Claims and Challenges in Evaluating Human-Level Intelligent Systems

John E. Laird[*], *Robert E. Wray III*[**], *Robert P. Marinier III*[*], *Pat Langley*[***]

[*]Division of Computer Science and Engineering, University of Michigan, Ann Arbor, MI 48109-2121
[**]Soar Technology, Inc., 3600 Green Court, Suite 600, Ann Arbor, MI 48105
[***]School of Computing and Information, Arizona State University, Tempe, AZ 85287-8809

Abstract

This paper represents a first step in attempting to engage the research community in discussions about evaluation of human-level intelligent systems. First, we discuss the challenges of evaluating human-level intelligent systems. Second, we explore the different types of claims that are made about HLI systems, which are the basis for confirmatory evaluations. Finally, we briefly discuss a range of experimental designs that support the evaluation of claims.

Introduction

One of the original goals of Artificial Intelligence (AI) was to create systems that had general intelligence, able to approach the breadth and depth of human-level intelligence (HLI). In the last five years, there has been a renewed interest in this pursuit with a significant increase in research in cognitive architectures and general intelligence as indicated by the first conference on Artificial General Intelligence. Although there is significant enthusiasm and activity, to date, evaluation of HLI systems has been weak, with few comparisons or evaluations of specific claims, making it difficult to determine when progress has been made. Moreover, shared evaluation procedures, testbeds, and infrastructure are missing. Establishing these elements could bring together the existing community and attract additional researchers interested in HLI who are currently inhibited by the difficulty of breaking into the field.

To confront the issue of evaluation, the first in a series of workshops was held in October 2008 at the University of Michigan, to discuss issues related to evaluation and comparison of human-level intelligent systems. This paper is a summarization of some of the discussions and conclusions of that workshop. The emphasis of the workshop was to explore issues related to the evaluation of HLI, but to stop short of making proposals for specific evaluation methodologies or testbeds. That is our ultimate goal and it will be pursued at future workshops. In this first workshop, we explored the challenges in HLI evaluation, the claims that are typically made about HLI, and how those claims can be evaluated.[1]

Challenges in Evaluating HLI Systems

Defining the goal for HLI

One of the first steps in determining how to evaluate research in a field is to develop a crisp definition its goals, and if possible, what the requirements are for achieving those goals. Legg and Hutter (2007) review a wide variety of informal and formal definitions and tests of intelligence. Unfortunately, none of these definitions provide practical guidance in how to evaluate and compare the current state of the art in HLI systems.

Over fifty years ago, Turing (1950) tried to finesse the issue of defining HLI by creating a test that involved comparison to human behavior, the Turing Test. In this test, no analysis of the components of intelligence was necessary; the only question was whether or not a system behaved in a way that was indistinguishable from humans. Although widely known and popular with the press, the Turing Test has failed as a scientific tool because of its many flaws: it is informal, imprecise, and is not designed for easy replication. Moreover, it tests only a subset of characteristics normally associated with intelligence, and it does not have a set of incremental challenges that can pull science forward (Cohen, 2005). As a result, none of the major research projects pursuing HLI use the Turing Test as an evaluation tool, and none of the major competitors in the Loebner Prize (an annual competition based on the Turing Test) appear to be pursuing HLI.

One alternative to the Turing Test is the approach taken in cognitive modeling, where researchers attempt to develop computational models that think and learn similar to humans. In cognitive modeling, the goal is not only to build intelligent systems, but also to better understand human intelligence from a computational perspective. For this goal, matching the details of human performance in terms of reaction times, error rates, and similar metrics is an appropriate approach to evaluation. In contrast, the goal of HLI research is to create systems, possibly inspired by humans, but using that as a tactic instead of a necessity. Thus, HLI is not defined in terms of matching human

[1] For an in depth and more complete discussion of evaluation of AI systems in general, see Cohen (1995).

reaction times, error rates, or exact responses, but instead, the goal is to build computer systems that exhibit the full range of the cognitive capabilities we find in humans.

Primacy of Generality

One of the defining characteristics of HLI is that there is no single domain or task that defines it. Instead, it involves the ability to pursue tasks across a broad range of domains, in complex physical and social environments. An HLI system needs broad competence. It needs to successfully work on a wide variety of problems, using different types of knowledge and learning in different situations, but it does not need to generate optimal behavior; in fact, the expectation is it rarely will. This will have a significant impact on evaluation, as defining and evaluating broad competency is more difficult than evaluating narrow optimality.

Another aspect of generality is that, within the context of a domain, an HLI system can perform a variety of related tasks. For example, a system that has a degree of competence in chess should be able to play chess, teach chess, provide commentary for a chess game, or even develop and play variants of chess (such as Kriegspeil chess). Thus, evaluation should not be limited to a single task within a domain.

Integrated Structure of HLI Systems

Much of the success of AI has been not only in single tasks, but also in specific cognitive capabilities, such as planning, language understanding, specific types of reasoning, or learning. To achieve HLI, it is widely accepted that a system must integrate many capabilities to create coherent end-to-end behavior, with non-trivial interactions between the capabilities. Not only is this challenging from the standpoint of research and development, but it complicates evaluation because it is often difficult to identify which aspects of a system are responsible for specific aspects of behavior.

A further complication is that many HLI systems are developed not by integrating separate implementations of the cognitive capabilities listed earlier, but instead by further decomposing functionality into more primitive structures and process, such as short-term and long-term memories, primitive decision making and learning, representations of knowledge, and interfaces between components, such as shown in Figure 1. In this approach, higher-level cognitive capabilities, such as language processing or planning are implemented in a fixed substrate, differing in knowledge, but not in primitive structures and processes. This is the cognitive architecture approach to HLI development (Langley, Laird, & Rogers, in press), exemplified by Soar (Laird, 2008) and ICARUS (Langley & Choi, 2006).

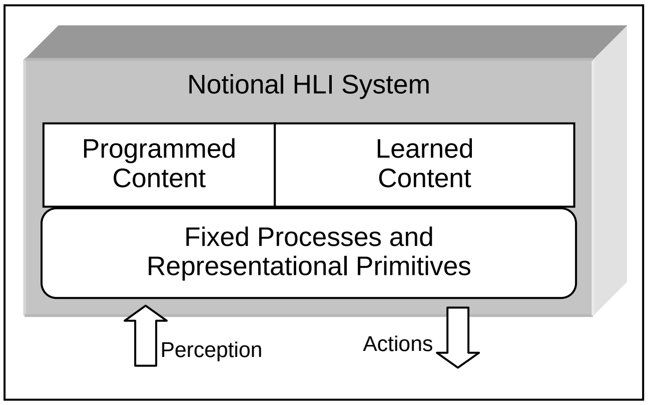

Figure 1: Structure of a Notional HLI System.

This issue is clear in research on cognitive architectures because they make the following distinctions:

- The architecture: the fixed structure that is shared across all higher-level cognitive capabilities and tasks.
- The initial knowledge/content that is encoded in the architecture to achieve capabilities and support the pursuit of a range of tasks.
- The knowledge/content that is learned through experience.

For any HLI system, it is often difficult to disentangle the contributions of the fixed processes and primitives of the systems to the system's behavior from any initial, domain-specific content and the learned knowledge, further complicating evaluation. There is a concern that when evaluating an agent's performance, the quality of behavior can be more a reflection of clever engineering of the content than properties of the HLI systems itself. Thus, an important aspect of evaluation for HLI is to recognize the role of a prior content in task performance and attempt to control for such differences in represented knowledge.

Long-term Existence

Although not a major problem in today's implementations, which typically focus on tasks of relatively short duration, HLI inherently involves long-term learning and long-term existence. It is one thing to evaluate behavior that is produced over seconds and minutes – it is possible to run many different scenarios, testing for the effects of variation. When a trial involves cumulative behavior over days, weeks, or months, such evaluation becomes extremely challenging due to the temporal extents of the experiments, and the fact that behavior becomes more and more a function of experience.

Claims about HLI

We are primarily interested in constructive approaches to HLI; thus, our claims are related to the functionality of our systems, and not their psychological or neurological realism. Achieving such realism is an important scientific goal, but one of the primary claims made by many practitioners in HLI is that it can be achieved without an exact reimplementation of the human mind and/or brain.

A major step in empirical evaluation is to consider the claims we want or expect to make about the systems we develop. Only by knowing these claims can we define appropriate experiments that test those claims and let us determine what we need to measure in those experiments. An explicit claim (hypothesis) is usually about the relationship between some characteristic of a HLI system and its behavior. To test the hypothesis, a manipulation experiment can be performed in which the characteristic (the independent factor) is varied and changes in behavior along some dimensions (dependent variables) are measured. Many of the difficulties described earlier arise because of the types of claims that we as researchers are attempting to make in the pursuit of HLI.

HLI System Claims

There are varieties of claims that can be made about HLI at the systems level. We highlight four types of claims below:

1. A computer system (HLI1) can achieve some type of behavior or cognitive capability related to HLI. There are many examples of this type of claim from the early history of cognitive architectures. For example, cognitive architectures were used to illustrate capabilities such as associative retrieval and learning, improvements in performance with experience, and the "emergence" of so-called high-level capabilities, such as planning or natural-language comprehension, from the primitives of the architecture. Such a claim is invariably a sufficiency claim, where the structure of the agent is not claimed to be the only way of achieving the desired behavior (a necessity claim). These claims are generally made within the context of some class of environments, tasks, and an agent's ability to interact with its environment. A few cases where necessity claims have been made about the general properties of architectures such as Newell and Simon's (1976) symbol system hypothesis.
2. A modification of a system (HLI1′) leads to expanding the set of problems the system can solve or improving behavior along some dimension related to HLI across a range of tasks (see section on dependent variables for a discussion of metrics related to behavior). For example, the progression from version 7 of Soar to version 8 led to improvements in system robustness and learning capability in Soar (Wray & Laird, 2003). This is probably the most common claim, as it is part of the standard practice of systematically improving and extending the capabilities of an HLI system.
3. One system (HLI1) differs from another system (HLI2) in the set of problems that can be solved or in its performance. This claim usually involves comparing two systems and is currently less common, as it involves creating two separate systems and applying them to similar tasks. Once notable example of systematic comparison was the Agent Modeling and Behavior Representation (AMBR) program, sponsored by the Air Force Research Laboratory (Gluck & Pew, 2005). AMBR compared four different architectures on a few tasks in a single task domain. One lesson of AMBR is the importance and difficulty of controlling for a priori content, as suggested previously. The HLI community is capable of descriptive and analytical comparisons of architectures (e.g., see Jones & Wray, 2006) but empirical comparison of architectures and HLI systems (as opposed to example instances in single task domains) is currently infeasible.
4. One system (HLI1) has behavior similar along some relevant dimension to human behavior (H1). This is a special case of 3 above, where human behavior provides the target metric and the emphasis is usually on similarity. Even though we are concentrating on the functionality of HLI systems, humans often provide the best yardstick for comparison. However, even in the cognitive modeling community, evaluation is typically focused on model evaluation rather than evaluation of the underlying system. Anderson and Lebiere (2003) offer suggestions for more systematic evaluation of the paradigm supporting task models, which may also provide a framework for a near-term, descriptive approach to HLI evaluation.

There will often be a hierarchy of claims. Stronger claims are usually reserved for general properties of the architecture, such as that symbol systems are necessary in order to achieve general competence. Claims about the general properties of a specific component in relation to achieving competency will usually be weaker sufficiency claims. One can also make claims about specific algorithms and data structures, such as that the RETE algorithm achieves nearly constant match time even as the number of rules grows (Doorenbos, 1994).

Independent Variables

Central to claims are that there is some relationship among different variables, in particular that varying the *independent* variables leads to changes in the *dependent* variables. In HLI systems, the independent variables often fall in three classes:

- Components of the overall system: As components or modules are added, removed, or modified, it is claimed that there will be changes in behavior. With these types of independent variables, there often is not an ordering – these are categorical and not numeric, so that results are not summarized on a graph in which the values of the dependent variables can be connected with lines.
- Amount of knowledge: Knowledge is varied to determine how effectively the system can use or process knowledge to guide its behavior. One challenge is to compare knowledge across systems, given their different representations. However, within a given architecture, it is usually easy to measure the impact of knowledge by

comparing behavior with and without specific knowledge elements.
- System parameters: Many systems have parameters that affect their behavior, such as the learning rate in a reinforcement learning agent. This leads to parametric studies that involve systematic variation of system parameters. The current state-of-art in computational cognitive modeling provides examples of how much parametric exploration is possible and offers glimpses into how those explorations can inform one's evaluation of contributions to overall behavior.

In addition to changes in the HLI systems, many claims concern how changes in the environment or task influence the behavior of an HLI system. For such claims, the independent variables are properties of the environment and task. These are often more difficult to vary systematically.

Examples of environmental independent variables include:
- Experience in an environment: For claims related to efficacy of learning, independent variables can be the amount of experience, the time existing in the environment, and related properties.
- Complexity of the environment: Analysis of how a system responds as one changes the number of objects, their relations and properties, and the types of interactions between objects.
- Accessibility of the environment: The kinds of information can be known/perceived at any time.
- Indeterminacy in environmental interaction: The ease of unambiguously sensing and acting in the environment.
- Dynamics of the environment: How different aspects of the environment change independently of the HLI system and how fast the environment changes relative to the basic processing rate of the system.

Examples of task-related independent variables include:
- Whether the task requires satisficing vs. optimizing. Given the competing constraints on HLI systems, often the goal is to satisfice.
- Complexity of the task: How many goals/subgoals must be achieved? What interdependences are there between goals?
- Length of existence: How long does the system behave in the environment in ways that put significant stress on its ability to respond quickly to environmental dynamics?

Dependent Variables: Metrics

Dependent variables allow measurement of properties of behavior relevant to evaluating claims. These metrics can be either quantitative or qualitative, and we evaluations will often involve multiple metrics.

We do not consider properties of theories of HLI, such as parsimony, because they relate to claims about theories as opposed to properties of the HLI system.

Our analysis of metrics is split into two parts. The first addresses concrete metrics that directly measure some aspect of behavior, such as solution quality, while the second will address metrics that cover abstract properties of HLI systems that cannot be measured directly, such as robustness and flexibility.

Concrete metrics:

- **Performance** includes measures such as solution time, quality of solution, and whether or not a solution is found. These are the standard metrics used in evaluating AI systems. One must careful when using CPU time because of variation in the underlying hardware. Usually solution time will be in some hardware independent measure (such as nodes expanded in a search) that can then be mapped to specific hardware.
- **Scalability** involves change in some performance variable as problem complexity changes. Scalability is an important metric for HLI systems because of the need for large bodies of knowledge acquired through long-term learning. Other scalability issues can arise from interacting with complex environments where the number of relevant objects varies.

Evaluating only concrete metrics of behavior poses the danger of driving research toward engineering optimizations. Behavioral evaluations should include both a notion of behavior (e.g., learning optimization) and what goes in (level of programming, research, etc.). Current practice is usually just to measure behavior. However, a general claim is that an HLI approach should decrease the amount of re-engineering (what goes in) required for a task. Thus, there are other metrics that are typically independent variables (varied during testing to determine their effect on performance and scalability) but that can become dependent variables if the experiment is set up to determine when a certain level of performance is achieved. For example, one could measure how much knowledge or training is required to achieve a certain level of performance or how much additional knowledge and training (or re-engineering of the architecture) is required to perform on a new task, a property termed *incrementality* (Wray & Lebiere, 2007).

Abstract metrics

Concrete metrics have the advantage that they are usually easy to measure; however, many of the claims about HLI systems are not directly grounded in concrete metrics such as performance measures or scalability. Usually claims concern more abstract properties, such as generality, expressiveness, and robustness. Abstract metrics are often properties that involve integration of multiple properties across multiple trials and even across multiple tasks and domains. One challenge is to determine how to ground these abstract metrics in measurable properties of HLI systems' behavior.

- **Task and Domain Generality**: How well does a system (or architecture) support behavior across a wide range of tasks and domains? Concerns about task and domain generality are one of the primary factors that distinguish research in HLI from much of the other research in AI. This requires measures of diversity of tasks and domains, which are currently lacking. Given the primacy of generality, it is not surprising that many other abstract metrics address aspects of behavior and system construction that are related to generality.
- **Expressivity**: What kinds or range of knowledge can an HLI system accept and use to influence behavior? This relates to generality because restrictions on expressiveness can, in turn, restrict whether a system can successfully pursue a task in a domain. For example, systems that only support propositional representations will have difficulty reasoning about problems that are inherently relational.
- **Robustness**: How does speed or quality of solutions change as a task is perturbed or some knowledge is removed or added? One can also measure robustness of an architecture – how behavior changes as an aspect of the architecture is degraded – but this is rarely considered an important feature of HLI systems. Instead, the interest lies in how well the system can respond to partial or incomplete knowledge, incorrect knowledge, and changes in a task that require some mapping of existing knowledge to a novel situation.
- **Instructability:** How well can a system accept knowledge from another agent? Instructability emphasizes acquiring new skills and knowledge, as well as acquiring new tasks. Finer-grain measures of instructability include the language needed for instruction, the breadth of behavior that can be taught, and the types of interactions supported, such as whether the instructor is in control, whether the agent is in control, or whether dynamic passing of control occurs during instruction.
- **Taskability**: To what extent can a system accept and/or generate, understand, and start on a new task? Taskability is related to instructability, but focuses working on new tasks. Humans are inherently taskable and retaskable, being able to attempt new tasks without requiring a external programmer that understands its internal representations. Humans also generate new tasks on their own. In contrast, most current systems only pursue the tasks and subtasks with which they were originally programmed and cannot dynamically extend the tasks they pursue.
- **Explainability**: Can the system explain what it has learned or experienced, or why it is carrying out some behavior? Humans do not have "complete" explanability – the ability to provide justifications for all decisions leading up to external behavior – so this capability is a matter of degree.

Conclusion

It is obvious that developing human-level intelligence is a huge challenge. However, important parts of that scientific and engineering enterprise are the methods and practices for evaluating the systems as they are developed. In this paper, we present some of the primary challenges that arise in evaluation that distinguish it from research on more specialized aspects of artificial intelligence. We also attempt to characterize the types of scientific claims that arise in research on HLI, distinguishing different classes of claims that can be made at the system level, and then further analyzing the independent and dependent variables of those claims.

Having clear, explicit claims has always been a critical part of scientific progress, and we encourage researchers to be more explicit in the claims of the theories and systems they develop. This not only helps ourselves in designing appropriate experiments, it also makes it much easier for other researchers to evaluate the contribution of a piece of work. In addition to being specific about claims, the field also needs shared notions of methodologies and metrics associated with evaluating those claims. The abstract metrics enumerated here suggest some ways in which HLI researchers can begin to better distinguish these systems from more traditional AI systems. However, much work remains to identify methods for measuring and evaluating these capabilities.

The next step for this effort is to explore tasks and environments that can be shared across the community. Given the broad goals of HLI research, we need multiple testbeds that support environments in which many tasks can be pursued, and which include tools for performing experimentation and evaluation. Working on common problems will simplify cross-system evaluation and collaboration, both important steps toward developing human-level intelligence.

Acknowledgments

This paper is based on discussions at a workshop on evaluation of human-level intelligence, held at the University of Michigan in October, 2008. Participants included Joscha Bach, Paul Bello, Nick Cassimatis, Ken Forbus, Ben Goertzel, John Laird, Christian Lebiere, Pat Langley, Robert Mariner, Stacey Marsella, Matthias Scheultz, Satinder Singh, and Robert Wray. The workshop was funded under grant N00014-08-1-1214 from ONR.

References

Anderson, J. R. & Lebiere, C. L. 2003. The Newell test for a theory of cognition. Behavioral & Brain Science 26, 587-637.

Cohen, P. R., 1995. *Empirical methods for artificial intelligence,* Cambridge, MA: MIT Press.

Cohen. P.R. 2005. If not Turing's test, then what? *AI Magazine*, Winter, 26; 61-68.

Doorenbos, R. B. 1994. Combining left and right unlinking for matching a large number of learned rules. *In Proceedings of the Twelfth National Conference on Artificial Intelligence*.

Gluck, K. A. & Pew, R.W. 2005. Modeling human behavior with integrated cognitive architectures: comparison, evaluation, and validation LEA/Routledge.

Jones, R. M., & Wray, R. E. 2006. Comparative analysis of frameworks for knowledge-intensive intelligent agents. AI Magazine, 27, 57-70.

Laird, J. E., 2008. Extending the Soar cognitive architecture. In *Proceedings of the First Conference on Artificial General Intelligence*.

Langley, P., & Choi, D., 2006. A unified cognitive architecture for physical agents**.** In *Proceedings of the Twenty-First National Conference on Artificial Intelligence.* Boston: AAAI Press.

Langley, P., Laird, J. E., & Rogers, S., in press. Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*.

Langley, P., & Messina, E. 2004. Experimental studies of integrated cognitive systems. Proceedings of the Performance Metrics for Intelligent Systems Workshop. Gaithersburg, MD.

Legg , S. & Hutter, M. 2007. Universal Intelligence: A Definition of Machine Intelligence, *Minds and Machines,* 17:4, 391-444.

Newell, A., Simon, H. A., 1976. Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, **19**

Turing, A., 1950. Computing Machinery and Intelligence. *Mind,* 59; 433–460.

Wray, R. E., & Laird, J. E. 2003. An architectural approach to consistency in hierarchical execution. *Journal of Artificial Intelligence Research*. 19; 355-398.

Wray, R. E., & Lebiere, C. 2007. Metrics for Cognitive Architecture Evaluation. In Proceedings of the AAAI-07 Workshop on Evaluating Architectures for Intelligence, Vancouver, B. C.