because KDD is an iterative process where any change in one of the source databases should represent an input to a new knowledge discovery process.

As in [9] we do not presume that an ontology is complete at the time a new data mining application is begun to the contrary, we believe that new domains will bring new types of variables and knowledge about them. However, we also believe that data mining is not simply the one-time application of a program to a new database. In our own work, data mining frequently starts with small pilot studies and manual bias space search, including feature construction. With preliminary confirmation that the programs can find some interesting relationships, more data and greater expectations are introduced.

In order to keep the knowledge domain up to date, sharable, and reusable for different applications, we are investigating a hybrid approach putting together the state of art of the AI methods for knowledge discovery in large databases (KDD) and the ontology engineering (figure 2).

The framework integrates both content and mechanism theories. Evolving connectionist systems (ECOS) [15] paradigm, that is aimed at building on-line, adaptive intelligent systems that have both their structure and functionality evolving in time, is used as a mechanism to find new relationship and patterns from the data. The rules extracted update the ontology that is used as knowledge visualization tool for another data mining process.

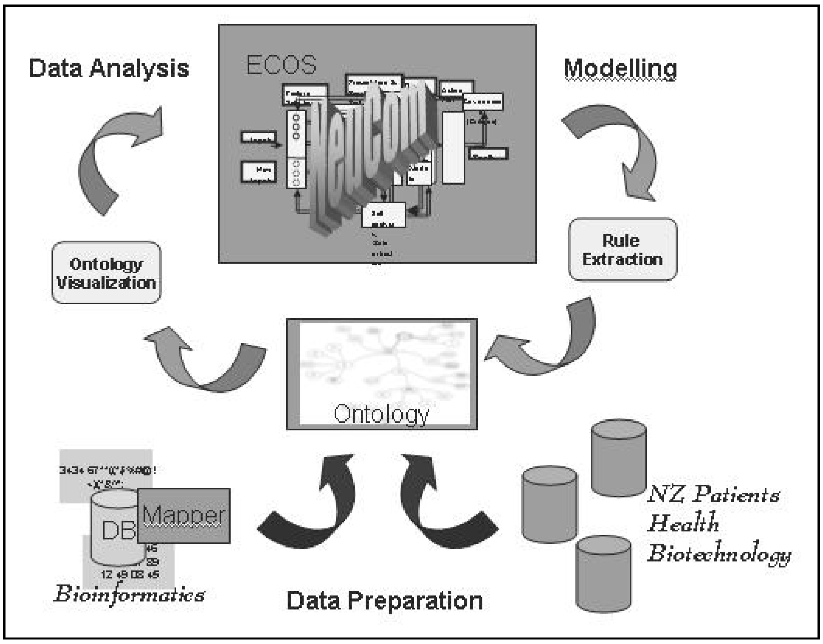

Figure 2. Proposed framework.

Our development is part of Neucom [16] and brings to it a new dimensional in terms of data preparation. In this stage we are using manual mapping techniques to build the ontology from data. The same mapper is used to acquire, transform and analyze data from flat files, relational data and ontology integration and pass it to Neucom environment for further analysis and modelling.

Our approach has some similarities with Phillips's, and Slatz' s approaches in terms of rules generation but improve it because our rules are evolving dynamically with new knowledge inputs and are represented as meaningful fuzzy rules. In this stage we have similarities with Johannesson's, Kashyap's, Rubin's, and Stojanovic's approaches in terms of mapping technique, and user intervention, but we differ from their approaches because we can learning from different schemas, such as, XML schemas, relational databases, flat files as well as from Ontologies.

The approach followed by Spyns and colleagues, which uses an interesting graphic modelling language technique based on well known ORM methodology to build Ontologies, differs from our approach that is based on MCS methodology [17] and Protégé's Graph Widget [18], but has similarities because both are concerned about previous knowledge hided in the schemas models.

We believe that our method and Kashyap's method are more adequate than other initiatives for KDD because the main goal of these methods is create and refine the ontology. However our method is improving this approach because we are considering other schemas models instead just relational like flat file considered by Phillips.

5 Conclusion

Ontology learning from data is quite new and open area for ontology engineering and database communities. However we believe that a different approach from the current ones should be followed that includes as many as possible data schemas instead of the majority effort in relational data. Moreover, we should think in approaches that are able to integrate both Ontologies and mechanisms paradigms, such as, fuzzy, machine learning, neural network, etc, and consider the dynamic of the real world problems.

Our effort is an attempt to integrate both paradigms aiming leverage the semantic content of ontologies to improve knowledge discovery in complex and dynamical domains. Neucom has a solid set of data analysis and modelling tools and its integration with Ontologies and data schemas is proving to be a good path. However, it is still done in a manual way. Furthermore, our mapper requires a lot of interaction with the user and it slow down the process of use previous knowledge from data schemas.

We are implementing a medical case study in which new methods and sources are being used. The current results show us that our approach is very promise and powerful in terms of knowledge discovery and decision support system.

6 Future Directions

Although our approach attacks many of the current problems in the ontology learning area, we can identify one major source of investigation: How to integrate text mining and learning from flat files. We need some tool able to learn from the table name, filed name and its content to infer new knowledge and help the ontology engineer in the process of knowledge acquisition. This direction will guide our research in the next phase.

References

[1] Kasabov, N. (2001). Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. IEEE Transactions on Systems, Man and Cybernetics, 31(1083-4419), 902-918.

[2] Sowa, J.F. (2001) Ontology, Metadata, and Semiotics. Retrievable from the internet 08/07/2003 at: http://users.bestweb.net/~sowa/peirce/ontometa.htm.

[3] Gruber, T. (2002) What is an Ontology? Retrievable from the internet 28/01/2002 at: http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.

[4] Chandrasekaran, B.J., et all. (1999). What are ontologies, and why do we need them? IEEE Intelligent Systems, 1999, 14(1094-7167): 20-26.

[5] Spyns, P., Meersman, R. & Jarrar, M. (2002) Data modelling versus ontology engineering, SIGMOD Record Special Issue on Semantic Web, Database Management and Information Systems, 31.

[6] Asunción G.P. & David M. (2003) Survey of ontology learning methods and techniques. Retrievable from the internet 08/07/2003 at www.ontoweb.org.

[7] Johannesson P. (1994) A Method for Transforming Relational Schemas into Conceptual Schemas. In 10th International Conference on Data Engineering, Ed. M. Rusinkiewicz, pp. 115 - 122, Houston,IEEE Press.

[8] Kashyap, V. (1999). Design and Creation of Ontologies for Environmental Information Retrieval. 12th Workshop on Knowledge Acquisition, Modelling and Management Voyager Inn, Banff, Alberta, Canada. October.

[9] Phillips, J. et all. (2001) Ontology-guided knowledge discovery in databases, Paper presented at the Proceedings of the international conference on Knowledge capture, Victoria, British Columbia, Canada.

[10] Rubin D.L., et all. (2002). Automatic data acquisition into ontologies from pharmacogenetics relational data sources using declarative object definitions and XML. In: Proceedings of the Pacific Symposium on Biology, Lihue, HI, 2002 (Eds. R.B. et all).

[11] Stojanovic, L.; Stojanovic, N.; Volz R. (2002). Migrating data-intensive Web Sites into the Semantic Web. Proceedings of the 17th ACM symposium on applied computing (SAC), ACM Press, 2002, pp. 1100-1107.

[12] Saltz, J. et all.(2002) Towards a Knowledge Base Management System:An Ontology - Aware Database Management System (DBMS).

[13] Halpin T., (2001), Information Modeling and Relational Databases: from conceptual analysis to logical design, Morgan-Kaufmann, San Francisco.

[14] Mitra, S. (2000). Neuro-fuzzy rule generation: survey in soft computing framework. IEEE Transactions on Neural Networks, 2000, 11(3): 748-768.

[15] Kasabov, N. (2002). Evolving connectionist systems for adaptive learning and knowledge discovery: methods, tools, applications. IEEE Intelligent Systems, 2002, 24-28.

[16] www.theneucom.com

[17] Gottgtroy, P. C. M. (2000) A Conceptual Model Proposal for Dynamic Systems Requirements Elicitation Computer Science Department, pp. 168 (Natal, Federal University of Rio Grande do Norte).

[18] http://protege.stanford.edu/index.html