Ontology Learning and Its Application to Automated Terminology Translation

Roberto Navigli and Paola Velardi, *Universitá di Roma La Sapienza*

Aldo Gangemi, *Institute of Cognitive Sciences and Technology*

The OntoLearn system for automated ontology learning extracts relevant domain terms from a corpus of text, relates them to appropriate concepts in a general-purpose ontology, and detects taxonomic and other semantic relations among the concepts. The authors used it to automatically translate multiword terms from English to Italian.

Although the IT community widely acknowledges the usefulness of domain ontologies, especially in relation to the Semantic Web,[1,2] we must overcome several barriers before they become practical and useful tools. Thus far, only a few specific research environments have ontologies. (The "What Is an Ontology?" sidebar on page 24 provides a definition and some background.) Many in the computational-linguistics research community use WordNet,[3] but large-scale IT applications based on it require heavy customization.

Thus, a critical issue is ontology construction—identifying, defining, and entering concept definitions. In large, complex application domains, this task can be lengthy, costly, and controversial, because people can have different points of view about the same concept. Two main approaches aid large-scale ontology construction. The first one facilitates manual ontology engineering by providing natural language processing tools, including editors, consistency checkers, mediators to support shared decisions, and ontology import tools. The second approach relies on machine learning and automated language-processing techniques to extract concepts and ontological relations from structured and unstructured data such as databases and text. Few systems exploit both approaches. The first approach predominates in most development toolsets such as Kaon, Protégé, Chimaera, and WebOnto, but some systems also implement machine learning techniques.[1]

Our OntoLearn system is an infrastructure for automated ontology learning from domain text. It is the only system, as far as we know, that uses natural language processing and machine learning techniques, and is part of a more general ontology engineering architecture.[4,5] Here, we describe the system and an experiment in which we used a machine-learned tourism ontology to automatically translate multiword terms from English to Italian. The method can apply to other domains without manual adaptation.

OntoLearn architecture

Figure 1 shows the elements of the architecture. Using the Ariosto language processor,[6] OntoLearn extracts terminology from a corpus of domain text, such as specialized Web sites and warehouses or documents exchanged among members of a virtual community. It then filters the terminology using natural language processing and statistical techniques that perform comparative analysis across different domains, or contrastive corpora. Next, we use the WordNet and SemCor[3] lexical knowledge bases to semantically interpret the terms. We then relate concepts according to taxonomic (kind-of) and other semantic relations, generating a *domain concept forest*. We use WordNet and our rule-based inductive-learning method to extract such relations.

Finally, OntoLearn integrates the domain concept forest with WordNet to create a pruned and specialized view of the domain ontology. We then use ontology editing, validation, and management tools[4,5] that are

1094-7167/03/$17.00 © 2003 IEEE
Published by the IEEE Computer Society

part of the more general architecture to enrich, correct, and update the generated ontology.

The main novel aspect of our machine learning method is *semantic interpretation*. We identify the right senses (concepts) for complex domain term components and the semantic relations between them. For example, the result of this process on the compound *transport company* selects the sense "company-enterprise" for *company* (as opposed to the "social gathering" or "crew" senses) and the sense "commercial enterprise" for *transport* (as opposed to the "exchange of molecules," "overwhelming emotion," and "transport of magnetic tape" senses), and associates a *purpose* relation (a company *for* transporting goods and people) between the two concepts.

We know of nothing similar in the ontology learning literature. Many described methods first extract domain terms using various statistical methods, then detect the taxonomic and other types of relations between terms. The literature[1, 7-9] uses the notion of domain *term* and domain *concept* interchangeably, but no semantic interpretation of terms actually takes place. For example, the concept *digital printing technology* is considered a kind of *printing technology* by virtue of simple string inclusion.[7] However, *printing* has four senses in WordNet and *technology* has two. The WordNet lexical ontology thus contains eight possible concept combinations for *printing technology*. Identifying the correct sense combinations is important in determining the taxonomic relations between a new domain concept and an existing ontology. Furthermore, semantic interpretation is relevant in view of document semantic indexing and retrieval. For example, a query for *hotel facility* could retrieve documents including concepts that are related by a kinship relation, such as *swimming pool* or *conference room*. Semantic interpretation is also useful for automatic translation, as we later demonstrate. Knowing the right senses of the complex term components in the source language—given a bilingual dictionary—greatly simplifies the problem of selecting the correct translation for each component in the target language.

OntoLearn has three main phases: terminology extraction, semantic interpretation, and creation of a specialized view of WordNet.

Phase 1: Terminology extraction

Terminology can be considered the surface appearance of relevant domain concepts. Candidate terminological expressions are usually captured with shallow techniques that range from stochastic methods to more sophisticated syntactic approaches.

Obviously, richer syntactic information positively influences the quality of the result to be input to statistical filtering. We use Ariosto to parse documents in the application domain to extract a list T_c of syntactically plausible terminological candidates. Examples include compounds (*credit card*), adjective-nouns (*public transport*), and base noun phrases (*board of directors*).

High frequency in a corpus is a property observable for terminological as well as non-terminological expressions, such as *last week*. We measure the specificity of a terminology candidate with respect to the target domain via comparative analysis across different domains. To this end, we define a specific *domain relevance* score. A quantitative definition of the *DR* can be given according to the amount of information captured in the target corpus relative to the entire collection of corpora. More precisely, given a set of n domains $\{D_1, \ldots, D_n\}$, the domain relevance of a term t in class D_k is computed as

$$DR_{t,k} = \frac{P(t|D_k)}{\sum_{j=1}^{n} P(t|D_j)},$$

where $P(t|D_k)$ is estimated by

$$E\left(P\left(t|D_k\right)\right) = \frac{f_{t,k}}{\sum_{t' \in D_k} f_{t',k}},$$

where $f_{t,k}$ is the frequency of term t in the domain D_k.

A second filter operates on the principle that a term cannot be a clue for a domain D_k unless it appears in several documents; there must be some consensus on using that term in the domain D_k. The *domain consensus* of term t in class D_k captures those terms that appear frequently across a given domain's documents. *DC* is defined as

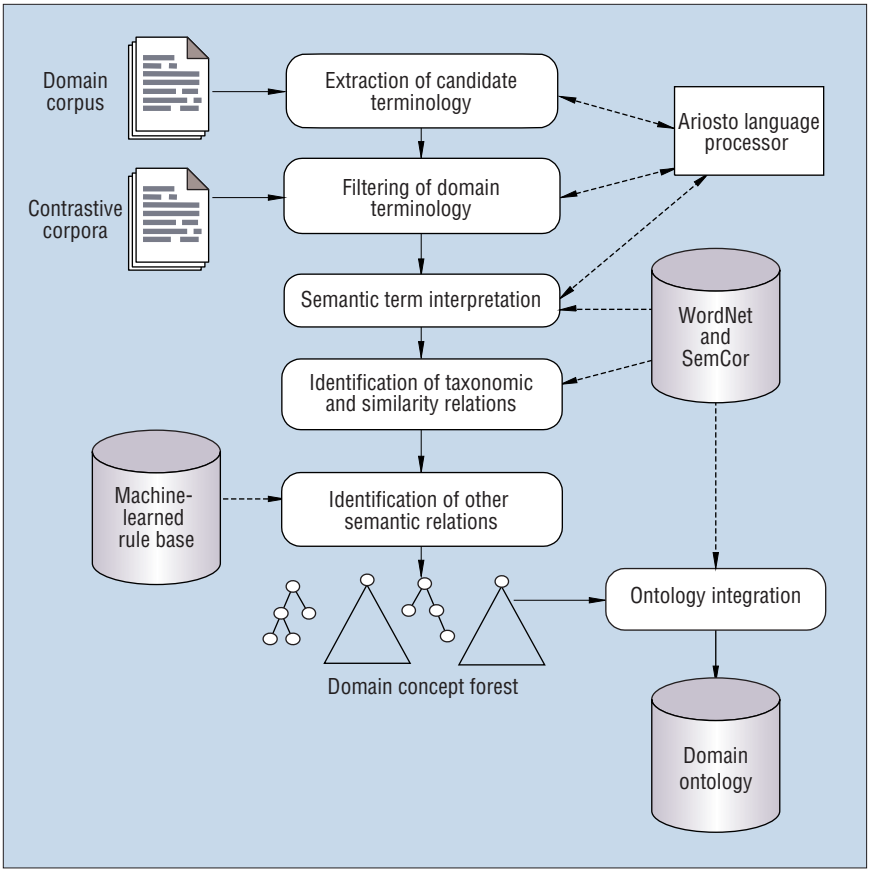

Figure 1. The OntoLearn architecture.

What Is an Ontology?

A domain ontology seeks to reduce or eliminate conceptual and terminological confusion among the members of a user community who need to share various kinds of electronic documents and information. It does so by identifying and properly defining a set of relevant concepts that characterize a given application domain, say, for travel agents or medical practitioners. An ontology specifies a *shared understanding* of a domain. It contains a set of generic concepts (such as "object," "process," "accommodation," and "single room"), together with their definitions and interrelationships. The construction of its unifying conceptual framework fosters communication and cooperation among people, better enterprise organization, and system interoperability. It also provides such system-engineering benefits as reusability, reliability, and specification.

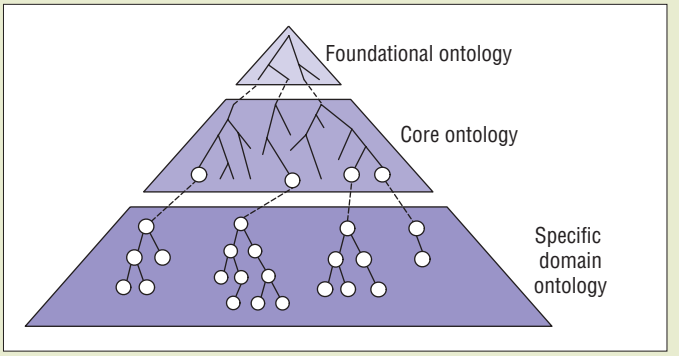

Figure A. The three levels of generality in a domain ontology.

Ontologies can have different degrees of formality, but they must include metadata such as concepts, relations, axioms, instances, or terms that lexicalize concepts. From the terminological viewpoint, an ontology can even be seen as a vocabulary containing a set of formal descriptions (made up of axioms) that approximate term meanings and enable a consistent interpretation of the terms and their relationships.

To construct an ontology, specialists from several fields must thoroughly analyze the domain by

- Examining the vocabulary that describes the entities that populate it
- Developing formal descriptions of the terms (formalized into concepts, relationships, or instances of concepts) in that vocabulary
- Characterizing the conceptual relations that hold among or within those terms

Philosophical and AI ontologists usually help define basic kinds and structures of concepts considered to be domain independent. These include metaproperties and topmost categories of entities and relationships. Identifying these few basic principles creates a *foundational ontology*[1] and supports a model's generality to ensure reusability across different domains (see Figure A).[2] Then domain modelers and knowledge engineers help identify key domain conceptualizations and describe them according to the organizational structure established by the top ontology. The result, the *core ontology*, usually includes a few hundred application-domain concepts. Although many projects eventually succeed in defining a core domain ontology, populating the third-level *specific domain ontology* with specific concepts is difficult. The projects that have overcome this barrier[3–5] have done so at the price of inconsistencies and limitations.

References

1. A. Gangemi et al., "Sweetening Ontologies with DOLCE," *Proc. 13th Int'l Conf. Knowledge Eng. and Knowledge Management* (EKAW 02), Springer-Verlag, New York, 2002; www.ladseb.pd.cnr.it/infor/Ontology/Papers/DOLCE-EKAW.pdf.
2. B. Smith and C. Welty, "Ontology: Towards a New Synthesis," *Proc. Formal Ontology in Information Systems* (FOIS 2001), ACM Press, New York, 2001, pp. .3–.9.
3. A. Miller, "WordNet: An On-line Lexical Resource," *J. Lexicography*, vol. 3, no. 4, Dec. 1990.
4. D. Lenat, "CYC: A Large Scale Investment in Knowledge Infrastructure," *Comm. ACM*, vol. 38, no. 11, Nov. 1995, pp. 32–38.
5. T. Yokoi, "The EDR Electronic Dictionary," *Comm. ACM*, vol. 38, no. 11, Nov. 1995, pp. 42–44.

$$DC_{t,k} = \sum_{d \in D_k} \left(P_t(d) \log \frac{1}{P_t(d)} \right),$$

where $P_t(d)$ is the probability that document d includes t.

A linear combination of the two filters obtains the terminology

$$DW_{t,k} = \alpha DR_{t,k} + (1-\alpha) DC_{t,k}^{norm},$$

where $DC_{t,k}^{norm}$ is a normalized entropy and $\alpha \in (0, 1)$. Only the complex terms with a *DW* value over a given threshold are retained. Because the statistical significance can be influenced by the technicality of the language domain and by the dimension of the training corpus, we determine this threshold empirically.[4]

Phase 2: Semantic interpretation

Semantic interpretation is the process of first determining the right concept (sense) for each component of a complex term (this is known as *semantic disambiguation*) and then identifying the semantic relations holding among the concepts to build a complex concept. OntoLearn starts by hierarchically arranging the set of validated terms into subtrees according to simple string inclusion. Figure 2 is an example of a lexicalized tree **T**. In the absence of semantic interpretation, however, we cannot fully capture conceptual relationships between senses (for example, the kind-of relation between *bus service* and *public transport service* in Figure 2).

The WordNet lexical knowledge base associates a set of senses with each word. Sense names are called *synsets* (synonym sets). Each word in a synset is marked with its sense number. For example, for sense #3 of *transport*, it would provide "transportation#4," "shipping#1," and "transport#3." WordNet also provides sense definitions in natural language, called *glosses*, and several types of lexicosemantic relations, like taxonomic

(kind-of), similarity, and part-of relations.

WordNet includes over 120,000 words (and over 170,000 synsets) but few domain terms. For example, *transport* and *company* are individually included, but not *transport company*. The SemCor knowledge base is a balanced corpus of semantically annotated sentences in which every word is annotated with a sense tag selected from the WordNet sense inventory for that word. We use SemCor to automatically extract examples of concept co-occurrences.

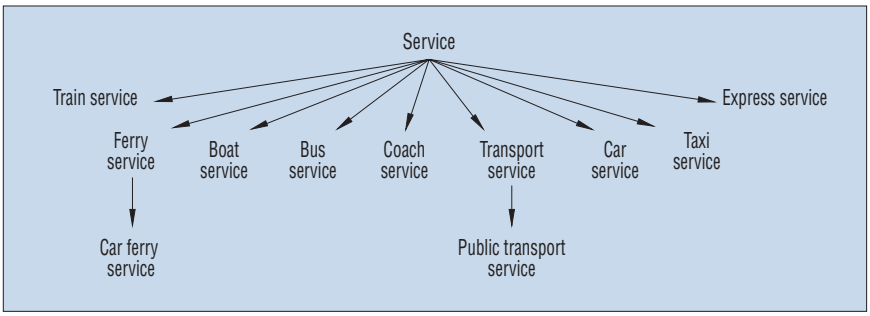

Figure 2. A lexicalized tree in a tourism domain.

Semantic disambiguation

Let $t = w_n \cdot \ldots \cdot w_2 \cdot w_1$ be a valid multiword term belonging to a lexicalized tree **T**. The process of semantic disambiguation associates the appropriate WordNet synset S_k^i to each word w_k in t. So, the *sense* of t is defined as

$$S(t) = \left\{ S_k^i \middle| \; S_k^i \in Synset(w_k), w_k \in t \right\},$$

where $Synset(w_k)$ is the set of senses provided by WordNet for word w_k. For instance,

S(*transport company*) =
{{ *transportation#4*, *shipping#1*, *transport#3* }, { *company#1* }}

corresponds to sense #1 of *company* ("an institution created to conduct business") and sense #3 of *transport* ("the commercial enterprise of transporting goods and material"). Let $t = w_n \cdot \ldots \cdot w_2 \cdot w_1$ be a multiword term, say, *public transport service*. Let w_1 be the head of the string (in English compounds, the leftmost).

Semantic disambiguation takes place in three steps. The first is creating semantic nets: For any $w_k \in t$ and any synset of w_k (where S_k^i is the ith sense of w_k in WordNet), create a *semantic net* (*SN*). OntoLearn automatically builds semantic nets by using the following lexicosemantic relations:

- Hyperonymy (car *is-a-kind-of* vehicle, denoted with $\rightarrow^{@}$)
- Hyponymy (its inverse, $\rightarrow^{\sim}$)
- Meronymy (room *has-a* wall, $\rightarrow^{\#}$)
- Holonymy (its inverse, $\rightarrow^{\%}$)
- Pertainymy (dental *pertains-to* tooth $\rightarrow^{\backslash}$)
- Attribute (dry *value-of* wetness, $\rightarrow^{=}$)
- Similarity (beautiful *similar-to* pretty, $\rightarrow^{\&}$)
- Gloss (Concept appears in the definition of another concept, $\rightarrow^{\text{gloss}}$)
- Topic (Concept often co-occurs with another concept, $\rightarrow^{\text{topic}}$)

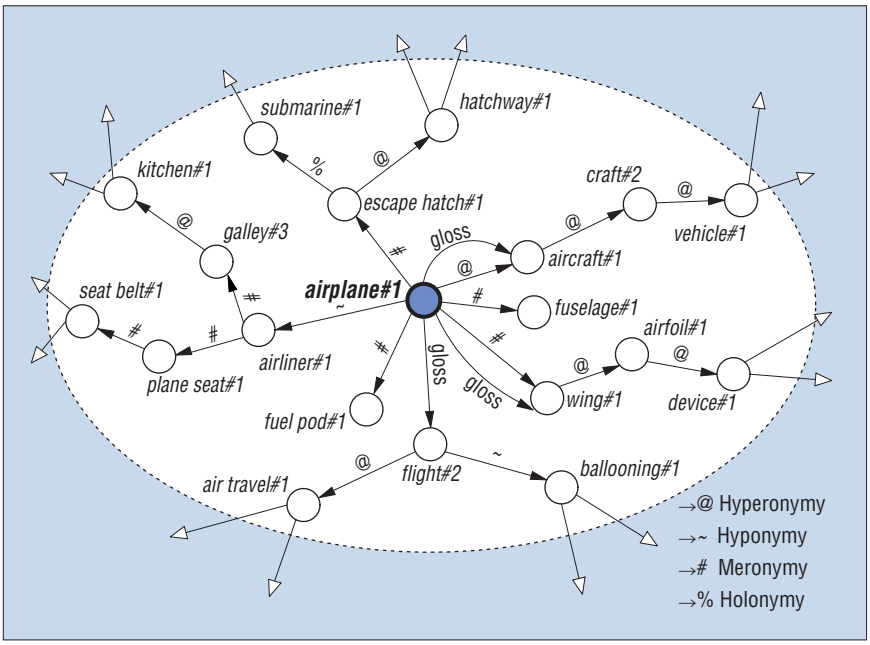

Figure 3. The semantic net for sense #1 of *airplane*.

Ariosto obtains the *gloss* and *topic* relations by parsing, respectively, the WordNet concept definitions and the SemCor sentences including that sense. OntoLearn extracts every other relation directly from WordNet. As shown by the arrows in Figure 3, to reduce the dimension of a semantic net, we consider only concepts at a distance not greater than three relations from S_k^i (the semantic net center). Figure 3 is an example of a semantic net generated for sense #1 of *airplane*.

The next step intersects the semantic nets. The algorithm evaluates pairwise, from left to right, alternative intersections. Given two adjacent words w_{i+1} and w_i, let

$$I = SN\left(S_{i+1}^j\right) \cap SN\left(S_i^k\right)$$

be one possible intersection. For each alternative intersection I, we compute a score that depends on the number and type of *semantic patterns* connecting the semantic net centers. Semantic patterns must be instances of 13 predefined metapatterns. In the following two metapattern examples, S_1 and S_2 represent the central concepts of each semantic net.

- *Topic*, if $S_1 \overset{\text{topic}}{\rightarrow} S_2$ (as in the term *archeological site*, where the two words co-occur and are tagged with sense #1 in a SemCor sentence)
- *Gloss + hyperonymy path*, if

$$\exists G, M \in Synset_{WN} : S_1 \overset{\text{gloss}}{\rightarrow} G \overset{@,\#\leq 3}{\rightarrow} M \overset{\leq 3\ \sim,\%}{\leftarrow} S_2 \vee S_1 \overset{\text{gloss}}{\rightarrow} G \overset{\sim,\%\leq 3}{\rightarrow} M \overset{\leq 3\ @,\#}{\leftarrow} S_2.$$

For instance, in *railway company*, the gloss

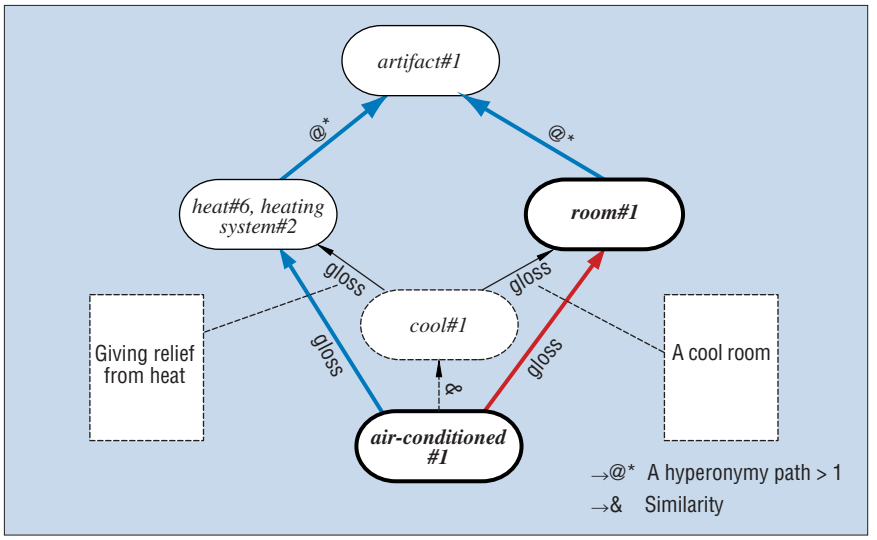

Figure 4. Intersection between the semantic nets of *air-conditioned#1* and *room#1*. The central nodes appear in bold. The blue and red arrows represent two instances of metapatterns found at an intersection.

of *railway#1* contains the word *organization* and an hyperonymy path exists from *company* to *organization*: *company#1* $\rightarrow^{@}$ *institution#1* $\rightarrow^{@}$ *organization#1*.

Figure 4 shows an example intersection between *air-conditioned#1* and *room#1* in which the following semantic paths are found[6]:

$$air\text{–}conditioned\#1 \xrightarrow{\text{gloss}} heat\#6 \xrightarrow{@3} artifact\#1 \xleftarrow{3@} room\#1$$

$$air\text{–}conditioned\#1 \xrightarrow{\text{gloss}} room\#1.$$

The last step is finding taxonomic relations. Initially, all the complex terms in a tree **T** are independently disambiguated. Then, the algorithm detects taxonomic relations between *concepts*, as in *ferry service* $\rightarrow^{@}$ *boat service*. OntoLearn infers this information from WordNet on the basis of the synsets now associated with each component of the complex term.

In this phase, since all elements in **T** are jointly considered, interpretation errors from the previous disambiguation step are corrected. In addition, certain concepts are fused in a unique "semantic domain" on the basis of pertainymy, adjectival similarity, and synonymy relations (for instance, respectively, *manor house* and *manorial house*, *expert guide* and *skilled guide*, *bus service* and *coach service*). Note that we detect semantic relations between concepts, not words. For example, *bus#1* and *coach#5* are synonyms, but this relation does not hold for other senses of these two words.

At the end of this step, the lexicalized tree **T** is reorganized into a domain concept tree **D** in which terms have been replaced by concepts and the appropriate taxonomic relations have been detected. Figure 5 shows the domain concept tree **D** obtained from the lexicalized tree **T** of Figure 2. Numbers for concepts are shown only when more than one semantic interpretation holds for a term, as for *coach service* and *bus service* (for example, sense #3 of *bus* refers to *old cars*).

At this point, a WordNet synset is associated with each component of a complex term, and taxonomic relations connect the head components of each complex concept. To complete the interpretation process, we must determine the semantic relations that hold between the components of a complex concept. These relations provide richer semantic information for many applications such as information extraction, query answering, and machine translation.

Extracting semantic relations

To extract the relations in a complex concept, we must

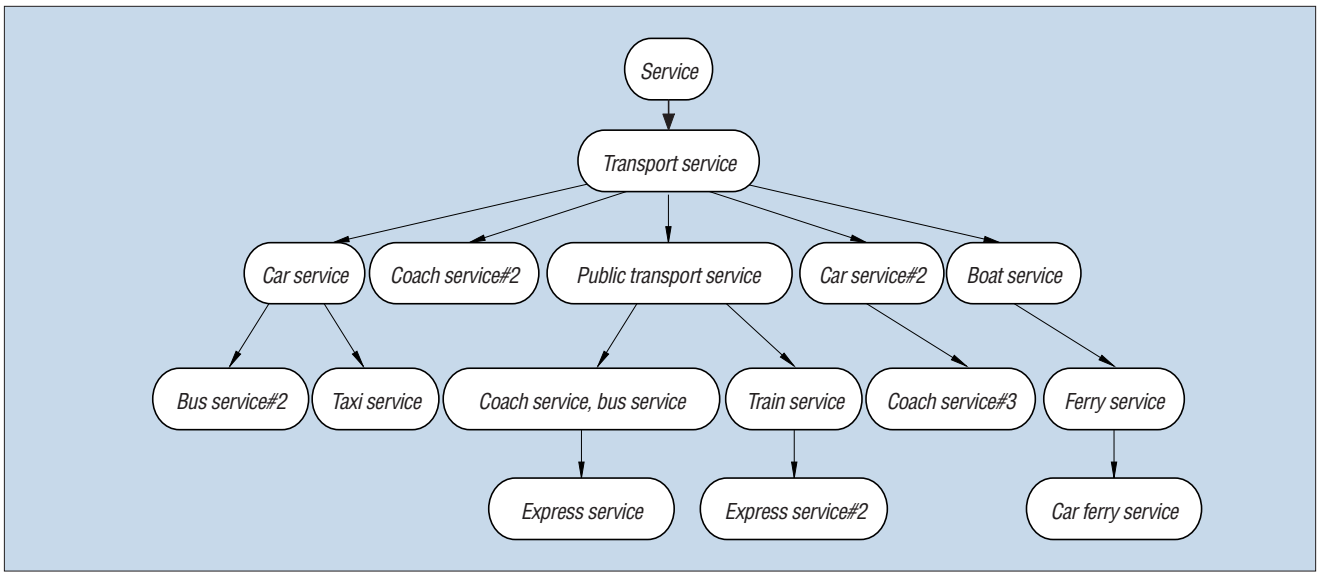

Figure 5. A domain concept tree.

- Select an inventory of domain-appropriate semantic relations.
- Learn a formal model to select the relations that hold between pairs of concepts, given ontological information on these concepts.
- Apply the model to semantically relate the components.

First, we selected an inventory of semantic relations types. To this end, we consulted John Sowa's[10] formalization on conceptual relations, as well as other studies conducted within the CoreLex[11] and EuroWordNet[12] systems. Because the literature did not provide systematic definitions for semantic relations, we selected only the more intuitive and widely used. (The WonderWeb project has begun work on providing a well-funded set of rigorously defined semantic relations.)

To begin, we selected a kernel inventory, including the following 10 relations that we found pertinent to the tourism domain (examples of conceptual relations are expressed in a style similar to Sowa's conceptual graphs notation).

- Place (for example, room←PLACE←service, that reads: "the service *has* place in a room" or—when the arrows point to the right—"the room *is* the place of service")
- Time (afternoon←TIME←tea)
- Matter (ceramics←MATTER←tile)
- Theme (art←THEME←gallery)
- Manner (bus←MANNER←service)
- Beneficiary (customer←BENEF←service)
- Purpose (booking←PURPOSE←service)
- Object (wine←OBJ←production)
- Attribute (historical←ATTR←town)
- Characteristics (first-class←CHRC←hotel)

You can easily adapt this set or extend it to other domains.

To associate the appropriate relations that hold among the components of a domain concept, we used *inductive machine learning*.[13] In inductive learning, you first manually tag a subset of domain concepts (the *learning set*) with the appropriate semantic relations and then let an inductive learner build a tagging model. We selected the C4.5 program[14] because it produces a decision tree or a set of rules as output. Unlike algebraic or probabilistic learners, decision-tree or rule learners produce output that humans can easily understand and modify.

An inductive-learning system represents the instances of a domain through a feature vector. In our case, instances are concept-relation-concept triples (for example, wine ←OBJ←production), in which the type of relation is given only in the learning set.

If in modifier [knowledge_domain#1, knowledge-base#] =1 then relation THEME (63%)
Ex: arts festival, science center

If in modifier [building-material#] =1 then relation MATTER (50%)
Ex.: stave church, cobblestone street

If in modifier [conveyance#3, transport#] =1 and in head [act#1, human_act#] =1
then relation MANNER (92.2%)
Ex.: bus service, coach tour

Figure 6. Three classification rules with confidence factors.

After WordNet attaches the domain concept trees under the appropriate nodes, all branches not containing a domain node can be removed from the WordNet hierarchy.

We explored several possibilities for features selection. We obtained the best result when we represented each concept component with the complete sequence of its hyperonyms (up to the topmost). We began with a complex term (*board of directors* or *transport company*) and, after interpretation, we had an ordered list of concepts in which the rightmost is the concept head (*board* or *company*), while the modifiers are placed on the left. Therefore, we have, say, "director#2, board#1" for *board of directors* and "transport#3, company#1" for *transport company*. For each complex concept represented in this way, we generated a feature vector in which we follow the same order as for concept components and represented each concept component with a list of all its hyperonyms. The features are then hyperonym lists. Therefore, we have a list of lists:

$$feature\_vector\{\{list\_of\_hyperonims\}_{mod}^{1-n}, \{\text{list\_of\_}hyperonims\}_{head}\}$$

For example, the feature vector for *tourism operator*, in which *tourism* is the modifier and *operator* is the head, becomes the sequence of hyperonyms for *tourism #1*:

{*tourism#1,commercial_enterprise#2, commerce#1,transaction#1, group_action#1,act,human-action#1*}.

These were followed by the sequence of hyperonyms for *operator#2*:

{*operator#2,capitalist#2,causal_agent#1, entity#1,life_form#1,person-individual#1*}.

Features are converted in a binary representation to obtain vectors of equal length.

We ran several experiments, using two-fold cross-validation and a tagged set of 405 complex concepts—a varying fragment for learning and the other for testing. Overall, the best experiment provided a 6 percent error rate over 405 examples and produced around 20 classification rules. Figure 6 shows three of the extracted rules, along with their confidence factors and a few examples.

Phase 3: Creating a specialized view of WordNet

At the end of the last phase, we obtained a domain concept forest showing the taxonomic and other semantic relations among complex domain concepts. We now integrate this DCF with a core domain ontology if available or automatically create one from WordNet, extending it with the DCF and pruning concepts not relative to the domain. After WordNet attaches the domain concept trees under the appropriate nodes, all branches not containing a domain node can be removed from the WordNet hierarchy. An

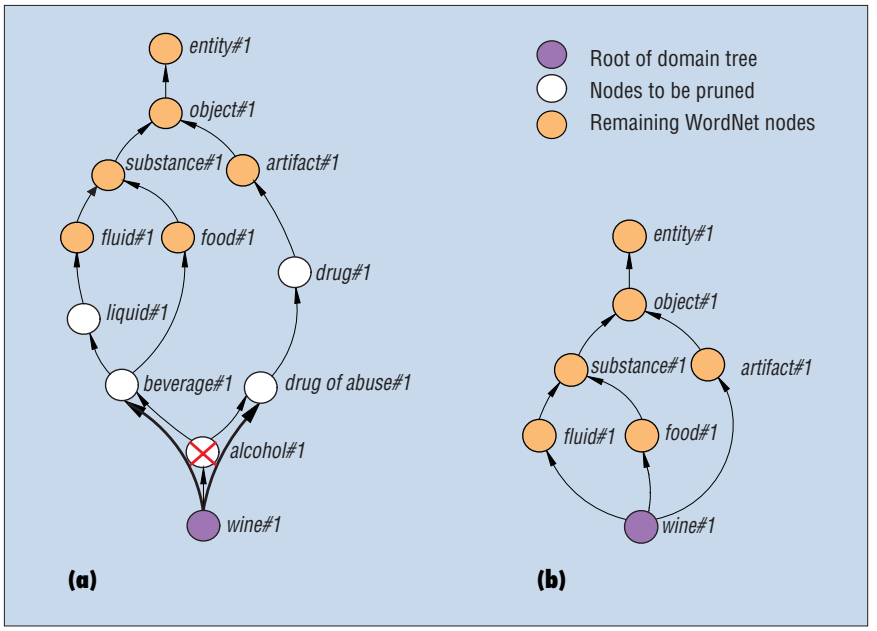

Figure 7. An intermediate step (a) and the final pruning step (b) in the domain concept tree for *wine#1*. Circles indicate WordNet concepts.

intermediate node in WordNet is pruned whenever the following conditions hold together: The node

- Has no "brother" nodes
- Has only one direct hyponym
- Is not the root of a domain concept tree
- Is not at a distance ≤ 2 from a WordNet *topmost concept* (this is to preserve a minimal top ontology)

Figure 7 shows an example of pruning the nodes located over the domain concept tree with root *wine#1*.

Evaluation of OntoLearn

Although a complete field evaluation is still progressing within the European Community project Harmonise, some basic facts indicate our method's validity. After one year, the tourism experts released the most general layer of the tourism ontology, comprising approximately 300 concepts. After OntoLearn was introduced, the ontology grew to approximately 3,000 concepts within six months. Manual verification of automatically acquired domain concepts actually took only a few days.

We have also evaluated OntoLearn independently from the ontology engineering process. Starting from a one-million-word corpus of travel descriptions found on dedicated Web sites, the system automatically extracted a terminology of 3,840 terms. Domain experts manually evaluated these terms, obtaining a precision ranging from 72.9 percent to approximately 80 percent and a recall of 52.74 percent. The precision shift is due to the fact that experts have different intuitions. The experts estimated the recall after automatically extracting 14,000 terminological candidates from a fragment of the corpus and then manually selecting the "true" domain terms. Our automatic terminology filter identified 52.74 percent of these terms.

Because the participants in the EC project were not skilled in evaluating the semantic interpretation algorithm, we personally evaluated it. We used a testbed of about 650 extracted complex terms that had been manually assigned to the appropriate WordNet concepts. These terms contributed to 90 syntactic trees created out of a variable number of concepts (from two to about 30) and variable depth (the maximum is four). The number and depth of generated trees is a property of the sublanguage. In an economic domain, which is more technical than tourism is, we found a much larger number of hierarchically related domain concepts.

An extensive evaluation of the whole semantic disambiguation process highlighted that some heuristics contribute more than others. In particular, rules that use glosses bring precise semantic information for term disambiguation. In fact, while the inclusion of those heuristics gives a precision of 84.56 percent (hence this number represents the OntoLearn average precision for semantic disambiguation), their exclusion decreases precision to 79 percent. The precision grows to about 89 percent for highly structured subtrees, as those in Figure 5. We also computed a baseline, selecting for each concept the first WordNet sense (ordered by frequency of use). The baseline obtained a precision of 75.06 percent.

Although the results are encouraging, an objective evaluation of an ontology as a standalone artifact is not fully satisfactory. The only possible success indicator is the subjective acceptance–rejection rate provided by ontology engineers after inspecting automatically extracted information. An ontology can also be better evaluated when many users access it regularly and use this shared knowledge to better communicate and access prominent information and documents. An ontology can also be objectively evaluated in the context of another software application in terms of performance improvement. To this end, we designed an ontology-based multiword translation experiment.

Applying OntoLearn to multiword term translation

Automatic translation of complex nouns is highly relevant to a variety of applications, especially where technical terms occur frequently. The best-performing approaches retrieve the correct translation by using bilingual parallel (that is, aligned) corpora. However, parallel corpora are rare, especially when dealing with such sublanguages as medicine, tourism, and commerce. Other methods use monolingual corpora in the source and target language, exploiting context similarity to extract possible term correspondences. Statistical, algebraic, and machine-learning methods select the appropriate translation. Dictionary-based approaches use the translation of constituent words to build the translation of the full complex term. To deal with the combined ambiguity of two languages (each constituent has many senses and each one has several translations), using evidence from corpora and statistical methods can reduce the set of possible translations. (See the "Related Work" sidebar for further information.)

To bypass the difficulty of obtaining correct translations, cross-language information retrieval establishes translingual associations between the full query in the source language

Related Work

It is difficult to compare our work to other projects in the literature, because few papers on noun-phrase translation contain evaluation results. In one paper on complex noun translation from English to Japanese,[1] the results are quite poor: approximately 34 percent precision and 60 percent recall for a limited test of 10 compound nouns. (In our algorithm, the low recall is due to the poor dictionary. In principle, for a full mapping of synsets from source to target language, the recall is 100 percent.) Another project[2] recently obtained better results by using a Web search for generating translation candidates, much as we do, and then applying a filter based on a Bayesian classifier or on statistical filters. The performance obtained with the best methods combination is 68.6 percent precision and 86.2-percent coverage. Another paper[3] reports using a Web search exclusively, without additional filters, to achieve 86 to 87 percent precision on a very-large-scale experiment. In all the papers, however, the translated terms are base noun phrases extracted from a dictionary or an encyclopedia rather than from an automatically extracted terminology.

These results confirm the extreme effectiveness of Web search heuristics. Nonetheless, using an ontology can greatly reduce the search space with respect to word-to-word translation.

Further related work concerns a large translingual information retrieval (TIR) experiment.[4] Here, the comparative recall-precision graph for various methods reaches the maximum value of 35 percent precision at 60 percent recall, but the maximum precision is 74 percent at almost zero recall. More recent results from the TIR track of TREC 2001 (the annual Text Retrieval Conference) show a precision of 40 to 45 percent for the best systems.

The literature concerning multiword term translation contains a limited number of experiments,[1] and the results seem modest.

Although the number of contributions in the area of ontology learning and construction has considerably increased, experimental data on the utility of ontologies is not available except for Ontology Server,[5] which presents an analysis of user distribution and requests. A better performance indicator would have been the number of regular users that access Ontology Server, but the authors mention that regular users are only a small percentage. Although recent efforts are being made in terms of ontology evaluation tools and methods such as the ONE-T system at Stanford University, results are methodological rather than experimental.

References

1. I. Sharhzad et al., "Identifying Translations of Compound Nouns Using Non-aligned Corpora," *Proc. Multilingual Information Processing and Asian Language Processing* (MAL-99 Workshop); http://kibs.kaist.ac.kr/nlprs99/w_mal99.htm.
2. Y. Cao and H. Li, "Base Noun Phrase Translation Using Web Data and the EM Algorithm," *Proc. 19th Int'l Conf. Computational Linguistics* (COLING 2002), Morgan Kaufmann, San Francisco, pp. 127–133; http://research.microsoft.com/users/hangli/HP_files/Cao-Li-COLING02.pdf.
3. G. Grefenstette, "The World Wide Web as a Resource for Example-Based Machine Translation Task ASLIB," *Translating and the Computer 21*, www.xrce.xerox.com/competencies/content-analysis/publications/Documents/P49030/content/gg_aslib.pdf.
4. Y.Yang et al, "Translingual Information Retrieval: Learning from Bilingual Corpora," *Artificial Intelligence,* vol. 103, nos. 1-2, 30 Aug. 1998, pp. 323–345; http://citeseer.nj.nec.com/cache/papers/cs/508/http:zSzzSzwww.cs.cmu.eduzSz~yimingzSzpapers.yyzSzaij98.pdf/yang97translingual.pdf.
5. A. Farquhar et al., "Collaborative Ontology Construction for Information Integration," http://www.ksl.stanford.edu/KSL_Abstracts/KSL-95-63.html.

and the answering text in the target language.[15] However, because terminology conveys most of a text's meaning, complex terms need high-quality translations.

The experiment

OntoLearn makes the dictionary-based approach more feasible because it relates a complex term to an unambiguous complex concept, thus eliminating a major source of ambiguity. To verify this claim, we performed an experiment of multiword term translation in the tourism domain, using EuroWordNet[12] to translate English to Italian. We used OntoLearn to extract domain terminology, associate each constituent of a term to its EuroWordNet synset in the source language, and derive the appropriate semantic relation.

A complete example of the information derived by OntoLearn for *room service* is

synset_WN(*room*) = {*room%1*}
gloss_WN({*room%1*}) = an area within a building enclosed by walls and floor and ceiling
synset_WN(*service*) = {*service%1*}
gloss_WN({*service%1*}) = work done by one person or group that benefits another
{*room%1*}(PLACE){*service%1*}

where synset_WN(*w*) and gloss_WN(*w*) are, respectively, the WordNet synset and WordNet definition for a word *w*.

For each synset, there are three possible cases:

- The translation of the same synset is available in the target language.
- One or more hyponyms, hypernyms, or near synonyms are available in the target language.
- No correspondences exist (the most frequent case in Italian EuroWordNet).

The algorithm selects a corresponding synset only in the first case.

A list of synonyms often serves as a basis for a synset translation, but usually only one word is appropriate to generate it. For example, the translation of sense #1 of *center* (*centre*, *center*, *middle*, *heart*, *eye*) is (*centro*, *cuore*), corresponding to words #1 and #4 of the English synset. However, only *centro* is an appropriate translation in the context of the term *health center*. To select a translation, we queried the Web with alternative translations and counted the hits in Google for each alternative. In contrast to other dictionary-based

Table 1. Results* of the automated translation experiment for complex domain concepts that have corresponding synsets.

Quality of translation	Good	Acceptable	Poor	Total
Manually corrected input	74% (84)	14% (16)	12% (13)	113
OntoLearn-generated input	70% (71)	14% (14)	16% (16)	101

*In percentages and absolute numbers

methods, we were dealing with disambiguated terms, which reduces the number of alternate translations. The hit-count method worked well. In some cases, more than one translation was acceptable.

The last step was to replace the source language construction for a term (usually a compound or adjective noun in English) with the appropriate construction in the target language. We used the information provided by the OntoLearn semantic relations to do this. In Italian, a compound corresponds to a postmodified prepositional phrase, and the type of preposition depends on the subsumed semantic relation. We used mapping between conceptual relations and prepositions to construct the appropriate complex nominal. For example,

{*room#1*} (PLACE) {*service#1*}

becomes

SERVIZIO in CAMERA

For some relations, more than one preposition might be appropriate. We selected the most frequently used, again using supporting evidence from the Web.

Using the pertonimy relation in WordNet 1.6 proved to be good heuristics. When a noun was related to its adjectival realization, we created a postmodified adjectival phrase, usually the most appropriate translation in Italian. For example, the English term *folklore festival* is better translated as *sagra folcloristica* (folkloristic festival) than as *festival sul folklore* (festival on folklore), although the prepositional realization is acceptable.

Adjectival constructs are preferable only for a subset of semantic relations. Even in this case, it would be better to produce more alternatives and use corpus evidence to restrict the set depending upon usage, but we could not apply this heuristic extensively due to the limited number of adjectives encoded in the Italian EuroWordNet.

Results

We conducted the experiment on the 405 complex terms extracted from our manually validated tourism corpus. The validation concerns both semantic disambiguation and extraction of semantic relations. The cumulative error rate of OntoLearn on these data was 16.8 percent, 2 percent worse than the performance we computed over a set of 650 complex disambiguation terms. The semantic relation was inappropriate in only three cases. All other errors were due to semantic disambiguation. However, as the translation experiment results later proved, the sense selected by OntoLearn was often very close to the manual one.

We submitted both the validated and the error-prone data to the translation procedure so that we could distinguish between errors that occurred during semantic interpretation and those that occurred during the generation of translations. Unfortunately, the primary obstacle was the poor encoding of Italian EuroWordNet with respect to WordNet 1.5 and 1.6. In the English version, 66,042 synsets are coded for nouns and 17,944 for verbs. In Italian, these numbers drop to 31,806 and 3,873.

Our 405 complex terms were made out of 335 different words. Several words occurred frequently: 44 complex terms with the head *center*, 45 with the head *service*, and 16 with the head *hotel*. OntoLearn associated these words to 317 different synsets, since some words were synonyms. Of these synsets, only 164 had a correspondent in Italian WordNet. For 53, there was a near synonym, hyponym, or hyperonym; for 41, the synset was not encoded, but other unrelated synsets of the source language word were available; and in 59 cases, none of the English word synsets had a correspondent.

Consequently, we had only 101 complex concepts (113 when manually corrected) for which our translation procedure determined the correspondent synsets in the target language for all components. In some cases, the Italian translation of a synset was rather odd or archaic. For example, the translation of *trade* in the sense #1 "the commercial exchange of goods and services," as in *trade center*, is *tratta,* which connotes more of the "patronage" sense. In other cases, the preferred adjectival realization was not available in Italian WordNet. For example, *tour operator* is better translated to *operatore turistico* (adjective) than to *operatore del turismo* (noun). In other cases, the English term is more popular than its reported Italian translation, as in *Internet* and its translation *cyberspazio*. In all these cases, even though the translations are poor or barely acceptable, we decided not to prune them from our evaluation because this would be subjective. Rather, we plan to run our next experiment by using other target languages such as Spanish, for which EuroWordNet provides a richer bilingual dictionary. Table 1 summarizes our results.

Note in Table 1 that the difference in precision between the manually corrected and automatically generated input in the first column is 4 percent, which is much less than the OntoLearn error rate. This demonstrates that often an OntoLearn-generated synset is only subtly different in meaning from the manually chosen synset. The translation algorithm selects the right term anyhow. In some cases, two English synsets are coded with the same Italian synset.

To better evaluate our results, we computed a baseline by selecting the most probable synset in WordNet for each complex term component—for example, sense #1. We then evaluated the quality of translations. In general, the first-sense heuristics produced the right sense combinations for 68 of 113 cases (60.1 percent). When the synsets are not dramatically different (this happens rarely, as when erroneously selecting the "transmission" sense for *channel* in *channel island*), the Web search heuristics "adjust" the translation. Therefore, the overall translation quality for the baseline is much higher than expected, given the initial disambiguation error rate. For a fair comparison, we selected the 60 complex terms for which both OntoLearn and the baseline method could produce a translation—that is, all the synsets in a complex concept that had a correspondent synset in Italian. In these cases, the OntoLearn-based translation produced four errors (6.6 percent), whereas the first-sense heuristics had seven (11.6 percent).

Although not striking, these results are sufficient, and we believe they could improve significantly with a richer version of Italian EuroWordNet or with yet another EuroWordNet language.

We consider our results to be good, given that the experiment is preliminary and there is room for improvement. The most serious problem is that our approach works well for only those complex terms that correspond with their translation on a part-by-part basis. We found few exceptions in our experiment, but the problem certainly arises for languages such as German or Japanese. In these cases, additional computational devices are needed to split or merge the various parts.

This translation experiment also sought to demonstrate the utility of an automatically learned ontology *within* an application. This is not a trivial result. To continue our analysis of ontology-sensitive applications, our ongoing research focuses on ontology-based information retrieval. ■

Acknowledgments

The OntoLearn project was funded in part by the EC projects Fetish ITS-13015 and Harmonise IST-2000-29329 (http://dbs.cordis.lu). The project was also funded in part by other ontology-related projects such as the OntoWeb thematic network (http://ontoweb.aifb.uni-karlsruhe.de) and WonderWeb IST-2001-33052 (http://wonderweb.semanticweb.org).

References

1. A. Maedche and S. Staab, "Ontology Learning for the Semantic Web," *IEEE Intelligent Systems*, vol. 16, no. 2, Mar./Apr. 2001, pp. 72–79.
2. D. Fensel et al., "OIL: An Ontology Infrastructure for the Semantic Web," *IEEE Intelligent Systems*, vol. 16, no. 2, Mar./Apr. 2001, pp. 38–45.
3. A. Miller, "WordNet: An On-line Lexical Resource," *J. Lexicography*, vol. 3, no. 4, Dec. 1990.
4. R. Navigli and P. Velardi, "Semantic Interpretation of Terminological Strings," *Proc. 6th Int'l Conf. Terminology and Knowledge Eng.* (TKE 2002), INIST-CNRS, Vandoeuvre-lès-Nancy, France, 2002, pp. 95–100.
5. M. Missikoff, R. Navigli, and P. Velardi, "The Usable Ontology: An Environment for Building and Assessing a Domain Ontology," *Proc. 1st Int'l Semantic Web Conf.* (ISWC 2002), Lecture Notes in Computer Science, no. 2342, Springer-Verlag, Berlin, 2002, pp. 39–53.
6. R. Basili, M.T. Pazienza, and P. Velardi, "An Empirical Symbolic Approach to Natural Language Processing," *Artificial Intelligence*, vol. 85, nos. 1–2, Aug. 1996, pp. 59–99.
7. P. Vossen, "Extending, Trimming and Fusing WordNet for Technical Documents," *NAACL 2001 Workshop WordNet and Other Lexical Resources*, Assoc. for Computational Linguistics, East Stroudsburg, Pa., 2001, pp. 125–131.
8. E. Morin, "Automatic Acquisition of Semantic Relations between Terms from Technical Corpora," *Proc. 5th Int'l Congress Terminology and Knowledge Eng.* (TKE 99), Int'l Network for Terminology, Vienna, 1999.
9. E. Agirre et al., "Enriching Very Large Ontologies Using the WWW," *ECAI 1st Ontology Learning Workshop*, 2000, http://ol2000.aifb.uni-karlsruhe.de.
10. J. Sowa, *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Boston, 1984.
11. P. Buitelaar, "CoreLex: An Ontology of Systematic Polysemous Classes," *Proc. Int'l Conf. Formal Ontology in Information Systems* (FOIS 98), www.dfki.de/~paulb/pub.html.
12. "EuroWordNet: Building a Multilingual Database with Wordnets for Several European Languages," www.illc.uva.nl/EuroWordNet.
13. T. Mitchell, *Machine Learning,* McGraw-Hill, New York, 1997.
14. J.R. Quinlan, *C4.5: Programs for Machine Learning,* Morgan Kaufmann, San Francisco, 1993.
15. Y. Cao and H. Li, "Base Noun Phrase Translation Using Web Data and the EM Algorithm, *Proc. 19th Int'l Conf. Computational Linguistics (*COLING 2002), Morgan Kaufmann, San Francisco, 2002, pp. 127–133; http://research.microsoft.com/users/hangli/HP_files/Cao-Li-COLING02.pdf.

The Authors

Roberto Navigli is a grant recipient in the Department of Computer Science at the Universitá di Roma. His research interests include natural language processing and knowledge representation. He received his Laurea degree in computer science from Universitá di Roma La Sapienza. Contact him at Univ. di Roma La Sapienza, Dipart. di Informatica, Via Salaria 113, 00196 Roma, Italy; roberto.navigli@iol.it.

Paola Velardi is a full professor in the Department of Computer Science at the Universitá di Roma. Her research interests include natural language processing, machine learning, and the Semantic Web. She received her Laurea degree in electrical engineering from Universitá di Roma La Sapienza. Contact her at velardi@dsi.uniroma1.it.

Aldo Gangemi is a research scientist at the Institute for Cognitive Sciences and Technology. His research interests include conceptual modeling, ontological engineering, and the Semantic Web. He received his Laurea degree in philosophy from Universitá di Roma La Sapienza. Contact him at gangemi@ip.rm.cnr.it.