Learning of Ontologies for the Web: the Analysis of Existent Approaches

Borys Omelayenko

Vrije Universiteit Amsterdam,
Division of Mathematics and Computer Science,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Email: borys@cs.vu.nl, URL: www.cs.vu.nl/~borys

Abstract

The next generation of the Web, called Semantic Web, has to improve the Web with semantic (ontological) page annotations to enable knowledge-level querying and searches. Manual construction of these ontologies will require tremendous efforts that force future integration of machine learning with knowledge acquisition to enable highly automated ontology learning. In the paper we present the state of the-art in the field of ontology learning from the Web to see how it can contribute to the task of semantic Web querying. We consider three components of the query processing system: natural language ontologies, domain ontologies and ontology instances. We discuss the requirements for machine learning algorithms to be applied for the learning of the ontologies of each type from the Web documents, and survey the existent ontology learning and other closely related approaches.

Introduction

Nowadays the Internet contains a huge collection of data stored in billions of pages and it is used for the worldwide exchange of information. The pages represent mainly textual data and have no semantic annotation. Thus, query processing based mostly on inefficient keyword-matching techniques becomes a bottleneck of the Web.

Tim Berners-Lee coined the vision of the next version of the Web, called Semantic Web [Berners-Lee&Fischetti, 1999], that would provide much more automated services based on machine-processable semantics of data and heuristics that make use of these metadata. The explicit representation of the semantics of data accompanied by domain theories (i.e. ontologies) will enable a Knowledge Web that provides a qualitatively new level of service. It will weave together a net linking incredibly large segments of human knowledge and complement it with machine processability.

This will require enrichment of the entire Web with lots of ontologies that capture the domain theories. Their manual construction will require enormous human efforts, thus *ontology acquisition becomes a bottleneck of the Semantic Web*.

Recently ontologies have become a hot topic in the areas of knowledge engineering, intelligent information integration, knowledge management, and electronic commerce [Fensel, 2000]. *Ontologies* are knowledge bodies that provide a formal representation of a shared conceptualization of a particular domain. Modern research focus lies in Web-based ontology representation languages based on XML and RDF standards and further application of ontologies on the Web (see [Decker et al., 2000]). *Ontology learning* (OL) is an emerging field aimed at assisting a knowledge engineer in ontology construction and semantic page annotation with the help of machine learning (ML) techniques.

In the next section of the paper we discuss the general scheme for semantic querying of the Web with three ontological components required; the subsequent sections discuss OL tasks and available ML techniques. The survey section describes the applications of ML techniques for the learning of different ontology types, and we conclude with comparison of the approaches.

Semantic Querying of the Web

In this section we discuss the general scheme for semantic querying of the Web, the types of

---

In: Proceedings of the International Workshop on Web Dynamics, held in conj. with the 8th International Conference on Database Theory (ICDT'01), London, UK, 3 January 2001

ontologies involved in query process, and basic ML algorithms available for learning the ontologies.

The General Scheme

The general scheme of the querying process is presented in Figure 1. First, the user formulates the query in natural language. Then the query is transformed into a formal query with the help of the natural language ontology and the domain ontology. The Web pages are (possibly incomplete) instances of some domain ontologies, and they will contain pieces of data semantically marked up according to the underlying domain ontology. The query processor has to find the mapping between the concepts of the initial query, the domain model used to expand the query, and the ontology instances on the Web. This mapping will be non-trivial and will require inference over domain ontologies.

Ontological Components

There are a number of domains where ontologies were successfully applied. The three ontologies that are important for querying the Web (see Figure 1) are:

Natural Language Ontologies (NLO) that contain lexical relations between the language concepts; they are large in size and do not require frequent updates. Usually they represent the background knowledge of the system and are used to expand user's queries. These ontologies belong to so-called 'horizontal' ontologies that try to capture all possible concepts, but they do not provide detailed description of each of the concepts.

Domain ontologies capture knowledge of one particular domain, i.e. pharmacological ontology, or printer ontology. These ontologies provide detailed description of the domain concepts from a restricted domain (so-called 'vertical' ontologies). Usually they are constructed manually but different learning techniques can assist the (especially inexperienced) knowledge engineer.

Ontology instances represent the main piece of knowledge presented in the future Semantic Web. As today's Web is full of HTML documents of different layout, the future Web will be full of instances of different domain ontologies. The ontology instances will serve as the Web pages and will contain the links to other instances (similar to the links to other Web pages). They can be generated automatically and frequently updated (i.e. a company profile from the Yellow Pages catalogue will be updated frequently while the ontology of the catalogue will remain the same).

The Semantic Web will require creation and maintenance of a huge number of the ontologies of all three types, and the following ontology learning tasks will become important.

Ontology Learning Tasks

Previous research in the area of ontology acquisition proposed lots of guidelines for manual ontology development (see [Lopez, 1999] for an overview) that organize the work of the knowledge engineer, but they pay no attention to the process of the acquiring of the ontology by humans. The human experts have to evolve the best knowledge acquisition process themselves from their past experience acquired by passing through numerous case studies. Thus, we have to separate several tasks in OL on our own:

Ontology creation from scratch by the knowledge engineer. In this task ML assists the knowledge engineer by suggesting the most important relations in the field or checking and verifying the constructed knowledge bases.

Ontology schema extraction from Web documents. In this task ML systems take the data and meta-knowledge (like a meta-ontology) as input and generate the ready-to-use ontology as output with the possible help of the knowledge engineer.

Extraction of ontology instances populates given ontology schemas and extracts the instances of the ontology presented in the Web documents. This task is similar to information extraction and page annotation and can apply the techniques developed in these areas.

Ontology integration and navigation deals with reconstructing and navigating in large and possibly machine-learned knowledge bases. For example, the task can be to change the propositional-level knowledge base of the machine learner into a first-order knowledge base.

Ontology update task updates some parts of the ontology that are designed to be updated (like formatting tags that have to track the changes made in the page layout).

Ontology enrichment (or ontology tuning) includes automated modification of minor relations into existing ontology. This does not change major concepts and structures but makes the ontology more precise. Unlike ontology update, this task deals with the relations that are not specially designed to be updated.

The first three tasks relate to ontology acquisition tasks in knowledge engineering, and the next three to ontology maintenance tasks. In this paper we do not consider ontology integration and ontology update tasks.

Machine Learning Techniques

The main requirement for ontology representation is that ontologies must be symbolic, human-readable and understandable. This forces us to deal only with symbolic learning algorithms that make generalizations and skip other methods, like neural networks, genetic algorithms and the family of 'lazy learners' (see [Mitchell, 1997] for an introduction to ML and the algorithms mentioned below). The foreseen potentially applicable ML algorithms include:

Propositional rule learning algorithms that learn association rules, or other attribute-value rules. The algorithms are generally based on a greedy search of the attribute-value tests that can be added to the rule preserving its consistency with the set of training instances. Decision tree learning algorithms, mostly represented by the C4.5 algorithm and its modifications, are used quite often to produce high-quality propositional-level rules. The algorithm uses statistical heuristics over the training instances, like entropy, that guide hill-climbing search of the decision trees. Learned decision trees are equivalent to the sets of propositional-level classification rules that are conjunctions of attribute-value tests.

Bayesian learning is mostly represented by Naive Bayes classifier. It is based on the Bayes theorem and generates probabilistic attribute-value rules based on the assumption of conditional independence between the attributes of the training instances.

First-order logic rules learning induces the rules that contain variables, called first-order Horn clauses. The algorithms usually belong to the FOIL family of algorithms and perform general-to-specific hill-climbing search for the rules that cover all available positive training instances. With each iteration it adds one more literal to specialize the rule until it avoids all negative instances.

Clustering algorithms group the instances together based on the similarity or distance measures between a pair of instances defined in terms of their attribute values. Various search strategies can guide the clustering process. Iterative application of the algorithm will produce hierarchical structures of the concepts.

The knowledge bases built by ML techniques substantially differ from the knowledge bases that we call ontologies. The differences are inspired by the fact that ontologies are constructed to be used by humans, while ML knowledge bases are only used automatically. This leads to several differences listed in Table 1.

To enable automatic OL we must adapt ML techniques so that they can automatically construct ontologies with the properties of manually constructed ontologies. Thus, OL techniques have to possess the following properties, which we trace in the survey:

- ability to interact with a human to acquire his

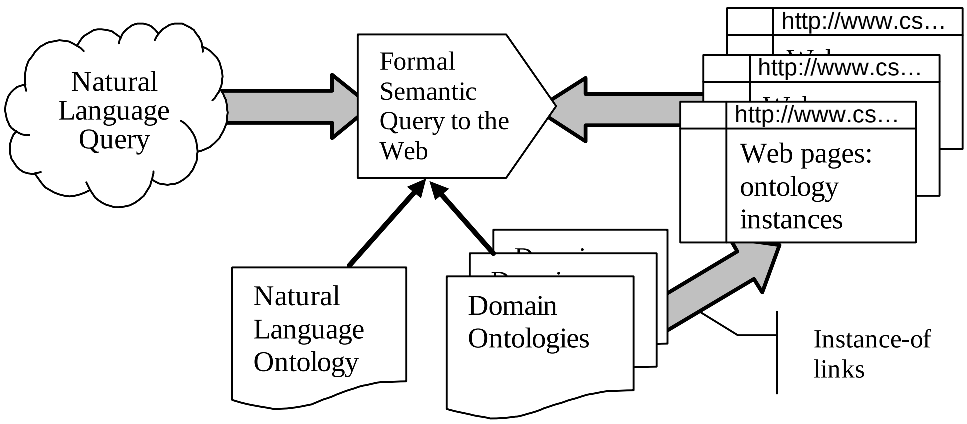

Figure 1. Semantic querying of the Web

knowledge and to assist him; this requires readability of internal and external results of the learner;
- ability to use complex modelling primitives;
- ability to deal with complex solution space, including composed solutions.

Each ontology type has special requirements for ML algorithms applied for learning these types of ontologies.

Table 1. Manual and machine representations

Machine-learned knowledge bases	Manually constructed ontologies
Modelling primitives	
Simple and limited. For example, decision tree learning algorithms generate the rules in the form of conjunctions over attribute-value tests.	Rich set of modelling primitives (frames, subclass relation, rules with rich set of operations, functions, etc.).
Knowledge base structure	
Flat and homogeneous.	Hierarchical, consists of various components with subclass-of, part-of and other relations.
Tasks	
Classification and clusterization that map the objects described by the attribute-value pairs into a limited and unstructured set of class or cluster labels.	Classification task requires mapping of objects into a tree of structured classes. It can require construction of class descriptions instead of selection.
Problem-solving methods	
Very primitive, based on simple search strategies, like hill-climbing in decision tree learning.	Complicated, require inference over a knowledge base with a rich structure, often domain-specific and application-specific.
Solution space	
The non-extensible, fixed set of class labels.	Extensible set of primitive and compound solutions.
Readability of the knowledge bases to a human	
Not required. They can be used only automatically and only in special domains.	Required. They may be (at least potentially) used by humans.

NLO contain hierarchical clustering of the language concepts (words and their senses). The set of relations (slots) used in the representation is limited. The main relations between the concepts are: 'synonyms', 'antonyms', 'is-a', 'part-of'. The verbs can contain several additional relations to describe the actions. Concept features are usually represented by adjectives or adjective nouns (like 'strong-strength'). Thus the ontology can be represented by frames with a limited structure.

NLOs define the first and basic interpretation of user's query, and they must link the query to specific terminology and specific domain ontology. General language knowledge contained in a general-purpose NLO like WordNet [Fellbaum, 1998] is not sufficient for such a purpose. In order to achieve this, lots of research efforts have been focused on NLO enrichment. NLO enrichment from domain texts is a suitable task for ML algorithms, because it provides a good set of training data for the learner (the corpus).

NLOs do not require either frequent or automatic updates. They are updated from time to time with intensive cooperation from a human, thus ML algorithms for NLO learning are not required to be fast.

Domain ontologies use the whole set of modelling primitives, like (multiple) inheritance, numerous slots and relations, etc. They are complex in structure and are usually constructed manually. Domain ontology learning concentrates on discovering statistically valid patterns in the data in order to suggest them to the knowledge engineer who guides the ontology acquisition process. In future we would like to see an ML system that guides this process and asks the human to validate pieces of the constructed ontology.

ML will be used to predict the changes made by the human to reduce the number of interactions. The input of this learner will consist of the ontology being constructed, human suggestions and domain knowledge.

Domain ontologies require more frequent updates than NLOs (just as new technical objects appear before the community has agreed about the surrounding terminology), their updates are done manually and ML algorithms that assist this process are also not required to be fast.

Ontology instances are contained in the Web pages marked up with the concepts of the underlying domain ontology with information extraction or annotation rules. The instances will require more frequent updates than domain ontologies or NLOs (i.e. a company profile in a catalogue will be updated faster than the ontology of a company catalogue).

The Survey

This section presents the survey of existing techniques related to the learning and enriching of the NLO from the Web, Web-based support for domain ontology construction, and extraction of ontology instances. These approaches cover various issues in the field and show different applications of ML techniques.

Learning of NLO

Lots of conceptual clustering methods can be used for ontology construction but no methodology or tool has been developed to support the elaboration of conceptual clustering methods that build task-specific ontologies. The Mo'K tool [Bisson et al., 2000] supports development of conceptual clustering methods for ontology building. The paper focuses on elaboration of the clustering methods to perform human-assisted learning of conceptual hierarchies from corpora. The input for the clustering methods is represented by the classes (nouns) and their attributes (grammatical relations) received after syntactical analysis of the corpora, which are in turn characterized by the frequency with which they occur in the corpora.

The algorithm uses bottom-up clustering to group 'similar' objects to create the classes and to subsequently group 'similar' classes to form the hierarchy. The user may adjust several parameters of the process to improve performance: select input examples and their attributes, level of pruning, and distance evaluation functions. The paper presents an experimental study that illustrates how learning quality depends on the different combinations of parameters.

While the system allows the user to tune its parameters, it performs no interactions during clustering. It builds the hierarchy of the frames that contain lexical knowledge about the concepts. The input corpora can be naturally found on the Web, and the next paper presents a way of integrating NLO enrichment with the Web search of the relevant texts.

The system [Agirre et al., 2000] exploits the text from the Web to enrich the concepts in the WordNet [Fellbaum, 1998] ontology. The proposed method constructs lists of topically related words for each concept in the WordNet, where each word sense has one associated list of related words. For example, the word 'waiter' has two senses: the waiter in the restaurant (related words: waiter–restaurant, menu, dinner); and a person who waits (related words: waiter–station, airport, hospital). The system queries the Web for the documents related to each concept from the WordNet and then builds a list of words associated with the topic. The lists are called topic signatures and contain the weight (called strength) of each word. The documents are retrieved by querying the Web with the AltaVista search engine by asking for the documents that contain the words related to a particular sense and *do not* contain the words related to the other senses of the word. A typical query may look something like 'waiter AND (restaurant OR menu) AND NOT (station OR airport)' to get the documents that correspond to the 'waiter, server' concept.

NLOs, like EuroWordNet or WordNet, help in the understanding of natural language queries and in bringing semantics to the Web. But in specific domains general language knowledge becomes insufficient and that requires creation of domain-specific NLOs. Early attempts to create such domain ontologies to perform information extraction from texts failed because the experts used to create the ontologies with lots of a priori information that was not reflected in the texts. The paper [Faure&Poibeau, 2000] suggests improving NLO by unsupervised domain-specific clustering of texts from corpora. The system Asium described in the paper cooperatively learns semantic knowledge from texts which are syntactically parsed, without previous manual processing. It uses the syntactic parser Sylex to generate the syntactical structure of the texts. Asium uses only head nouns of complements and links to verbs and performs bottom-up breadth-first conceptual clustering of the corpora to form the concepts of ontology level. On each level it allows the expert to validate and/or label the concepts. The system generalizes the concepts that occur in the same role in the texts and uses generalized concepts to represent the verbs.

Thus, state of the art in NLO learning looks quite optimistic: not only does a stable general-purpose NLO exist but so do techniques for automatically or semiautomatically constructing and enriching domain-specific NLO.

Learning of Domain Ontologies

Domain-specific NLO significantly improves semantic Web querying but in specific domains general language knowledge becomes insufficient and query processing requires special domain ontologies.

The paper [Maedche&Staab, 2000] presents an algorithm for semiautomatic ontology learning from texts. The learner uses a kind of algorithm for discovering generalized association rules. The input data for the learner is a set of transactions, each of which consists of a set of items that appear together in the transaction. The algorithm extracts association rules represented by sets of items that occur together sufficiently often and presents the rules to the knowledge engineer. For example, shopping transactions may include the items purchased together. The association rule may say that 'snacks are purchased together with drinks' rather than 'crisps are purchased with beer'. The algorithm uses two parameters: support and confidence for a rule. Support is the percentage of transactions that contain all the items mentioned in the rule, and confidence for the rule *X*? *Y* is conditional percentage of transactions where *Y* is seen, given that *X* also appeared in the transaction. The ontology learner [Maedche&Staab, 2000] applies this method straightforwardly for ontology learning from texts to support the knowledge engineer in the ontology acquisition environment.

The main problem in applying ML algorithms for OL is that the knowledge bases constructed by the ML algorithms have a flat homogeneous structure, and very often have prepositional level representation (see Table 1). Thus several efforts focus on improving ML algorithms in terms of ability to work with complicated structures.

The first step in applying ML techniques to discover hierarchical relations between textually described classes is taken with the help of Ripple-Down Rules [Suryanto&Compton, 2000]. The authors start with the discovery of the class relations between classification rules. Three basic relations are considered: intersection (called subsumption in marginal cases) of classes, mutual-exclusivity, and similarity. For each possible relation they define a measure to evaluate the degree of subsumption, mutual exclusivity, and similarity between the classes. For input, the measures use the attributes of the rules that lead to the classes. After the measures between all classes have been discovered, simple techniques can be used to create the hierarchical (taxonomic) relations between the classes.

Knowledge extraction from the Web (data mining from the Web) uses domain ontologies to represent the extracted knowledge to the user of the knowledge in terms of the common understanding of the domain, i.e. in the terms of the domain ontology.

The system for ontology-based induction of high-level classification rules [Taylor et al., 1997] goes further and uses ontologies not only to explain the discovered rules for a user, but also to guide learning algorithms. The algorithm consequently generates queries for an external learner ParkaDB, that uses the domain ontology and the input data to check consistency of the query, and consistent queries become classification rules. The query generation process continues until the set of queries covers the whole data set. Currently the domain ontologies used there are restricted to simple concept hierarchies where each attribute has its own hierarchy of concepts. On the bottom level the hierarchy contains attribute values present in the data, the next level contains a generalization about these attribute values. This forms one-dimensional concepts, and a domain ontology of a very specialized type.

The approach uses a knowledge-base system and its inference engine to validate classification rules. It generates the rules in terms of the underlying ontology, where the ontology still has a very restricted type.

The paper [Webb, Wells, Zheng, 1999] experimentally demonstrates how the integration of machine learning techniques with knowledge acquisition from experts can both improve the accuracy of the developed domain ontology and reduce development time. The paper analyses three types of knowledge acquisition system: the systems for manual knowledge acquisition from experts, ML systems and the integrated systems built for two domains. The knowledge bases were developed by experienced computer users who were novices in knowledge engineering.

The knowledge representation scheme was restricted to flat attribute-value classification rules and the knowledge base was restricted to a set of production rules. The rationale behind this restriction was based on the difficulties that novice users experience when working with first-order representations. The ML system used the C4.5 decision tree learning algorithm to support the knowledge engineer and to construct the knowledge bases automatically.

The use of machine learning with knowledge acquisition by experts led to the production of more accurate rules in significantly less time than knowledge acquisition alone (up to eight times less). The complexity of the constructed knowledge bases was mostly the same for all systems. The questionnaire presented in the paper showed that the users found the ML facilities useful and thought that they made the knowledge acquisition process easier.

Future prospects for research listed in [Webb, Wells, Zheng, 1999] were to lead to ‘a more ambitious extension of this type of study that would examine larger scale tasks that included the formulation of appropriate ontologies’.

Learning of the domain ontologies is far less developed than NLO improvement. **The acquisition of the domain ontologies is still guided by a human knowledge engineer, and automated learning techniques play a minor role in knowledge acquisition.** They have to find statistically valid dependencies in the domain texts and suggest them to the knowledge engineer.

Learning of Ontology Instances

In this subsection we survey several methods for learning of the ontology instances.

The traditional propositional-level ML approach represents knowledge about the individuals as a list of attributes, with each individual being represented by a set of attribute-value pairs. The structure of ontology instances is too rich to be adequately captured by such a representation. The paper [Bowers et al., 2000] uses a typed, higher-order logic to represent the knowledge about the individuals.

In a classical setting the algorithm C4.5 will take the instances described by attribute-value pairs and produce a tree with nodes that are attribute-value tests. The authors propose replacing the attribute-value dictionary with a more expressive one that consists of simple data types, tuples, sets, and graphs. The method [Bowers et al., 2000] uses a modified C4.5 learner to generate a classification tree that consists of tests on these structures, as opposed to attribute value tests in a classical setting. Experiments showed that on the data sets with structured instances the performance of this algorithm is comparable to standard C4.5 but task-oriented modifications of C4.5 perform much better.

The system CRYSTAL [Soderland et al., 1995] extends the ideas of the previous system AutoSlog, which showed great performance increase (about 200 times better than the manual system) on a creation of concept node definitions for a terrorism domain. It uses an even richer set of modelling primitives and creates the text extraction and mark-up rules, with a given domain model as input, by generalizing semantic mark-up of the manually marked-up training corpora. Manually created mark-up is automatically converted into a set of case frames called ‘concept nodes’ using a dictionary of rules that can be present in the concept node. The concept nodes represent the ontology instances and the domain-specific dictionary of rules defines the list of allowable slots in the ontology instance.

Table 2. Comparison of the ontology learning approaches

Approach	Type			OL Task				ML technique				Modifications of ML techniques		
	NLO	Domain Ontologies	Ontology Instances	Creation	Extraction	Instance Extraction	Enrichment	Propositional learn.	Bayesian learning	First-Order Rule learn.	Clustering	Human interaction	Complex modelling primitives	Complex solution space
[Bisson et al., 2000]	X				X						X	Partial	No	Concept hierarchy
[Faure&Poibeau, 2000]	X						X				X	Yes	Simplified frames	Simplified frames
[Agirre et al., 2000]	X						X				X	No	No	No
[Junker et al., 1999]			X			X				X		No	Several predicates	No
[Craven et al., 2000]			X			X			X	X		No	No	No
[Bowers et al., 2000]			X			X		X				No	Yes, rich structure	Yes, rich structure
[Taylor et al., 1997]		X			X			X				No	Yes, but restricted	No
[Webb, Wells, Zheng, 1999]		X		X				X				Yes	No	No
[Soderland et al., 1995]		X			X	X		X			X	No	Yes	Yes
[Maedche&Staab, 2000]		X			X			X				No	No	No

After formalizing the instance level of the hierarchy, CRYSTAL performs a search-based generalization of the concept nodes. A pair of nodes is generalized by creating a parent class with the attributes that both classes have in common.

The knowledge representation language for the concept nodes is very expressive, which leads to an enormous branching factor for the search performed during the generalization. The system stores the concept nodes in a way that best suits the distance measure function, and therefore performs reasonably efficiently. Experiments on a medical domain showed that the number of positive training instances required for a good recall was limited; after between 1 and 2 thousand, recall measure no longer grows significantly.

The system performs two stages necessary for OL: it formalizes ontology instances from the text and generates a concept hierarchy from these instances.

A systematic study of the extraction of ontology instances from the Web documents was carried out in the project Web-KB [Craven et al., 2000]. In their paper the authors used the ontology of an academic web-site to populate it with actual instances and relations from CS departments' web sites. The paper targets three learning tasks:

(1) recognizing class instances from the hypertext documents guided by the ontology;

(2) recognizing relation instances from the chains of hyperlinks;

(3) recognizing class and relations instances from the pieces of hypertext.

The tasks are dealt with using two supervised learning approaches: Naive Bayes algorithm and first-order rule learner (modified FOIL).

The system automatically creates mapping between the manually constructed domain ontology and the Web pages by generalizing from the training instances. The system performance was surprisingly good for the restricted domain of a CS website where it was tested.

Major ML techniques applied for text categorization performed to some degree of effectiveness [Junker et al., 1999], but beyond that, effectiveness appeared difficult to attain and was only possible in a small number of isolated cases with substantial heuristic modification of the learners. This shows the need for combining these modifications in a single framework based on first-order rule learning.

The paper [Junker et al., 1999] defines three basic types (one for text, one for word, and one for text position) and three predicates governing these types for treating text categorization rules as logical programs and applying first-order rule learning algorithms. The rules learned are derived from five basic constructs of a logical pattern language used in the framework to define the ontologies. The learned rules are directly exploited in automated annotation of the documents to become the ontology instances.

The task of learning of the ontology instances fits nicely into an ML framework, and there are several successful applications of ML algorithms for this. But these applications are either strictly dependent on the domain ontology or populate the mark-up without relating to any domain theory. A general-purpose technique for extracting ontology instances from texts given the domain ontology as input has still not been developed.

Conclusions

The above case study is summarized in Table 2. The first column specifies the approach; the next columns represent the ontological component of the Web query system, the OL tasks, and the relevant ML technique respectively. The last three columns describe the degree to which the system interacts with the user and the properties of the knowledge representation scheme.

From the table we see that a number of systems related to the natural-language domain deal with domain-specific tuning and enrichment of the NLOs with various clustering techniques.

Learning of the domain ontologies is done by now only on a propositional level, and first-order representations are used only in the extraction of ontology instances (see Table 2).

There are several approaches in the field of domain ontology extraction, but the systems used there are the variants of propositional-level ML algorithms.

Each OL paper modifies the applied ML algorithm to handle human interaction, complex modelling primitives or complex solution space together. Only one paper [Faure&Poibeau, 2000] makes all three modifications of the ML algorithm for NLO learning, as also shown in the table.

The research in OL goes mostly in the way of straightforward application of the ML algorithms. This was a successful strategy for beginning, but we would need substantial modifications of the ML algorithms for OL tasks.

Acknowledgements

The author would like to thank Dieter Fensel for helpful discussions and comments, and Heiner Stuckenschmidt and four anonymous reviewers for their comments.

References

[Agirre et al., 2000] **E. Agirre, O. Ansa, E. Hovy, D. Martinez:** Enriching very large ontologies using the WWW. *In: S. Staab, A. Maedche, C. Nedellec, P. Wiemer-Hastings (eds.), Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI'00*, Berlin, Germany, August 20-25, 2000.

[Berners-Lee&Fischetti, 1999] **T. Berners-Lee, M. Fischetti.** *Weaving the Web*. Harper, San Francisco, 1999.

[Bisson et al., 2000] **G. Bisson, C. Nedellec, D. Canamero:** Designing Clustering Methods for Ontology Building - The Mo'K Workbench. *In: S. Staab, A. Maedche, C. Nedellec, P. Wiemer-Hastings (eds.), Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI'00*, Berlin, Germany, August 20-25, 2000.

[Bowers et al., 2000] **A. Bowers, C. Giraud-Carrier, J. Lloyd:** Classification of Individuals with Complex Structure. *In: Proceedings of the Seventeenth International Conference on Machine Learning (ICML'2000)*, Stanford, US, June 29-July 2, 2000, pp. 81-88.

[Craven et al., 2000] **M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, S. Slattery:** Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118: 69-113, 2000.

[Decker et al., 2000] **S. Decker, D. Fensel, F. van Harmelen, I. Horrocks, S. Melnik, M. Klein, J. Broekstra:** Knowledge Representation on the Web. *In: Proceedings of the 2000 International Workshop on Description Logics (DL2000)*, Aachen, Germany, August, 2000.

[Evett, 1994] **M. Evett.** PARKA: A System for Massively Parallel Knowledge Representation. *Computer Science Department, University of Maryland at College Park*, 1994.

[Faure&Poibeau, 2000] **D. Faure, T. Poibeau:** First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX. *In: S. Staab, A. Maedche, C. Nedellec, P. Wiemer-Hastings (eds.), Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI'00*, Berlin, Germany, August 20-25, 2000.

[Fellbaum, 1998] **C. Fellbaum.** *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[Fensel, 2000] **D. Fensel.** *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, Berlin, 2000.

[Junker et al., 1999] **M. Junker, M. Sintek, M. Rinck:** Learning for Text Categorization and Information Extraction with ILP. *In: J. Cussens (eds.), Proceedings of the 1st Workshop on Learning Language in Logic*, Bled, Slovenia, June 1999, 1999, pp. 84-93.

[Lopez, 1999] **F. Lopez:** Overview of Methodologies for Building Ontologies. *In: Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods*, Stockholm, Sweden, August 2, 1999.

[Maedche&Staab, 2000] **A. Maedche, S. Staab:** Semi-automatic Engineering of Ontologies from Text. *In: Proceedings of the Twelfth International Conference on Software Engineering and Knowledge Engineering*, Chicago, 2000.

[Mitchell, 1997] **T. Mitchell.** *Machine Learning*. McGraw Hill, 1997.

[Soderland et al., 1995] **S. Soderland, D. Fisher, J. Aseltine, W. Lehnert:** Issues in Inductive Learning of Domain-Specific Text Extraction Rules. *In: Proceedings of the Workshop on New Approaches* *to Learning for Natural Language Processing at IJCAI'95*, Montreal, Quebec, Canada, 1995.

[Studer et al., 1998] **R. Studer, R. Benjamins, D. Fensel:** Knowledge Engineering: Principles and methods. *Data and Knowledge Engineering*, 25: 161-197, 1998.

[Suryanto&Compton, 2000] **H. Suryanto, P. Compton:** Learning Classification taxonomies from a classification knowledge based system. *In: S. Staab, A. Maedche, C. Nedellec, P. Wiemer-Hastings (eds.), Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI'00*, Berlin, Germany, August 20-25, 2000.

[Taylor et al., 1997] **M. Taylor, K. Stoffel, J. Hendler:** Ontology-based Induction of High Level Classification Rules. *In: Proceedings of the SIGMOD Data Mining and Knowledge Discovery workshop*, Tuscon, Arizona, 1997.

[Webb, Wells, Zheng, 1999] **G. Webb, J. Wells, Z. Zheng:** An Experimental Evaluation of Integrating Machine Learning with Knowledge Acquisition. *Machine Learning*, 31(1): 5-23, 1999.