Learning Agents Teachable by Typical Computer Users

Gheorghe Tecuci, Mihai Boicu, Dorin Marcu

Learning Agents Laboratory, Department of Computer Science, MSN 4A5, George Mason University, Fairfax, VA 22030
{tecuci, mboicu, dmarcu}@gmu.edu

Abstract

One way to identify new and promising directions for Machine Learning research is to define an important challenge problem that has broad applicability, and then to identify what kind of machine learning capabilities are required to solve it. This paper introduces a great challenge problem, developing agents by typical computer users, that has far reaching consequences on our society, and raises several Machine Learning research issues from a different paradigm than that used in current Machine Learning research. The paper identifies these issues and extends them to general research directions for Machine Learning. It also presents some preliminary results on approaching the challenge problem and these issues, in order to support their feasability.

Introduction

A great challenge for Artificial Intelligence is the development of theories, methods and tools that would allow users that do not have any special training in knowledge engineering or computer science, to build by themselves intelligent agents. We believe that providing solutions to this challenge problem will have an even greater impact on our society then the development of personal computers. Indeed, if personal computers allowed every person to be a computer user, without the need for special training in computer science, solutions to this challenge problem would allow any such person to be an agent developer, and the computers to become intelligent assistants, helping their users in a wide range of tasks. The key issue is that the development of such an agent should be as easy for the user as it currently is to use a word processor, an internet browser or an email program.

While requiring integrated research in various areas of Artificial Intelligence, this challenge problem is primarily addressed to the field of Machine Learning for which it raises several research issues that we believe must be solved using a new paradigm. We will not only introduce these research issues, as they arise in the context of the considered challenge problem, but we will also extend them to more general research directions for Machine Learning. We will also present in this paper some preliminary results in dealing with this challenge problem. We believe that attempting to solve it will lead to a broadening in scope of Machine Learning research, and will enable it to address significantly more advanced applications.

Research Problems for Machine Learning

If a typical computer user is to become able to develop an intelligent agent without any assistance from a knowledge engineer, agent development should only involve activities that are natural for such a user, and certainly computer programming or knowledge engineering, because of their complexity, cannot be among them.

One possible approach is to view agent development as a process similar to human teaching. In such a case, the starting point would be a learning agent shell that has some very limited yet generally applicable knowledge, coupled with the capability of learning from a person, in the same way a human student can learn from a human teacher. The agent will be taught by the user, evolving from the level of a primary school student (when it has little knowledge and very often asks "Why?"), to the level of an assistant (that assists the user by not only providing solutions to routine problems, but also by helping with suggestions in situations requiring creative solutions), and ultimately to the level of an expert for the type of problems it was trained to solve.

As mentioned above, solving this challenge problem raises several Machine Learning research issues from a different paradigm than that used in current Machine Learning research. At the basis of all the processes involved in this type of agent development, whether they concern user-agent communication, domain modeling, problem solving, or tutoring, there should be *mixed-initiative reasoning that will integrate complementary human and automated reasoning to take advantage of their respective reasoning styles and computational strengths.* The mixed-initiative reasoning should be based on a division of responsibility between the user and the agent for those elements of knowledge engineering for which they have the most aptitude, such that together they form a complete team for the development of the agent's knowledge base. This will require permanent coordination of the dialogue between the human and the agent, and continuous shift of the initiative and control.

Solving the proposed challenge problem requires *a deeper understanding of the relationship between teaching and learning*, and the development of synergistic methods where the user helps the agent to learn, and the agent helps the user to teach it. The field of Machine Learning could benefit from the research done in the field of Intelligent Tutoring Systems and, of course, the same is true for ITS. In general, *an integration of Machine Learning and Intelligent Tutoring Systems would be very promising research directions for both fields* (Hamburger and Tecuci, 1998). Consider, for instance, an approach where a teacher teaches the agent how to solve some class of problems, and then the agent teaches students in a way that is similar to how it was taught by their teacher (Tecuci and Keeling, 1999).

Another research issue is to *develop an integrated approach to domain modeling, problem solving and learning.* The agent has to acquire a representation of the model of the real world that exists only in the mind of the user who is teaching the agent. However, the teacher is a typical computer user, not a knowledge engineer. Therefore the user cannot be expected to be able to appropriately formalize his or her model of the expertise domain. On the other hand, the learning agent does not know what is to be formalized and has to get this information from the user. In domain modeling the user is the primary source of knowledge, but the agent should be able to help by performing analogies with other parts of the domain that have already been modeled. In problem solving these roles are reversed, with the agent being the primary reasoner, but receiving help for the more difficult problem solving situations. In all the situations the agent will learn from the contributions of the user. This requires the development of methods for reasoning and learning when dealing with more abstract and informal knowledge, as is the case in domain modeling. Because domain modeling is, in essence, a creative activity, this research issue is also an instance of a more general one: *How to learn to assist a user in a creative activity, what kind of assistance is useful and how to provide it?*

Yet another research issue is the development of learning methods that allow simultaneous learning of both the language of a domain (and even the language of a specific expert in that domain) and problem solving knowledge in that domain. Notice that this issue cannot be addressed only from the perspective of the current natural language processing systems that assume a predefined lexicon, grammar and associated semantics. While the syntax could be considered domain independent, a sizable portion of the lexicon and much of the semantics are domain specific, and would need to be learned from the user.

The most effective communication, however, is the one that integrates written (natural or more formal) language with several other communication modalities, such as speech, diagrams, or gesture. Therefore, a general research issue for Machine Learning research is *how an agent could learn to communicate with a user, using a combination of communication media* (written language, speech, diagram, gestures, etc.).

Another research problem for machine learning is how to define a learnable knowledge representation, that would represent knowledge at various levels of abstraction and formalizations needed for different processes, such as domain modeling (where the knowledge is informal), problem solving (that needs a formal representation), or communication (that has to be natural). A challenge for learning is to maintain the equivalence between these levels, as new knowledge is acquired at one of them. Another aspect is that the agent would need to be able to adapt its reasoning mechanisms to the corresponding level of the representation.

A related research issue is how an agent could learn from a more informed agent that uses a different knowledge representation. This should go beyond knowledge export and import, and may involve the same type of interactions, mixed-initiative reasoning, and integrated teaching and learning, that is envisioned for solving the proposed challenge problem where an agent learns from a human. In addition, this process may itself be mediated by a human.

A great opportunity for machine learning that has been recognized a long time ago, but it is not receiving much attention in the current machine learning research, is *the automation of the knowledge acquisition process*. An important aspect of the proposed challenge problem is that it implicitly provides a solution to the knowledge acquisition bottleneck. Indeed, this would allow an expert to teach his or her knowledge to the agent, without any assistance from a knowledge engineer. In general, *the use of learning to facilitate the acquisition of knowledge from single domain experts, from teams of knowledge engineers and domain experts, or from collaborating domain experts (that are assisted or not by knowledge engineers) is an important and timely research issue.*

Another great opportunity for machine learning research is in *facilitating the maintenance of knowledge based systems.* It is estimated that around 80% of the effort spent during the life-time of a system is devoted to system's maintenance. The development of agents that learn from their users (or their environment) eliminates the distinction that is currently made in software and knowledge engineering between system development and system maintenance. Indeed, because the whole process of agent development is one of creating and adapting knowledge pieces in its knowledge base, this creation and adaptation may also occur in response to changes in the application environment or goals of the agent, which are the primary reasons for agent maintenance.

The development of the kind of instructable agents proposed in this paper requires a change in the paradigm of the current machine learning research. Most of this current research has a very narrow focus, generally addressing only one activity. Here we propose however that learning should be a basic ingredient of a wide range of integrated activities, including communication, domain modeling, problem-solving, and tutoring.

Finally, we believe very strongly that *the machine learning research has to pay much more attention to addressing real world problems.* This is the best way to identify new research directions, and to test the developed learning methods.

The idea of an agent that learns directly from a user is not new. While first mentioned by McCarthy (1958), it has received increasing attention after the influential paper on the learning apprentice system LEAP (Mitchell et al., 1985). Since then several prototype learning apprentice systems have been developed (Tecuci, 1988; Wilkins, 1990; De Raedt, 1991). Currently this problem is largely ignored, being considered too difficult for real-world domains. There are, however, several things about the challenge problem proposed in this paper that are novel from a machine learning point of view:

- The user from which the agent learns is a typical computer user. The type of user was not a concern in the previous apprentice systems. As in many other cases of machine learning research the assumption was that the learner always receives the input it needs, with no real concern of how and by whom this input is produced.
- The types of user-learner interactions are very limited in the developed interactive learning systems, generally the user being regarded as an example initiator or as an oracle. We propose to research complex multi-modal interactions, that not only expends on the type of written language interactions, but also consider interactions through speech, diagrams and images.
- The type of learning performed by a typical machine learning system is often very limited. In general, the representation language is considered a priori defined, and learning mostly consists in generalizing examples, revising rules or acquiring new facts. We propose that learning should be regarded as a simultaneous process of acquiring both the representation language, and knowledge in that language. For instance, the agent would learn both new concepts in its ontology, and problem solving rules expressed with these concepts.
- Genuine *mixed-initiative reasoning* is not really present in the developed learning systems. This type of reasoning is at the very basis of the research directions proposed in this paper.
- Learning was not significantly integrated with other processes that are very important in agent development by a typical user, such as domain modeling, multi-modal communication, or tutoring.
- From the point of view of practical applicability, previously developed learning apprentice systems are very limited. The proposed research directions emphasize scalability and applicability to real world problems.

Much of the narrowness of the current machine learning research may be attributed to the fact that some important and broad research issues have been considered very difficult to tackle. Therefore, in the rest of this paper we present our current results in dealing with the proposed challenge problem, to demonstrate that addressing it is both feasible and timely, and will be very stimulating in broadening the scope of the current machine learning research.

The Disciple Approach to the CP

Our approach to developing agents by non-programmers, called Disciple (Tecuci, 1998), relies on building a series of increasingly more capable learning and reasoning agents that can be taught to solve problems in an application domain by a user that is an expert in that domain, but does not have knowledge engineering or computer science experience. While in the previous versions of Disciple the expert required significant assistance from a knowledge engineer, the new versions attempt to reduce the amount of assistance needed. The Disciple agent learns from the expert, developing its knowledge base that consists of an ontology that defines the terms from the application domain, and a set of general problem solving rules expressed with these terms.

The agent development process includes importing ontological knowledge from existing repositories of knowledge, such as CYC (Lenat, 1995) or Loom (MacGregor, 1999), and teaching the agent how to perform various tasks, in a way that resembles how the expert would teach a human apprentice. This is a mixed-initiative approach, premised upon a division of responsibility between the expert and the agent where each is accorded responsibility for those elements of knowledge engineering for which they have the most aptitude, and together they form a complete team for knowledge base development.

The approach is based on several levels of synergism between the expert that has the knowledge to be formalized and the agent that is able to formalize it. At the highest level there is the synergism in solving complex problems, where the agent contributes routine and innovative problem solving steps and the expert contributes inventive and creative ones. At the next level down, there is the synergism between teaching and learning, where the expert helps the agent to understand the problem solving steps contributed by him or her, and the agent learns general problem solving rules that will allow it to apply similar steps in future problem solving situations. Finally, at the lowest level, there is the synergism between different learning strategies employed by the agent to learn from the expert in situations in which no single strategy learning method would be sufficient. In this way, the agent learns continuously from the expert, building, refining, verifying and improving its knowledge base.

Current Results

Over the years, the Disciple approach has been developed and scaled-up continuously, most recently as part of the 1997-1999 High Performance Knowledge Bases (HPKB) program supported by DARPA and AFOSR (Cohen et al., 1998). With respect to the Disciple approach and its most recent implementations, we formulate the following claims that have been tested during the intensive evaluations of the HPKB program:

- they significantly speed up the process of building and updating a high performance knowledge base;
- they enable rapid learning of problem solving knowledge from domain experts, with limited assistance from knowledge engineers;

- the learned problem solving knowledge is of a good enough quality to assure a high degree of correctness of the solutions generated by the agent;
- the acquired problem solving knowledge assures a high performance of the problem solver.

The organizations participating in HPKB were given the challenge of rapidly developing and updating knowledge-based systems for solving specially designed challenge problems. The aim of HPKB was to test the claim that, with the latest AI technology, large knowledge bases can be built and updated quickly and efficiently.

A challenge problem for the first phase of the HPKB program was to rapidly build and maintain a knowledge-based workaround agent that is able to plan how a convoy of military vehicles can “work around” (i.e. circumvent or overcome) obstacles in their path, such as damaged bridges or minefields. To solve this challenge problem we developed the Disciple-Workaround learning agent, demonstrating that a knowledge engineer can rapidly teach Disciple, using Military Engineering manuals and sample solutions provided by a domain expert. During the 17 days of DARPA’s 1998 evaluation, the knowledge base of Disciple was increased by 72% (from the equivalent of 5,920 simple axioms to 10,162 simple axioms) with almost no decrease in performance. An interesting result of this evaluation was that Disciple-Workaround generated some new correct solutions that were not considered by the evaluating experts. The Disciple agent also achieved the best scores among all the teams that participated in the workaround challenge problem, and was selected to represent the HPKB program at EFX’98, the Air Force’s show case of the most promising technologies. It is interesting to notice that the other teams that attempted this challenge problem, did not use a Machine Learning approach.

A challenge problem for the second phase of the HPKB program was to rapidly develop and maintain a critiquing agent to evaluate military Courses of Action that were developed as hasty candidate plans for ground combat operations. To solve this challenge problem we developed the Disciple-COA learning agent that identifies the strengths and the weaknesses of a course of action with respect to the principles of war and the tenets of army operations. In the process of developing Disciple-COA we achieved two significant milestones. For the first time we developed the knowledge base around an ontology created by another group (Teknowledge and CYCorp), demonstrating both the feasibility of knowledge reuse with the Disciple approach, and the generality of the Disciple rule learning and refinement methods. Moreover, the Disciple-COA agent was taught even more rapidly than the Disciple-Workaround agent. In this case, Disciple was taught jointly by a domain expert and a knowledge engineer, and its knowledge base increased by 46% in 8 days of evaluation, from a size of 6,229 simple axioms equivalent to a size of 9,092 simple axioms equivalent. The final knowledge base contained 801 concepts, 444 object and task features, 360 tasks and 342 task reduction rules. Also, each COA was represented with around 1,500 facts. Disciple-COA again demonstrated a significantly higher performance than the other developed critiquers. The other research groups that developed COA critiquers as part of the HPKB program were 1) Teknowledge and Cycorp that developed a critiquer based on the CYC system (Lenat, 1995), taking advantage of CYC’s large knowledge repository and inferential capabilities; 2) the Expect group from ISI that based its critiquer on the Expect shell for problem solving and knowledge acquisition (Kim and Gil, 1999); and 3) the LOOM group from ISI that developed a case-based critiquer as an extension to the LOOM system (MacGregor, 1999). Again, with the exception of Disciple-COA, none of these critiquers were based on machine learning.

During August 1999 we conducted a special one week knowledge acquisition experiment with Disciple-COA, at the US Army Battle Command Battle Lab, in Fort Leavenworth, Kansas. In this experiment that was entirely video-taped, four military experts that did not have any prior knowledge engineering experience received around 16 hours of training in Artificial Intelligence and the use of Disciple-COA. Then they trained Disciple-COA without receiving any significant assistance from knowledge engineers. During three hours of training the knowledge base Disciple-COA was extended with the equivalent of around 275 simple axioms. At the end of the experiment the experts completed a detailed questionnaire inquiring about their perceptions of the usefulness and usability of the Disciple tool and the Disciple critiquer. An analysis of their answers revealed very high scores for the Disciple approach (82.39% on the fitness of the Disciple critiquing agent for use in their organizations, 76.32% in the effect that Disciple-COA would have on their task performance, and 73.72% in system’s usability). One of the domain experts, LTC John N. Duquette, Chief of the Experimentation Division of BCBL stated: "The potential use of this tool by domain experts is only limited by their imagination—not their AI programming skills."

Conclusions

In this paper we have introduced a new research problem for machine learning: developing learning agents that can be taught by typical computer users that do not have prior knowledge engineering experience, and do not receive any help from a knowledge engineer.

We have shown that addressing this research problem raises several new research issues in machine learning:

- How to synergistically integrate teaching and learning (and, in general, the fields of Machine Learning and Intelligent Tutoring Systems)?
- How to develop mixed-initiative learning methods that synergistically integrate human and automated reasoning to take advantage of their respective reasoning styles and computational strengths?

- How to genuinly integrate domain modeling, problem solving and learning in the agent development process?
- How to learn to assist a user in a creative activity, what kind of assistance is useful and how to provide it?
- How to learn a protocol to exchange information between the user and the agent in a form that one can easily create and the other can easily understand?
- How could an agent learn simultaneouly both the language of a domain (and even of a specific expert in that domain) and to solve problems in that domain?
- How could an agent learn to communicate with a user based on a combination of communication media?
- How to define a learnable knowledge representation, that would represent knowledge at various levels of abstraction and formalizations needed for different agent development processes, such as domain modeling (where the knowledge is informal), problem solving (that needs a formal representation), or communication (that has to be natural)?
- How could an agent learn from a more informed agent that uses a different knowledge representation?
- How could learning facilitate the acquisition of knowledge from single domain experts, from teams of knowledge engineers and domain experts, or from collaborating domain experts (that are assisted or not by knowledge engineers)?
- How could learning facilitate the maintenance of knowledge based systems?

We have also reported some results in dealing with the proposed challenge problem, to demonstrate that addressing it is both feasible and timely.

Finally, we have stressed the great importance of addressing real world problems. They are one of the best sources for new research directions, and the ultimate test of our research results.

Acknowledgments. This research was supported by the AFOSR grant F49620-97-1-0188, as part of the DARPA's High Performance Knowledge Bases Program. Mike Bowman, Cristina Cacaval, Florin Ciucu, Cristian Levcovici, Liviu Panait, and Bogdan Stanescu have contributed to the current version of the Disciple approach.

References

Cohen P., Schrag R., Jones E., Pease A., Lin A., Starr B., Gunning D., and Burke M. (1998). The DARPA High-Performance Knowledge Bases Project, *AI Magazine*, 19(4),25-49.

McCarthy, J. (1958). Programs with Common Sense. In Proceedings of the Teddington Conference on the Mechanization of Thought Processes.

De Raedt, L. (1991). *Interactive Concept Learning*. Ph.D. Thesis, Catholic University of Leuven.

Hamburger H. and Tecuci G. (1998). Toward a Unification of Human-Computer Learning and Tutoring, In Goettl, B.P., Halff, H.M., Redfield, C.L. and Shute, V.J. (editors), *Intelligent Tutoring Systems*, pp. 444-453, Berlin: Springer-Verlag.

Kim, J. and Gil, Y. (1999). Deriving Expectations to Guide Knowledge Base Creation, in *AAAI-99/IAAI-99 Proceedings*, AAAI Press, Menlo Park, CA.

Lenat, D.B. (1995). CYC: A Large-scale Investment in Knowledge Infrastructure *Comm of the ACM* 38(11):33-38.

MacGregor, R. (1999). *Retrospective on LOOM*. Available online as: http://www.isi.edu/isd/LOOM/papers/macgregor/Loom_Retrospective.html

Mitchell, T. M., Mahadevan, S., and Steinberg, L. I. (1985). LEAP: A Learning Apprentice System for VLSI Design, In proc. IJCAI-85.

Tecuci, G. (1988). Disciple: A Theory, Methodology and System for Learning Expert Knowledge. *Doctoral Dissertation*. University of Paris South.

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

Tecuci G. and Keeling H. (1999). Developing an Intelligent Educational Agent with Disciple, *International Journal of Artificial Intelligence in Education*, vol. 10, no.3-4.

Tecuci, G., Boicu, M., Wright, K., Lee, S.W., Marcu, D. and Bowman, M. (1999). An Integrated Shell and Methodology for Rapid Development of Knowledge-Based Agents, in *AAAI-99/IAAI-99 Proceedings*, AAAI Press, Menlo Park, CA.

Wilkins, D.C. (1990). Knowledge Base Refinement as Improving an Incorrect and Incomplete Domain Theory, In Kodratoff, Y. and Michalski, R. S. (editors), *Machine Learning: An Artificial Intelligence Approach Volume 3*, Morgan Kaufmann. San Mateo, CA.