ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/221014133

An Ontology Learning and Reasoning Framework.

Conference Paper · January 2005

Source: DBLP

CITATIONS
5

READS
121

1 author:

Hele-mai Haav
Tallinn University of Technology

51 PUBLICATIONS **279** CITATIONS

SEE PROFILE

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

Project Model Based Java Software Development Technology View project

Project Semantic interoperability of Estonian e-government services View project

All content following this page was uploaded by Hele-mai Haav on 08 January 2015.

The user has requested enhancement of the downloaded file.

An Ontology Learning and Reasoning Framework

Hele-Mai HAAV
Institute of Cybernetics at Tallinn University of Technology,
Akadeemia tee 21, 12618 Tallinn, ESTONIA

Abstract. Most of current ontology languages and methodologies lack of learning support combined with reasoning mechanisms. This motivates us creating an ontology learning and reasoning framework in order to guarantee accuracy, transparency and consistency of ontology representation by automatic or semiautomatic methods of ontology learning and reasoning. Novelty of our approach is in combining ontology learning and reasoning into one framework. For ontology extraction and learning, we use Formal Concept Analysis. For ontology reasoning purposes, we require that learned ontology should be automatically mapped to some logic language. We use predicate logic as a target ontology language in this paper.

1. Introduction and motivation

In recent years, ontology has become a concept in computer science. In most widely used understanding, an ontology is constituted by a specific vocabulary used to describe a certain domain of interest and a set of explicit assumptions regarding the intended meaning of the vocabulary [11]. Assumptions are usually represented in a logic language.

As analysis of existing ontologies and ontology design methodologies shows, it is very difficult for a designer to develop accurate, transparent and consistent ontology [14, 3, 11]. In order to help experts in ontology design process, an initial domain ontology can be automatically or semi-automatically built from domain-specific knowledge captured in domain-specific texts, documents, database schema or data as demonstrated in [4, 12, 6]. This new approach is called *ontology learning*. The approach combines methods from Machine Learning (ML) and manual ontology description as well as NLP.

There is another important issue concerning ontology development languages and methodologies: their reasoning capabilities. *Ontology reasoning* is important for checking concept satisfiability, subsumption (is-a), instances, and classification.

According to recent study by Corcho and Gomez-Perez [1] most of well-known ontology specification languages do not have sound and complete inference engine implemented for them. Only F-logic [9] and OIL [8] from given list of languages supported sound and complete inference mechanism.

Analysis and evaluation of ontology development methodologies [3] result in conclusion that high level design methods do not support well nontrivial ontology-based reasoning. Recent study of DAML ontology library by Tempich and Volz [14] shows that semantic web ontologies are designed in rather heterogeneous way and many semantic web ontologies fail in being usable for inference. This indicates that quality of ontology representation is not high enough. *Consequently, most of current ontology languages and methodologies lack of learning support combined with reasoning mechanisms.*

This motivates us in current research to create an ontology learning and reasoning framework in order to provide assistance to ontology designer in ontology development process. The framework should guarantee accuracy, transparency and consistency of ontology representation by automatic or semiautomatic methods of ontology learning and reasoning. Novelty of our approach is in combining ontology learning and reasoning into one framework.

In this paper, we restrict our research to domain ontologies that capture the essential concepts and relationships among those concepts in particular field of interest (application area). This work is an extension of our previous work on semi-automatic extraction and expression of domain-specific ontologies using Formal Concept Analysis (FCA) [6, 7].

For ontology extraction and *learning*, we use Formal Concept Analysis as a solid mathematical basis. For ontology *reasoning* purposes, we require that learned ontology should be automatically transformed (mapped) to some logic language. We use in this paper predicate logic, but we plan to use also description logics as a target ontology language in our further work.

The rest of the paper is structured as follows. Section 2 gives an overview of Formal Concept Analysis. General ontology learning and reasoning framework is presented in section 3. Section 4 discusses related works and section 5 concludes the paper.

2. Learning Conceptual Structures by using FCA

A reader is referred to [5] for detailed knowledge about Formal Concept Analysis (FCA). In the following we give a very basic introduction to the main principles of FCA by using examples.

For example, a context K(O,C,R) of real estate domain can be as shown as in Table 1. The set of objects O is a collection of real estate domain specific texts (ads) about real estate items denoted by references like A1, A2, etc. in the table 1. The set of attributes C is the set of noun-phrases chunked from these texts by using NLP. If a text describing a real estate item contains certain noun-phrase, then the relationship R holds and we denote it by number 1 in the table below.

Table 1. Real estate domain context

Objects	Attributes (noun-phrases)					
	Real estate	Family house	Country house	Summer house	Blockhouse	Skyscraper
A1	1	1				
A2	1	1	1	1		
A3	1		1			
A4	1				1	1
A5	1				1	
A6	1			1		

A formal concept of the context K(O, C, R) is defined as pair (A, B), where $A \subseteq O$, $B \subseteq C$, $A' = B$ and $B' = A$, where A' is the set of attributes common to all the objects in A and B' is the set of objects having the attributes in B. The extent of the concept (A, B) is A and its intent is B.

For concepts (A1, B1) and (A2, B2) in the set S of all concepts of the context K(O,C,R) we have

$$(A1, B1) \leq (A2, B2) \Leftrightarrow A1 \subseteq A2 \Leftrightarrow B1 \supseteq B2.$$

The relation $\leq$ is an order on S. It is shown in [5] that $(S(K), \leq)$ is a complete lattice and this lattice is known as the concept lattice of the context K(O, C, R).

For example, the following concept lattice corresponds to the context presented in the table 1.

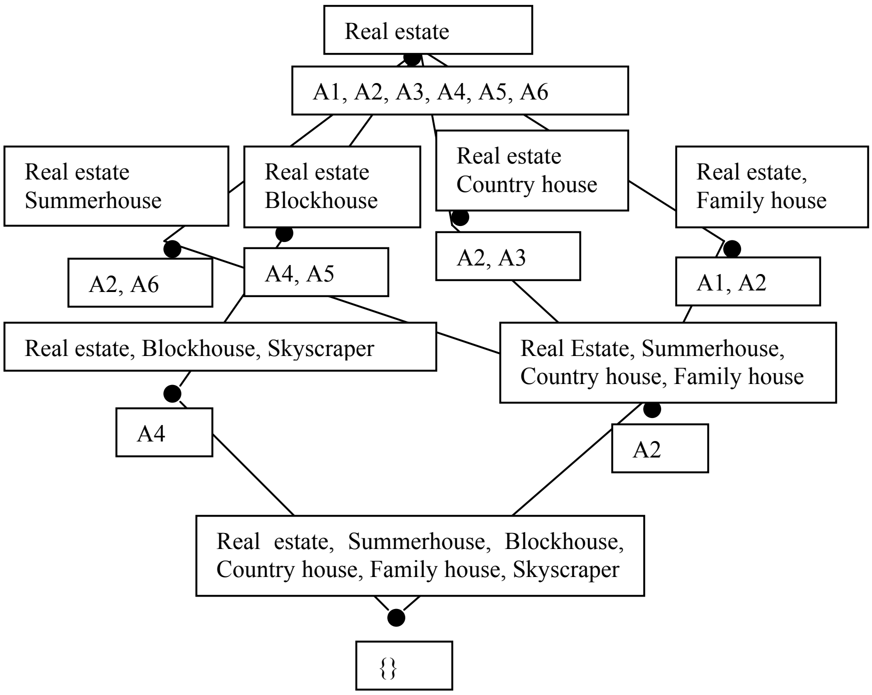

Fig. 1. The concept lattice of a real estate domain

Each node in this lattice (denoted by black circle) is a formal concept. For example, one of the formal concepts of the context described in Table 1 is
{A2, A6} $\times$ *{Real estate, Summerhouse},* where the set *{A2, A6}* is the extent of the concept and the set *{Real estate, Summerhouse}* is its intent. Sub and super-concept relationships between the formal concepts are represented by edges in the Hasse diagram in Fig. 1.

3. An Ontology Learning and Reasoning Framework

The proposed semi-automatic ontology learning and reasoning framework is based on the two concepts introduced in the following sections: concept lattice based ontology representation and its Horn logic model.

3.1 General Schema of the Framework

The framework comprises the following steps:
1. Extracting formal context of a domain from domain-specific texts or data
2. Computing initial ontology as the concept lattice from the context using FCA and reduction procedures
3. Transforming initial ontology to a set of expressions in first order language
4. Extending initial ontology with additional rules and facts

General schema of this method is drawn in the Fig. 2 as follows:

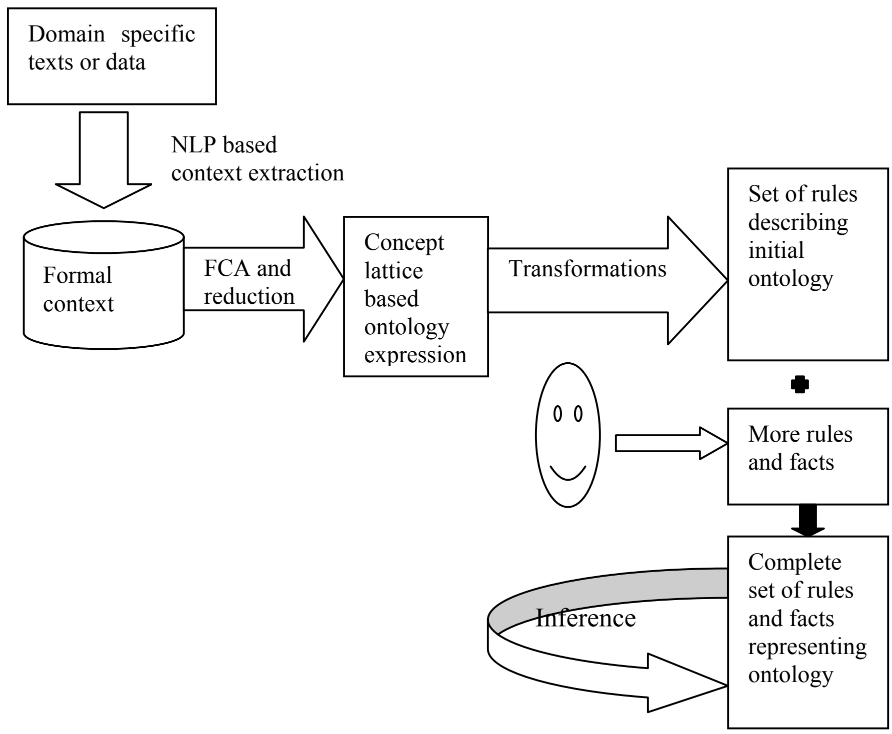

Fig. 2. General schema of ontology learning and reasoning framework

In the following we describe each step of the method in more detail.

3.2 Extraction of a Formal Context

According to the method, first task is to produce a formal context K(O,C,R) for extractable domain-specific ontology. For our approach, objects of formal context can be textual descriptions of domain entities written in some natural language. We assume that those descriptions use domain specific vocabulary and are rather short. For example, suitable descriptions can be ads of real estate items in the real estate web catalogues, descriptions of products in product catalogues, technical descriptions of components.

Attributes of an object for FCA are noun-phrases present in the domain-specific text describing a given domain-specific entity. Binary relationship R between descriptions (texts) of domain entities and noun phrases is discovered during the NLP process of text sources. A resulting set of noun phrases together with references to the domain-specific text sources are stored into the database table, which represents a context for the application domain in the form of binary relationship between descriptions of entities and noun phrases. A reader is referred to an example provided in the previous section, where the table 1 represented context of a sample domain.

3.3 Computing Initial Ontology

As a result of FCA, concept lattice corresponding to the given context is computed. The next step is to reduce the lattice and to perform certain naming procedure as described in the following.

There is redundant information in concept lattice. The two kinds of redundancy can be eliminated from concept lattice without loosing any information: redundant elements in formal concepts intents and redundant objects in formal concepts extents. Our reduction procedure has 2 steps: elimination of redundant elements from formal concepts intents and elimination of lattice of extents. We call the resulting lattice L_{CR} of reduction procedure as ontology lattice. Fig. 3 shows the lattice L_{CR} of our example.

Naming of concepts is done as follows:

1. A concept gets a unique name that is the name of the attribute(s) of formal concepts, which are left after reduction procedure.
2. After the previous naming procedure, there might be nodes that do not get names. In principle, the names for these nodes need to be provided by domain expert or ontology designer. It is possible automatically generate formal names (e.g. c1, c2…) for those nodes and then ask advice from human expert.

The Fig. 3 depicts real estate domain ontology produced from concept lattice shown in Fig.1 using reduction and naming procedures.

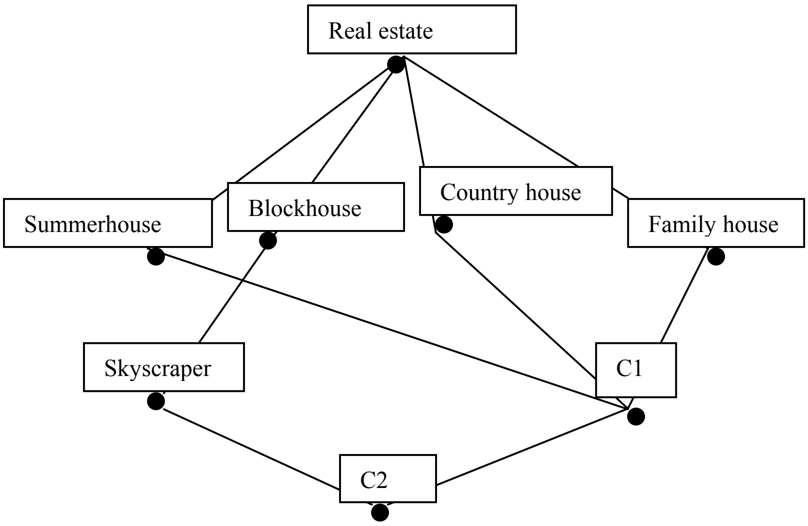

Fig. 3. Ontology lattice

One may notice that 2 nodes in the lattice are denoted by generated concept names C1 and C2. Ontology designer may analyse the lattice above and give the appropriate names to the concepts. It is also interesting that those nodes really denote new unknown (discovered) concepts, because the domain-specific texts did not include any noun-phrases for denoting these concepts. There exist several algorithms for FCA and construction of line diagrams of concept lattices. Excellent comparison of performance of those algorithms can be found in [10].

3.4 Transforming Initial Ontology to Horn Logic

Next step is to provide means for transforming ontology lattice into Horn logic. This process enables to produce logical expression of ontology lattice and specify intended semantics of the descriptions in first order logic.

As our approach uses first order language, then it is possible to attach different ontology inference engines for practical applications by translating ontology lattice logical expression to any inference engine rule language. This was one of the reasons behind choosing Horn logic based rule language. In the following, we provide first order logic model for L_{CR}. The model was first provided by us in [7].

3.4.1 Language Constructs

We use standard syntax for first order logic and define a simple rule language based on Horn clauses as follows.

An alphabet of the rule-language is defined as follows:

1. Set of constants **N** that consists of the set of concept names **C**, names of properties A, and special names **any** (lattice top, empty top is always True) and **nil** (lattice bottom, empty bottom is always False).
2. Set of variable names **V**. Uppercase letters denote the variables in **V**.
3. Set of predicate symbols **P**

An interpretation for the rule-language is defined as a set of ground atoms constructed from predicate names in **P** and constants in **N.**

3.4.2 Mapping of Concept Lattice to Rule Language

The mappings from lattice to rule language are defined as follows.

Mapping concepts

Concepts are represented using their names in ontology as constants in rule language. For example, house, summerhouse etc. If we like to refer to extents, then concepts can also be represented by predicates like house(X). At the moment, we are not interested in extents of concepts.

Mapping taxonomic relationships

The predicate subconcept(Concept1,Concept2) is used to denote that *Concept1* is an immediate subconcept of *Concept2*. Subconcept predicates are automatically generated according to the given lattice L_{CR}.

Predicate isa is used to represent partial order relationship between concepts.

Rules for lattice axioms

As L_{CR} is complete lattice, then the rules for lattice axioms are as follows:

Reflexivity:

isa(Concept, Concept)

Transitivity:

isa(Concept1, Concept2)←subconcept(Concept1, Concept2)

isa(Concept1, Concept2) ← isa(Concept1, Concept3), isa(Concept3, Concept2)

Predicate subconcept denotes that Concept1 is an immediate subconcept of Concept2.

Antisymmetry:

equal(Concept1,Concept2) ← isa(Concept1,Concept2), isa(Concept2,Concept1)

Rules for lattice operations

As L_{CR} is a complete lattice, then for each set of concepts, there exists always a greatest lower bound (glb or greatest common subconcept) and a least upper bound (lub or a least common superconcept). Lattice meet is used to calculate glb and join is operation to calculate lub. We define these lattice meet operation using the following set of rules.

Lattice meet operation

c_subconcept(C, $C_1,\ldots,C_k$)←isa(C, C_1),…, isa(C, C_k)

g_c_subconcept(C, $C_1,\ldots,C_k$) ← c_subconcept(C, $C_1,\ldots,C_k$),
c_subconcept(T, $C_1,\ldots,C_k$), isa(T, C)

The predicate c_subconcept(C, $C_1,\ldots,C_k$) means that the concept C is a common subconcept of the set of concepts $\{C_1,\ldots,C_k\}$

The predicate g_c_subconcept(C, $C_1,\ldots,C_k$) means that the concept C is the greatest common subconcept of the set of concepts $\{C_1,\ldots,C_k\}$.

Symmetrically, we define predicates and rules for join operation, as provided in [7].

Logical model of a given lattice L_{CR} can automatically be generated on the basis of mappings presented above.

The following is a subset of a set of automatically generated ground subconcept atoms for ontology lattice shown in Fig. 3.

subconcept(familyhouse, real-estate)
subconcept(c1, familyhouse)
subconcept(c2, c1)

3.5 Extending Initial Ontology with Additional Rules and Facts

As we have seen, taxonomic relationships between concepts can be automatically generated from a given lattice based ontology expression using logic-based formulation of concept lattice. In order to define non-taxonomic relationships the corresponding groups of predicates and rules are defined.

Properties of concepts

For defining properties of concepts, the following predicate can be used:
hasproperty(Conceptname, Propertyname).

Inheritance of properties

Inheritance of properties can be represented by the following rule:
hasproperty(C1, X)←isa(C1,C2), hasproperty(C2, X)

Ontological relationships

Ontological relationships like part-of, related-to etc can be easily represented via predicates. The following predicates demonstrate opportunities adding other ontological relationships:
partof(C1,C2)
synonyms(C1,C2), etc.

Non-taxonomic relationships give additional possibilities for ontology representation and reasoning about ontology.

3.6 Reasoning about Ontology

Reasoning is important to ensure the quality of design of ontology. It can be used to find contradictory concepts, to derive implied relationships, etc.

Inference rules for lattice axioms and operations can be used to decide taxonomic relationships between concepts as well as perform lattice operations.

For example, to find the least common superconcept of the set of concepts *{summerhouse; countryhouse},* we define the following query:
l_c_superconcept(X, summerhouse, countryhouse).

The answer is the concept *real-estate*. We may be interested in all the superconcepts of the concept *skyscraper*, for example. The query isa(skyscraper, X) gives the list of ground atoms as an answer. In our example, this is the list of the following atoms isa(skyscraper, real-estate) and isa(skyscraper, blockhouse).

Inference about non-taxonomic relationships is made possible due to additional rules. For example, a designer may add the fact hasproperty(blockhouse,no_of_floors). Using inference, we may ask query hasproperty(X, no_of_floors) and receive an answer that also the fact hasproperty(skyscraper, no_of_floors) holds.

4. Remarks on Related Works

There are several attempts on ontology learning and ontology extraction [16, 13, 2, 14, 15]. For example, in KRAFT [13], local ontology is extracted from shared ontology. In the system Infosleuth [16] ontologies are constructed semi-automatically from textual databases. They use ontology-learning approach, where human experts provide a set of initial words denoting concepts from high-level ontology. In [12] discovering non-taxonomic relationships from texts using shallow text processing methods is presented. Their technique is integrated into ontology learning tool TextToOnto. The method presented in [15] uses manually built initial ontology to build conceptual hierarchy semi-automatically from texts. Resulting ontology is presented in Description Logics.

Similarity of our approach and those discussed above is in using texts as descriptions of conceptualisation of domain and learning formal ontology from the given texts. To contrast our approach to the research mentioned above, we like to express that we combine automatic extraction of taxonomic relationships from domain-specific texts with semi-automatic expression of full ontology (including also non-taxonomic relationships) in first-order language for further reasoning and search.

5. Conclusion

We have provided an ontology learning and reasoning framework in order to help ontology designer automatically to extract initial ontology from given set of domain specific texts, to map it automatically to rule language and to use rule language for adding non-taxonomic relationships to the ontology representation.

Our main contribution resides in learning the ontology from the NL texts by using FCA and transforming it to Horn logic for reasoning.

References

[1] Corcho O.and Gómez-Pérez, A., A Roadmap to Ontology Specification Languages, In Lecture Notes on artificial Intelligence. 2nd International Conference on Knowledge Engineering and Knowledge Management. Springer-Verlag. October 2000, 80-96.

[2] Decker S., Erdmann M., Fensel D., and Studer R., Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information, Meersman R. et al. (eds.): Semantic Issues in Multimedia Systems. Proceedings of DS-8. Kluwer Academic Publisher, Boston, 1999, 351-369.

[3] Fernandez-Lopez M. and Gomez-Perez A., Overview and Analysis of Methodologies for Building Ontologies, The Knowledge Engineering Review, Vol. 17:2, 129-156, Cambridge University Press, 2002

[4] Gangemi A, et al, An Overview of the ONIONS Project, Data and Knowledge Engineering, 31, 1999

[5] Ganter, B. and Wille, R., Formal Concept Analysis, Mathematical Foundations, Springer, (1999)

[6] Haav, H.-M.: Learning Ontologies for Domain-specific Information Retrieval, W. Abramowicz (Ed), Knowledge-Based Information Retrieval and Filtering from the Web, Kluwer Academic Publishers, (2003), 285-300

[7] Haav, H.-M., Combining FCA and a Logic Language for Ontology Representations, J. Barzdins (Ed), Databases and Information Systems, Sixth International Baltic Conference on Databases and Information Systems, June 6-9, Riga, 2004, Scientific papers of University of Latvia, Vol 672, (2004), 436-451

[8] Horrocks, I.,et al.: *The Ontology Inference Layer OIL* http://www.ontoknowledge.org/oil/oilhome.shtml

[9] Kifer, M., Lausen, G., and Wu, J., Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of ACM, 1995, 42(4), pp 741-843

[10] Kuznetsov, S. O., and Obiedkov, S. A., Comparing performance of Algorithms for Generating Concept Lattices, Journal of Experimental & Theoretical Artificial Intelligence vol 14(1), 2002, Nos. 2-3, pp 189-216

[11] Maedche, A.: Ontology Learning for the Semantic Web, Kluwer Academic Publishers, (2002)

[12] Maedche, A. and Staab, S.: Discovering Conceptual Relations from Text. ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence, IOS Press, Amsterdam, (2000)

[13] Preece, A. D., Hui, K-J., Gray, W. A., et al., The Kraft arhitecture for knowledge fusion and transformation, In Proc. of 19th SGES Int. Conference, Springer 1999

[14] Tempich, C. and Volz, R.: Towards a benchmark for Semantic Web reasoners-an analysis of the DAML ontology library, Sure Y (ed) Proceedings of Workshop of Evaluation of Ontology-based Tools (EON 2003) at 2nd Int. Semantic Web Conference (ISWC 2003), USA, (2003)

[15] Todirascu, A. and Beuvron, F., Using Description Logics for Ontology Extraction, In: Proceedings of Conference TALN 2000, Lausanne, October 16-18, 2000

[16] Woelk, D. and Tomlinson, C., The Infosleuht project: Intelligent Search Management via Semantic Agents. In Second World Wide Web Conference '94:Mosaic and the Web, 1994

View publication stats