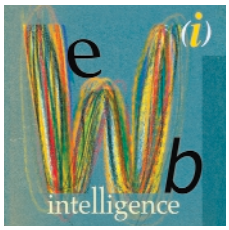
Integrated Approach to Web Ontology Learning and Engineering



The authors have built a software environment that supports the construction and assessment of a domain ontology for intelligent information integration within a virtual user community.

Michele Missikoff
IASI-CNR

Roberto Navigli
Paola Velardi
University of Rome

The development of the *semantic Web*[1]—which seeks to improve the semantic awareness of computers connected via the Internet—requires a systematic, computer-oriented representation of the world. Researchers often refer to such a world model as an *ontology*.

Despite the significant amount of work done on them in recent years, ontologies have yet to be widely applied and used. Research to date has mainly addressed the basic principles, such as knowledge representation formalisms, devoting only limited attention to more practical issues such as techniques and tools aimed at an ontology's actual construction and content.

We have developed a software environment, centered around the OntoLearn tool, that can build and assess a domain ontology for intelligent information integration within a virtual user community. Further, we have tested OntoLearn in two European projects, where it functioned as the basis for a semantic interoperability platform used by small- and medium-sized tourism enterprises.

ONTOLOGY ENGINEERING

Our approach to ontology engineering uses an iterative process that involves *automatic concept learning* with OntoLearn, machine-supported *concept validation* with Consys,[2] and *management* with SymOntoX.[3]

The engineering process starts with OntoLearn exploring available documents and related Web sites to learn domain concepts and detect taxonomic relations among them, producing as output a *domain concept forest*. Initially, we base concept learning on external, generic knowledge sources. In subsequent cycles, the domain ontology receives progressively more use as it becomes adequately populated. The self-learning cycle in Figure 1 shows this process.

Next, we undertake ontology validation with Consys, a Web-based groupware package that performs consensus building by thoroughly validating representatives of the communities active in the application domain. Throughout the cycle, OntoLearn operates in connection with SymOntoX. Ontology engineers can use this management system to define concepts and their mutual connections, thus allowing construction of a semantic net. Further, SymOntoX's environment can automatically attach learned domain concept trees under the appropriate nodes of the upper-domain ontology, thereby enriching concepts with additional information. SymOntoX also performs consistency checks.

Figure 2 shows OntoLearn's system architecture, which supports a three-phase process. First, the system extracts a domain terminology from texts available in the application domain—usually drawn from specialized Web sites or documents exchanged among members of a virtual community. The system then filters this information through a natural

0018-9162/02/$17.00 © 2002 IEEE

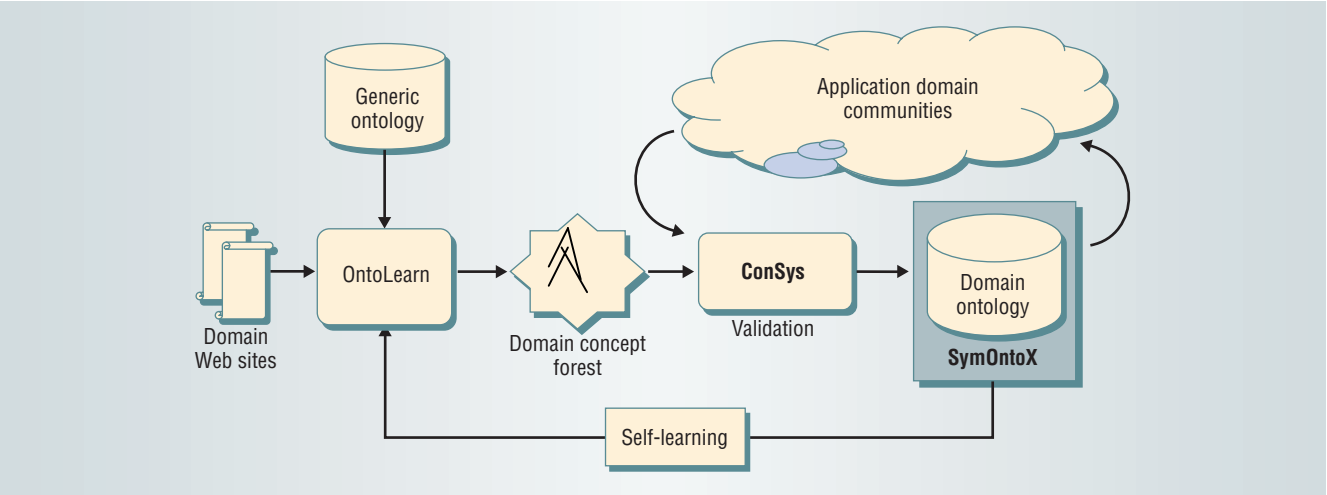


Figure 1. Ontology engineering chain. The self-learning process consists of two cycles. First, knowledge engineers use SymOntoX to update and automatically correct the learned domain ontology, which OntoLearn uses to learn new concepts from new documents. Second, domain users and experts use ConSys to validate the automatically learned ontology, then forward their suggestions to the knowledge engineers, who implement them as updates to SymOntoX.

language processor and applies several statistical techniques that use documents from different domains to perform contrastive analysis. This analysis identifies terminology used in the target domain but not seen in other domains.[4]

Second, the system semantically interprets terms, associating unambiguous concept identifiers with the extracted terms. Semantic interpretation is initially based on external, nondomain-specific knowledge sources. In our work, we use WordNet[5] and the SemCor semantically tagged corpus[6] to perform this function.

Third, we detect taxonomic and similarity relations among concepts, then generate a domain concept forest. We use SymOntoX to perform ontology matching by integrating the DCF with the existing upper ontology.

As Figure 2 shows, the first two modules seek to capture the domain terminology from available document repositories and Web documents. Terminology is often considered the surface realization of relevant domain concepts. However, in the absence of semantic interpretation, we cannot fully capture important semantic relationships between concepts, such as the *kind-of* generalization relation between *bus service* and *public transport service*, or the *instrument* relation between *bus* and *service*, which reveals that *bus service* is a service performed using a bus.

Capturing these relations—especially *kindship* relations—is clearly important for any ontology-based Web application. With these relations defined and in place, for example, a user could query the network for *hotel facility* and retrieve service descriptions that do not explicitly mention that term, but that do include terms such as *swimming pool* or *conference room*. Thus, OntoLearn's key function is semantic interpretation.

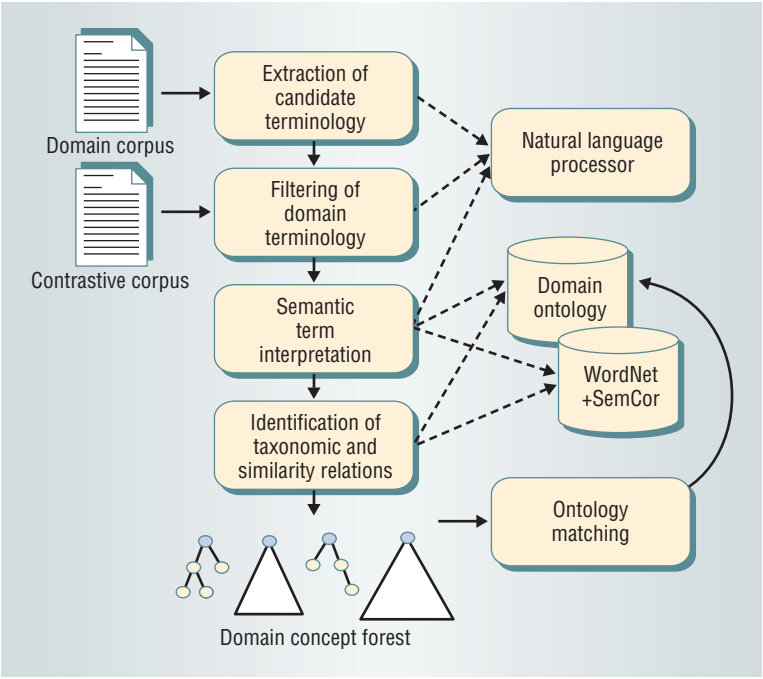


Figure 2. OntoLearn's architecture supports a three-phase process of terminology extraction and filtering, semantic interpretation, and domain-concept-forest generation.

INTERPRETING TERMS SEMANTICALLY

Semantic interpretation associates an appropriate and unique concept identifier with each term in the ontology. Thus, even though the upper domain ontology does not include a given term, it can include a conceptual entry for the various senses of an atomic word—or for some substring of the term. For example, although the ontology associates no concepts with the complex term *room service*, it can include concept descriptions for the words *room* and *service* individually. Therefore, it should be possible to cre-

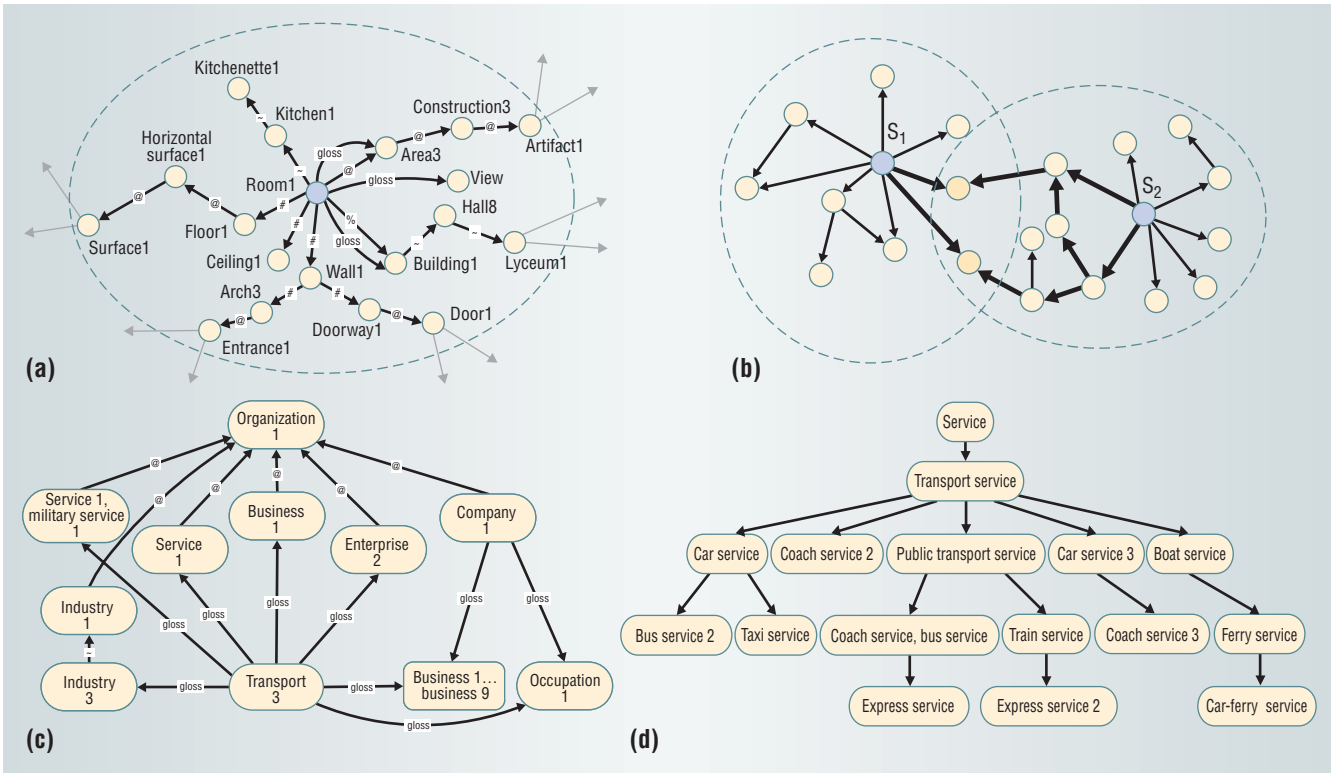


Figure 3. Semantic interpretation in OntoLearn. (a) A semantic net for the term* room1*; (b) intersecting semantic nets for two sense combinations; (c) common semantic patterns among the concepts transport3 and company1; and (d) a resulting domain concept tree rooted in service1.

ate a definition for a given complex term by selecting the appropriate definition for each term component.

If the initial upper domain ontology is very small, we can work from a general-purpose ontology such as WordNet. WordNet uniquely identifies a word sense in two ways: with a set of terms called *synset* and a textual definition called *gloss*. For example, for the third sense of *transport*, the synset list would consist of the words *transportation*, *shipping*, and *transport*. The gloss textual definition of that third sense would be "the commercial enterprise of transporting goods and materials." WordNet codes other types of semantic relations as well, such as kind-of, part-of, and several types of similarity relations.

WordNet includes more than 120,000 words and 170,000 synsets, but very few domain terms. For example, the terms *transport* and *company* appear individually in WordNet, but the unique term *transport company* does not. The word transport has five senses as a noun, while company has nine, which yields 45 possible sense combinations for these two words.

Thus, when performing semantic interpretation we seek to identify the correct sense combination, as Figure 3 shows.[4]

Figure 3a shows that for each sense of each word in a term, OntoLearn automatically creates a semantic net—in this case, for the first sense of *room*. Symbols on each arrow identify the type of semantic relation: kind-of relations appear with an @ symbol, part-of relations with a # symbol, and gloss definitions with the word gloss.

Figure 3b shows that OntoLearn evaluates alternative sense combinations by intersecting and weighting common semantic patterns until it selects the best sense combinations. Once the individual terminology strings have been semantically interpreted, WordNet's kind-of and similarity relations are used to organize concepts into domain concept trees. Figure 3d shows one such tree rooted in the first sense of *service*: "work done by one person or group that benefits another."

Notice that the interpretation detects relations between concepts, not words. For example, in Figure 3d the first sense of *bus* and the fifth sense of *coach* fuse into a unique concept—they have the same synset in WordNet and the same gloss definition: "a vehicle carrying many passengers used for public transport"—but this relation does not hold for other senses of these two words. Further, in its first sense, bus is a kind of public transport, while in its third sense—"a car that is old and unreliable"—bus is a kind of *car*.

RESULTS

Using OntoLearn on the Harmonise European project produced a remarkable increase in ontology building productivity. After one year of ontology engineering activities, tourism experts released the most general layer of the tourism ontology, which consisted of about 300 concepts. Subsequent use of the OntoLearn tool produced a significant acceler-

ation in ontology building: In the next six months, the tourism ontology grew to include about 3,000 concepts.

A numerical evaluation led to a precision ranging from 72.9 percent to about 80 percent and a recall of 52.74 percent. The precision shift derived from the significant differences experts often demonstrate in their intuitions. This phenomenon both explains and emphasizes the need for a consensus-building groupware package like Consys.

We estimated the recall by submitting a list of 6,000 terminological candidates to the experts, requiring them to mark actual terminological entries, then comparing this list with the list our filtering method produced. We manually evaluated the semantic disambiguation algorithm on a subset of 650 extracted terms, which generated 90 domain concept trees. Ultimately, we achieved a precision of 84 percent on the overall semantic disambiguation.

We envisage several novel aspects of OntoLearn with respect to the ontology learning methods described in the research literature. Many methods have been proposed to extract domain terminology or word associations from texts and use this information to build or enrich an ontology.[7-9]

These proposals, however, invariably identify terms with domain concepts, whereas we propose an actual semantic interpretation of terms. We can use this semantic interpretation to detect other types of relations beyond the taxonomic. Our ongoing work continues to extend the amount of extracted semantic relations, exploiting the information obtained from the intersections of semantic nets.

Although WordNet does not provide an ontological standard for the semantic Web, it functions as a de facto standard for the most widely used general-purpose lexical databases. An explicit relation between a domain ontology and WordNet may favor interoperability and harmonization between different domain ontologies. In any case, researchers can easily adapt OntoLearn to work with other general-purpose ontologies. ■

Acknowledgments

The ITS-13015 and ITS-29329 projects partially supported this work.

References

1. Semantic Web Community Portal; http://www.semanticWeb.org/index.html (current 30 Sept. 2002).
2. M. Missikoff and X.F. Wang "Consys—A Group Decision-Making Support System for Collaborative Ontology Building," *Proc. Int'l Conf. Group Decision and Negotiation*, 2001, |**URL**?|.
3. M. Missikoff and F. Taglino, *Business and Enterprise Ontology Management with SymOntoX*, Springer-Verlag, New York, 2002.
4. R. Navigli and P. Velardi, "Semantic Interpretation of Terminological Strings," *Proc. 4th Int'l Conf. Terminology and Knowledge Engineering* (TKE-2002), 2002, Lecture Notes in Computer Science 2300, Springer-Verlag, New York, 2002 pp. 325-353.
5. "WordNet: A Lexical Database for the English Language," http://mind.princeton.edu/wordnet/ (current 30 Sept. 2002).
6. "SemCor—The Semantic Concordance Corpus," http://www.cogsci.princeton.edu/~wn/wn1.6.shtml (current 30 Sept. 2002).
7. A. Maedche, "Emergent Semantics for Ontologies—Support by an Explicit Lexical Layer and Ontology Learning," *IEEE Intelligent Systems*, 2002, http://wim.fzi.de/wim/publications/entries/1010141835.
8. E. Morin, "Automatic Acquisition of Semantic Relations between Terms from Technical Corpora," *Proc. 5th Int'l Congress Terminology and Knowledge Extraction* (TKE-99), TermNet, Vienna, 1999, pp. 268-278.
9. P. Vossen, "Extending, Trimming and Fusing WordNet for Technical Documents," *NAACL 2001 Workshop WordNet and Other Lexical Resources,* 2001; http://citeseer.nj.nec.com/447335.html.

Michele Missikoff *is a coordinator of the Lab for Enterprise Knowledge and Systems at IASI-CNR. His research interests include knowledge representation systems, enterprise ontologies, and the semantic Web. Missikoff received a Laurea in physics from La Sapienza, University of Rome. Contact him at missikoff@iaisi.rm.cnr.it.*

Roberto Navigli *is a grant recipient in the Department of Computer Science at the University of Rome. His research interests include natural language processing and knowledge representation. Navigli received a Laurea in computer science from La Sapienza, University of Rome. Contact him at roberto.navigli@iol.it.*

Paola Velardi *is a professor in the Department of Computer Science at the University of Rome. Her research interests include natural language processing, machine learning, and the semantic Web. Velardi received a Laurea in electrical engineering from La Sapienza, University of Rome. Contact her at velardi@dsi.uniroma1.it.*