In Proceedings of the 16th International FLAIRS Conference (FLAIRS-2003), Special Track on Machine Learning, May 2003, Florida. AAAI Press, Menlo Park, CA 2003.

Improving the Representation Space through Exception-Based Learning

Cristina Boicu[1], Gheorghe Tecuci[1,2], Mihai Boicu[1], Dorin Marcu[1]

[1] Learning Agents Laboratory, Computer Science Department, George Mason University, 4400 University Dr, Fairfax, VA 22030
{ccascava, tecuci, mboicu, dmarcu}@gmu.edu, http://lalab.gmu.edu
[2] Center for Strategic Leadership, US Army War College, 650 Wright Ave, Carlisle Barracks, PA 17013

Abstract

This paper addresses the problem of improving the representation space in a rule-based intelligent system, through exception-based learning. Such a system generally learns rules containing exceptions because its representation language is incomplete. However, these exceptions suggest what may be missing from the system's ontology, which is the basis of the representation language. We describe an interactive exception-based learning method for eliciting new elements in the system's ontology in order to eliminate the exceptions of the rules. This method is implemented in the Disciple learning agent shell and has been evaluated in an agent training experiment at the US Army War College.

1 Introduction

One of the main challenges in developing knowledge-based agents for solving real-world problems is how to acquire and represent expert's problem solving knowledge. Subject matter experts usually express their knowledge informally, in natural language, using visual representations and common sense reasoning. By contrast, the knowledge of an agent must be represented in a formal, precise and fairly complete way. The consequence of this mismatch is that an expert's knowledge is only partially expressed in the agent's representation language. Therefore, an agent's representation of an application domain needs to be continuously extended in order to better characterize the subtle distinctions that real experts make in their domain.

In the case of a rule-based learning agent, the incompleteness of the representation language results in the learning of rules with exceptions (Tecuci 1998; Wrobel 1994; Ling 1991). However, the exceptions may indicate missing or partially represented knowledge. We have developed a method that performs an analysis of the exceptions and suggests extensions to the representation language of a learning agent, improving the rules by eliminating their exceptions.

We have implemented and experimentally evaluated this exception-based learning method in the context of the Disciple approach (Tecuci et al. 2002). However, the method may be used in any learning agent with a similar knowledge representation. Disciple is an evolving theory, methodology and family of agent shells for rapid development of end-to-end knowledge bases and agents, by subject matter experts, with limited assistance from knowledge engineers. The Disciple approach directly addresses the knowledge acquisition bottleneck (Buchanan and Wilkins 1993), which is considered a major barrier in the development of knowledge-based systems. This approach relies on a learning agent that can be taught by an expert to solve problems. First, the knowledge engineer and the subject matter expert develop an initial object ontology, which consists of hierarchical descriptions of objects and features from the application domain. Then, the expert teaches Disciple to solve problems in a way that resembles how the expert would teach a student or an apprentice. For instance, the expert defines a specific problem, helps the agent to understand each reasoning step toward the solution, and supervises and corrects the agent's behavior when it attempts to solve new problems. During this training process, the agent learns general problem solving rules from individual problem solving steps and the explanations of their success or failure. The key role in this multistrategy rule learning process is played by the object ontology, which is used as the generalization hierarchy.

In the next section we will illustrate how the incompleteness of the object ontology causes Disciple to learn rules with exceptions. In section 3 we will describe our mixed-initiative exception-based learning method and will show how it is integrated in the Disciple system. Then, in section 4, we will present an experiment performed at the US Army War College in Spring 2002, during which we evaluated this method. We will conclude the paper with a brief presentation of related research and conclusions.

2 Learning Rules with Exceptions

In the experiment performed at the US Army War College, military officers have taught personal Disciple agents to analyze center of gravity (COG) candidates for enemy and friendly forces at the strategic level of war. The center of gravity of a force (state, alliance, coalition, or group) represents the foundation of capability, power and movement, upon which everything depends (Clausewitz 1976). A force should concentrate its effort on its enemy's center of gravity, while adequately protecting its own.

During this experiment, the experts trained their Disciple agents using the task reduction paradigm (Tecuci et al. 2002). First, the expert has to formulate in natural language an initial problem solving task. Then he has to successively reduce this task to simpler tasks, guided by questions and answers, until a solution is found. Figure 1 shows an example of a task reduction step from the Center of Gravity analysis of the stabilization mission conducted in Grenada Island by USA and a union of several Caribbean States in 1983. This example consists of the current problem solving task, a question that is relevant to the reduction of this task, the answer to the question, and the subtask resulting from this answer.

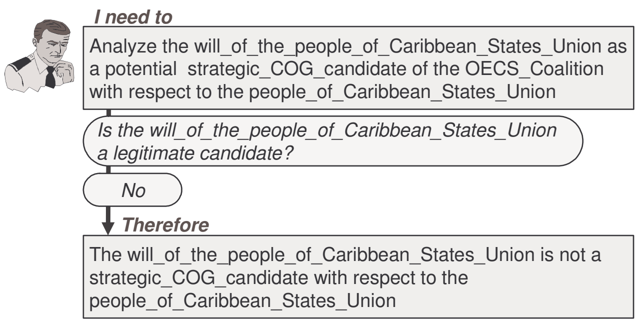

Figure 1: An example of a task reduction step

Based on this task reduction step, Disciple learns the general task reduction rule shown in Figure 2, through a mixed-initiative interaction. The rule consists of an informal structure shown in the top part of the figure, and a formal structure, shown in the bottom part. The informal structure of the rule preserves the expert's natural language from the example, and is used in the agent-user communication. The formal structure of the rule is used in the internal reasoning of the agent. The learned rule contains two applicability conditions: a plausible lower bound (PLB) condition and a plausible upper bound (PUB) condition, both approximating the exact applicability condition of the rule (Tecuci et al. 2002).

IF
Analyze the ?O2 as a potential strategic_COG_candidate of the ?O1 with respect to the ?O3

Question: *Is the ?O2 a legitimate candidate?*
Answer: *No*

THEN
The ?O2 is not a strategic_COG_candidate with respect to the ?O3

IF
Analyze the will of the people as a potential strategic COG candidate of a force with respect to the people of a force
The will is ?O2
The force is ?O1
The people is ?O3

Explanation
?O1 has_as_member?O4
?O4 has_as_people ?O3
?O3 has_as_will ?O2

PUB Condition	**PLB Condition**
?O1 is multi_member_force has_as_member ?O4	*?O1 is dominant_partner_multi_state_alliance has_as_member ?O4*
?O2 is will_of_agent	*?O2 is will_of_people*
?O3 is people has_as_will ?O2	*?O3 is people has_as_will ?O2*
?O4 is force has_as_people ?O3	*?O4 is single_state_force has_as_people ?O3*

THEN:
The will of the people is not a strategic_COG_candidate with respect to the people of a force
The will is ?O2
The people is ?O3

Figure 2: The rule learned from the example in Figure 1

The agent will apply the learned rule from Figure 2 to solve new problems and the feedback received from the expert will be used to refine it. For instance, this rule generates the problem solving step shown in Figure 3. However, the expert rejects it because the answer of the question and the resulting conclusion are wrong, the “will of the people of USA” being a legitimate COG candidate. Because the object ontology does not contain any relevant element that distinguishes between the examples shown in Figures 1 and 3, the rule cannot be specialized to uncover the incorrect reasoning step, which is kept as a negative exception of the rule. A ***negative exception*** is a negative example that is covered by the plausible lower bound condition of the rule and any rule's specialization that would uncover the exception would also result in the uncovering of some positive examples (Tecuci 1998).

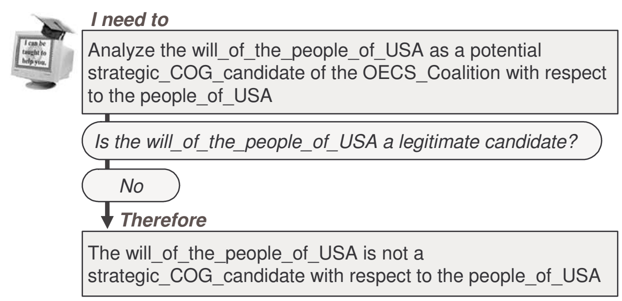

Figure 3: Incorrect example generated by the rule in Figure 2

3 Mixed-Initiative Exception-Based Learning

As discussed in the previous sections, an agent can learn rules with exceptions. Therefore, such an agent will face the problem of extending its representation language with new terms in order to eliminate the rules' exceptions and to improve the problem solving efficiency. Table 1 defines this general learning problem.

Table 1: The problem of exception-based learning

Given:
- an incomplete knowledge representation;
- a reasoning rule R containing negative exceptions.

Determine:
- new terms in the representation that differentiate between the positive examples and the negative exceptions of the rule R.

Result:

- an extended knowledge representation;
- a refined rule R with no or fewer exceptions.

To solve this problem, we have developed a mixed-initiative exception-based learning method. In this approach, the subject matter expert collaborates with the agent to analyze the negative exceptions of a rule, in order to discover possible ontology extensions (such as an additional object feature or feature value) that will eliminate the exceptions. This method synergistically integrates knowledge acquisition with machine-learning during a mixed-initiative human-agent interaction. It also uses several heuristics to perform plausible reasoning and knowledge base analysis.

The Exception-Based Learning method contains four main phases: 1) a **candidate discovery** phase in which the agent analyzes the rule and the ontology and finds the most plausible extensions of the ontology that may reduce or eliminate the exceptions; 2) a **candidate selection** phase in which the expert interacts with the agent to select one of the proposed candidates; 3) an **ontology refinement** phase in which the agent elicits knowledge from the expert to complete the description of the selected candidate and 4) a **rule refinement** phase in which the agent updates the rule and eliminates the rule's exceptions based on the performed ontology extension.

In the ***candidate discovery*** phase, the agent performs a heuristic analysis of the current knowledge base, trying to find plausible extensions of the ontology that may distinguish between all the positive examples of the rule, on one side, and its negative exceptions, on the other side. Table 2 presents the algorithm used in this phase. The method considers all the rule's variables that may be used to eliminate at least one exception. For each such variable it analyzes all the features that are applicable to its instances corresponding to the positive examples and the negative exceptions of the rule. For each candidate feature obtained, the method computes its plausibility, based on the number of exceptions that it may eliminate, and the number and type of facts that must be further elicited about it. These candidates are then ordered by their plausibility, to focus the analysis of the expert on the most plausible ontology extensions that may eliminate the exceptions of the rule.

In the ***candidate selection*** phase, Disciple proposes to the user the most plausible ontology candidates discovered and guides him to select one of them. In this mixed-initiative interaction, the user may ask the agent to filter or order the proposed candidates based on various criteria. For example, the user may select only the candidates for which he needs to define a new value of an existing feature.

Let us illustrate some of the ontology extension candidates discovered by the agent, in order to eliminate the negative exception from Figure 3. First, Disciple proposes ontology extension candidates for "Caribbean States Union" and "USA", in the form of distinct values for several features: "has as economy," "has as military contribution" and "has as strategic raw material." The expert may also define a new feature to distinguish between these two objects. Also, the agent proposes ontology extension candidates for the pair ("people of Caribbean States Union" "people of USA") and for the pair ("will of the people of Caribbean States Union" "will of the people of USA"), each pair corresponding to a different variable from the rule.

Table 2: Candidate discovery algorithm

Let R be a reasoning rule with the variables $(v_i)_{i=1,p}$ and the examples $(E_i)_{i=1,n}$;
$E_i = (v_1=o_{i,1},\ldots,v_p=o_{i,p})$, $o_{i,j}$ object in ontology;
PE = the set of positive examples;
NE = the set of negative exceptions;

Candidates$_k \leftarrow \varnothing$, for k = 1, p
for each rule variable $v_k \in \{v_1,\ldots,v_p\}$ **do**
  if $(\exists\, E_i \in NE)$ such that $o_{i,k} \notin \{o_{l,k} \mid E_l \in PE\}$ **then**
    for each $o_{j,k}$ with j = 1, n **do**
      for each $f \in$ Features, such that $o_{j,k} \in$ Domain(f) **do**
        if $\exists c \in$ Candidates$_k$, such that $f \in c$ **then**
          update the plausibility of $c(v_k, f)$
        else create candidate $c(v_k, f)$
          initialize the plausibility of $c(v_k, f)$
          add $c(v_k, f)$ to Candidates$_k$
  order Candidates$_k$ based on plausibility
Candidates $\leftarrow$ merge Candidates$_k$ for k = 1, p
return Candidates

Analyzing the proposed candidates, the expert decides to define a new feature that expresses the difference between the objects "Caribbean States Union" and "USA." What distinguishes them is that "Caribbean States Union" is a minor member of the "OECS Coalition," while "US" is not.

In the ***ontology refinement*** phase, the agent elicits from the expert the knowledge related to this selected candidate. Figure 4 shows the interface of the Exception-Based Learning module in this phase. The upper left panel of this module shows the negative exception which needs to be removed. Below the negative exception are the objects "Caribbean States Union" (from the positive example) and "USA" (from the negative exception), which are currently differentiated. The right panel shows the elicitation dialog. The expert is guided to specify the name and value of a new feature that capture the difference between "Caribbean States Union" and "USA." The expert defines the new relation "is minor member of" and specifies that its value for "Caribbean States Union" is "OECS Coalition." In this war scenario "USA" has no value for this feature because "USA" is actually the major member of the "OECS Coalition." Based on this elicitation, Disciple learns a general definition of the feature "is minor member of" and refines the ontology to incorporate this

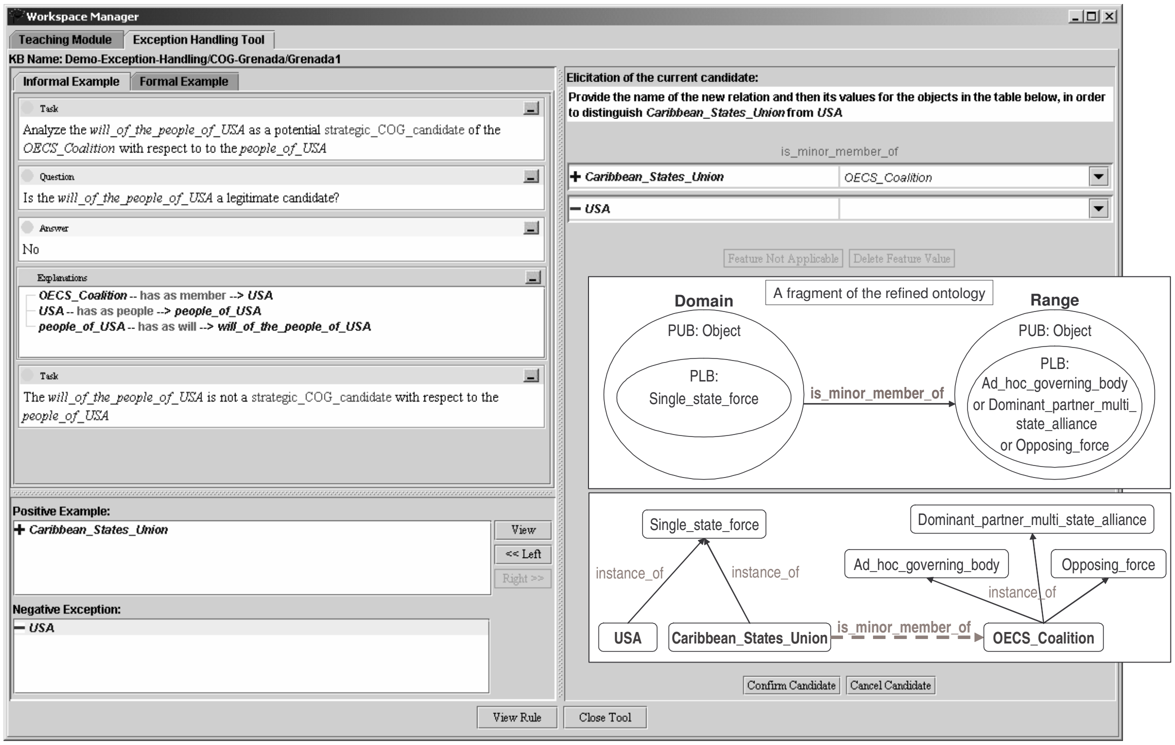

Figure 4: The interface of the Exception-Based Learning Module and a fragment of the refined ontology

knowledge. A fragment of the refined ontology is shown in the right part of Figure 4. Notice that both the domain and the range of the new feature "is minor member of" are represented as plausible version spaces. The plausible upper bound domain of this feature is "Object" and the plausible lower bound domain is "Single state force."

This candidate is used by the agent in the ***rule refinement*** phase to specialize the rule and to eliminate its negative exception. The newly defined feature "is minor member of" is used by Disciple to eliminate also the negative exceptions of other rules from the knowledge base (such an exception is: "the military of USA is not a strategic COG candidate"). This shows that the exceptions in the knowledge base are correlated. Therefore, by analyzing all the rules containing exceptions, one may discover a minimal set of ontology extensions which can eliminate all of them.

4 Experimental Results

In April-May 2002 we have completed an agent training experiment with Disciple at the US Army War College, as part of the "Military Applications of Artificial Intelligence" course. Seven teams comprising 15 senior military officers trained personal Disciple agents to identify and test strategic Center of Gravity candidates for various war scenarios.

During this experiment, the Exception-Based Learning module was used by seven subject matter experts with the assistance of a knowledge engineer, to remove the negative exceptions of the learned rules. We did not expect a significant number of missing elements, because before the experiment we attempted to develop a complete ontology, which contained 191 concepts and 206 features. However, during the experiment, 8 of the learned problem solving rules have accumulated 11 negative exceptions, indicating that some elements were not fully represented in the ontology. In order to eliminate these exceptions, the experts assisted by a knowledge engineer extended the ontology with 4 new features and 6 new facts (a fact has the form: object has feature with value). Some of the newly created features succeeded to eliminate the exceptions from several rules, which proved their general relevance. As a result of these ontology extensions, the rules were correspondingly refined.

Some of the experts' assessments of this module are presented in Figure 5. This experiment showed that the Exception-Based Learning method can extend the object ontology with new object features that represent better the subtle distinctions in the application domain. It allows the elimination or the reduction of the rules' exceptions. Thus, it improves the accuracy of the learned rules by refining their plausible version space conditions. It also improves the agent's problem solving efficiency by eliminating the need to explicitly check the exceptions.

5 Related Research and Conclusions

The presented method is an improvement of the Consistency-Driven Feature Elicitation method (Tecuci 1998; Tecuci and Hieb 1994). The Exception-Based Learning method proposes several possible ontology extensions based on their plausibility to eliminate the exceptions, instead of a single suggested extension proposed by the system. Also, our method is expert-oriented, instead of knowledge engineer oriented. Moreover, the Exception-Based Learning method considers a subset of the variables from the rule that are most plausible, while the Consistency-Driven Feature Elicitation method considers all the rule's variables, being less efficient. Additionally, our method considers ontology extensions for a subset of the positive examples and the negative exceptions and it allows the expert to choose the set of objects to be differentiated and the ontology extension to be performed.

Wrobel (1994) also addresses the problem of extending an incomplete representation language of a learning system to handle the exceptions of the learned rules, in the MOBAL system (Morik et al. 1993). The system contains a concept formation tool that supports a knowledge engineer in revising a knowledge base, by learning a concept definition that discriminates between the covered positive examples and all the exceptions of the rule. This concept is presented to the knowledge engineer, who may accept, modify or reject the system’s proposal. An advantage of our method is that it can define several features to distinguish more naturally the positive examples from the negative exceptions. Also, in our approach the subject matter expert plays a key role in the ontology extension process, while MOBAL’s tool is oriented toward a knowledge engineer.

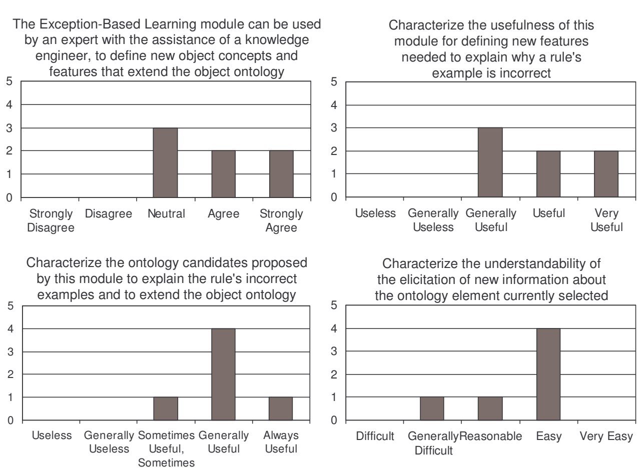

Figure 5: The assessments of the subject matter

The demand-driven introduction of new concepts or predicates has been exploited by other learning systems (Wnek and Michalski 1994; Muggleton 1994; Pazzani and Brunk 1991), in order to remove the inconsistencies from the representation language and to increase the efficiency of learning.

In conclusion, we have addressed in this paper the problem of learning with an evolving representation language. We presented a general approach to extend the knowledge representation of a rule-based system, which eliminates the exceptions of the reasoning rules. The performed experiments revealed that the rules learned from subject matter experts have a significant number of exceptions, indicating how the representation language should be extended to better represent the application domain.

We plan to extend the presented Exception-Based Learning method in several directions: use analogical reasoning and hints from the user in the discovery of ontology extension candidates; extend the method to discover new object concepts in order to remove the rules' exceptions; improve the methods that estimate the plausibility of the candidates; and extend the method to also remove the positive exceptions of the rules.

Acknowledgements. This research was sponsored by DARPA, AFRL, AFMC, USAF, under agreement number F30602-00-2-0546, by the AFOSR under grant no. F49620-00-1-0072, and by the US Army War College.

References

Buchanan, B.G., and Wilkins, D.C. eds. 1993. *Readings in Knowledge Acquisition and Learning: Automating the Construction and Improvement of Expert Systems.* San Francisco, CA: Morgan Kaufmann.

Clausewitz, C.V. 1976. *On War*. Translated and edited by Howard, M. and Paret, P. Princeton, NJ: Princeton University Press.

Ling, C.X. 1991. Non-monotonic Specialization. In *Proceedings of the First International Workshop on Inductive Logic Programming*, 59-68. Portugal.

Morik, K., Wrobel, S., Kietz, J., and Emde, W. 1993. *Knowledge Acqusition and Machine Learning – Theory, Methods and Applications.* London: Academic Press.

Muggleton S. 1994. Predicate invention and utilization. *Journal of Experimental and Theoretical Artificial Intelligence* 6(1):127-130.

Pazzani, M., and Brunk, C. 1991. Detecting and correcting errors in rule-based expert systems: an integration of empirical and explanation-based learning. *Knowledge Acquisition* 3:157-173.

Tecuci, G., and Hieb, M. 1994. Consistency-driven Knowledge Elicitation: Using a Learning-oriented Knowledge Representation that Supports Knowledge Elicitation in NeoDISCIPLE. *Knowledge Acquisition Journal* 6(1):23-46.

Tecuci, G. 1998. *Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies*. London, England: Academic Press.

Tecuci, G., Boicu, M., Marcu, D., Stanescu, B., Boicu, C., Comello, J., Lopez, A., Donlon, J., and Cleckner, W. 2002. Development and Deployment of a Disciple Agent for Center of Gravity Analysis. In *Proceedings of the Fourteenth Innovative Applications of Artificial Intelligence Conference,* 853-860. Menlo Park, CA: AAAI Press.

Wnek, J., and Michalski, R.S. 1994. Hypothesis-driven Constructive Induction in AQ17-HCI: A Method and Experiments. *Machine Learning* 14(2):139-168.

Wrobel, S. 1994. *Concept Formation and Knowledge Revision.* Dordrecht, Netherlands: Kluwer Academic Publishers.