A Gauntlet for Evaluating Cognitive Architectures

Marc Pickett I and **Don Miner** and **Tim Oates**
Cognition, Robotics, and Learning
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250

Abstract

We present a set of phenomena that can be used for evaluating cognitive architectures that aim at being designs for intelligent systems. To date, we know of few architectures that address more than a handful of these phenomena, and none that are able to explain all of them. Thus, these phenomena test the generality of a system and can be used to point out weaknesses in an architecture's design. The phenomena encourage autonomous learning, development of representations, and domain independence, which we argue are critical for a solution to the AI problem.

Introduction

Cognitive architectures can have different goals for different research communities. Psychologists and neuroscientists would like a cognitive architecture to help explain mental phenomena in people. This is the stated goal of ACT-R (Anderson 1993), and is also a driving force of the desiderata given by (Sun 2004). Pragmatists might prefer an architecture that would be useful as a tool, such as a medical advisor. However, researchers who are interested in the emergence of intelligent machines should be most excited about cognitive architectures that describe a fully intelligent system. These 3 goals often overlap, but there are differences. For example, a tool for doctors might contain an extensive, hand-designed, medical ontology, but without the ability to learn, such a system would hardly be considered a design for a fully intelligent agent. Another example where the goals diverge would be a system that (in addition to forming concepts) memorizes its raw input, which is possible with today's affordable massive storage devices. Such a "superhuman" memory would be undesirable for a psychologist, since people clearly forget things, but this quality isn't necessarily a problem for an AI researcher, who might even prefer this model. We're interested in a solution to the AI problem, and therefore our goals for a cognitive architecture don't necessarily include the making of immediately useful tools or the explanation of human cognitive phenomena (though such properties may result from pursuing our primary goal).

In this paper, we suggest a "gauntlet" of criteria for what an agent should be able to do. The gauntlet is non-exhaustive, but we know of no system that meets all the criteria, and such a system would be an advancement for AI. The gauntlet can be used to evaluate a cognitive architecture: the more fully a cognitive architecture addresses the items in the gauntlet, the better.

Background: Core Goals of Intelligence

Autonomous development of representations should be a primary goal of a cognitive architecture. Traditional approaches to AI focus on selecting an application and then constructing representations for that domain. These approaches are problematic in that they require much labor intensive knowledge engineering. Furthermore, these systems tend to be brittle, often failing when they encounter unanticipated situations. An alternate approach is to have the computer develop its representations autonomously. In this alternate approach, the robot is viewed as a "robot baby" (Cohen *et al.* 2002). The robot is provided a minimal amount of knowledge (implicit or otherwise) about the world and is expected to learn and develop a conceptual structure from large amounts of raw sensor data over a long period of time. This approach is attractive because it requires little knowledge engineering and is robust because the agent learns to adapt to unanticipated situations. This approach also directly addresses the Symbol Grounding Problem (Harnad 1990) –the problem of creating meaning using only a set of meaningless symbols– by directly grounding all an agent's knowledge in sensory data.

Since we provide a minimal amount of domain knowledge, domain independence and generality should be among the top criteria in evaluating an agent architecture. Therefore, an empirical demonstration of an agent architecture should contain several disparate (though data-rich) domains with a minimal amount of human-provided data "massaging". A set of domains might contain robot sonar sensor data, a large corpus of text, a series of images, and a simulation of Conway's Game of Life. For each of these, an architecture should, at a minimum, autonomously develop an ontology that's useful for characterizing that domain. For example, when given sonar data, the agent may build a hierarchy of motifs. When given images, the agent should develop edge filters, and when given Conway's Game of Life, the agent should develop the concept of a "glider".

To answer the question more precisely of exactly *what*

an intelligent agent should do with its data is perhaps tantamount to answering the question of what intelligence is. It has been suggested that a core purpose of intelligence is to concisely characterize a set of data (Wolff 2003), (Hutter 2004). That is, given data, an intelligent agent should generate a model that best compresses the data. This is the principle of Minimum Description Length (MDL). It is fundamentally equivalent to Ockham's Razor, which says, in effect, that "The shortest model (that predicts the data) is the best model.". If we assume that the prior probability of a model is inversely proportional to the exponent of its description length, then Ockham's Razor is also fundamentally equivalent to the Bayesian principle that states that "The most probable model is the best model.".

We somewhat agree with these claims. An intelligent agent should be able to build a model that concisely characterizes its sensor data, and it should be able to use this model to answer queries about the data. Such queries might consist of making accurate predictions about given situations. The agent should also be able to generate plans to accomplish goals (or obtain reward). However, the time needed (in terms of steps of computation) to answer these queries should also be taken into account. Thus, it is sometimes useful to occasionally trade memory for time. For example, an intelligent being might cache a result that it has deduced if it expects to use the result again.

To make this concrete, suppose our agent's domain is Euclidean Geometry. In this domain, a huge but finite set of theorems of can be "compressed" down to a model containing just 5 postulates and some rules for inference. Such a model would neither be very useful nor would it work the same way as a person. A professional (human) geometer would likely "cache" useful lemmas thereby speeding up his or her future deductions. It seems true that the same should apply to a generally intelligent being. Another example involves sensor data. If we equip our agent with a video camera, it's possible that the most concise representation of the data (if the pictures are fairly continuous) will be an encoding typical of many video compression algorithms. That is, the representation might fully describe the initial frame, then describe each subsequent frame as changes from its previous frame. A problem with this approach is that it would take longer to answer queries about the end of the day than the beginning (because the entire day would have to be "unwrapped"). This also seems contrary to our intuitions about what an intelligent agent should be able to do.

Thus, we propose an alternative to Ockham's Razor called Marctar's Axe, which states "The quickest model (that predicts the data) is the best model.". By quickest, we mean the model that takes the fewest steps of computation to get accurate answers to queries. Of course, there's a tradeoff between speed and accuracy, but this can be folded into a single number by setting a parameter that would act as an "exchange rate" between steps of computation and bits of accuracy. Marctar's Axe somewhat overlaps with Ockham's Razor in that fast models tend to be small and tidy so that computation isn't spent searching through disorganized sets of information. Marctar's Axe also addresses the utility of caching: caching the answers to frequent queries (or frequent "way points" in derivations) can yield a faster model.

In the next section we present the gauntlet, which is a set of phenomena that a cognitive architecture should be able to explain or produce. We view the end goal of AI in terms of Marctar's Axe. That is, to obtain quick and accurate answers or predictions about a set of data. The items in the gauntlet can be viewed as subgoals of Marctar's Axe.

The Gauntlet: Desirable Cognitive Phenomena

We consider the following list of cognitive phenomena to be necessary (but not necessarily sufficient) features of general intelligence. A cognitive architecture that explains general intelligence should have a story for how it addresses them. This list is incomplete, but many cognitive phenomena not on the list are corollaries of those on the list. For example, a full solution to the problem of representing, creating, and using invariant representations could readily be used to solve the Frame Problem (McCarthy & Hayes 1969), which is the problem of stating what remains unchanged when an event occurs.

The items in the list aren't necessarily independent. That is, some of the items might be corollaries of other items in the list. Therefore, these phenomena can either be directly addressed, or some may be solved as emergent properties of an architecture.

Concept Formation As we mentioned in the Background section, an agent should be able to develop its own representations of the world. These representations should, at some level, form a concept ontology, which should be arranged in a semantic *heterarchy*. For example, the concept that corresponds to a pterodactyl should belong to both the class of flying things, and the class of reptiles. The concept formation mechanism should be able to make concepts out of virtually anything, not only physical objects. There should be concepts that characterize relations, events, stories, actions, and even cognitive actions.

Invariant Representations and Analogy When a person dons a pair of green-tinted sunglasses for the first time, they have little trouble adapting to their altered visual input, but this isn't such a trivial task for a (visual) robot. In terms of raw sensor data, a green-tinted scene has very different values from the same scene in its natural color. We suspect that this is because people have abstract representations that are invariant of the instances that caused them. Representations developed from visual data should also be invariant to translation, rotation, and scaling. These *invariant representations* aren't limited to visual data. A stenographer can hear different speakers say the same phrase in different pitches, volumes, and speeds, yet produce the same transcription.

A particularly tricky problem is to discover a representation of a traffic wave. That is, given a bird's eye view of a simulation of automobile traffic on a highway, a person can readily point out where the traffic jams are. A traffic jam is different from a collection of cars. Individual cars move in and out of a traffic jam, and the jam itself usually moves

opposite the direction of traffic. A person could also create features to describe the traffic waves: e.g., its spread and how fast it's moving.

An important class of invariant representations consists of those formed through *analogy*. Some suggest that analogy may even be the "core of cognition" (Hofstadter 2001). Analogy allows us to focus on the relations among entities rather than superficial aspects of the entities. For example, we might notice that a red ant killing a black ant and stealing a piece of food it is analogous to a situation in Hamlet where Claudius murders Hamlet's father and usurps the throne of Denmark. In this situation *binding* is important. That is, we must be able to specify that the red ant corresponds to Claudius, the black ant to Hamlet's father, and the piece of food maps to the throne. Analogy is also useful for *knowledge transfer*: if an analogy is found, then conclusions about one domain can map to another domain.

Plato's Cave: Theory Building In his Allegory of the Cave (Plato 360 BC), Plato describes a group of people whose observations of the world are solely shadows that they see on the wall of a cave. The question may arise as to whether these observations are enough to propose a theory of 3-dimensional objects. In principle, this problem can be solved. If an agent is given a representational framework that's expressive enough to encode a theory of 3-dimensional objects, then the agent could go through the combinatorially huge number of theories expressed in this language (under a certain length) and choose the one that best explained the data (where "best" can be defined in terms of Ockham's Razor or Marctar's Axe). The best theory will likely include a description of 3-dimensional objects (assuming such a theory is of unrivaled utility for characterizing the data). Thus, our task is possible, given an exponential amount of time. There are other real examples of building theories of phenomena that aren't directly observable: neither atoms, genes, radio waves, black holes, nor multimillion year evolutionary processes are directly observable, yet scientists have built theories of these.

A robot given sonar sensor data (or uninformed visual data) is faced with fundamentally the same problem. Visual observations of a 3-dimensional object, such as a pen, can be very different (in terms of raw sensor data) depending on whether the pen is viewed lengthwise or head on. Therefore, the ability to propose "scientific theories" of this type is something a cognitive architecture should be able to explain. The architecture's representation framework needs to be expressive enough to encode such theories, and the architecture's model-builder should be able to discover theories in polynomial time.

A similar approach can be used to create causal theories from observational data (Pearl 2000). Statisticians point out that it's impossible to prove causality from observational data. This is true, but we can use our "theory language" to propose causal theories that are more likely than others.

Reasoning, Parsing, and Planning Clearly, the ability to reason is an essential component of intelligence. Reasoning requires rules of inference, and the ability to *learn* these rules should be another requirement of intelligence. That is, an agent shouldn't rely on being told rules such as "If it is raining, and a robot goes outside, then that robot will get wet.".

An agent should be capable of *hypothetical reasoning*: an agent should be able to represent counterfactual situations, and deduce consequences of these situations. An agent might also pay special attention to cases where many different hypotheses yield the same conclusion (and thereby develop a general rule). For example, a robot might "imagine" several scenarios in which it falls from great heights, each simulation resulting in the conclusion that the robot would be damaged. The robot should then generalize that falling from heights causes damage. An essential component of hypothetical reasoning is being able to represent that an event is merely make-believe. Some otherwise promising systems, such as that described by (Hawkins & Blakeslee 2004) seem to lack this ability. Furthermore, reasoning should be able to continue for more than a few steps, even under uncertainty (in which case an architecture should be able to investigate different scenario branches).

An agent should be able to explain its world in terms of its learned conceptual structure. This should be done by parsing, or classifying data according to its current concepts, and by using rules of inference that it has developed. An architecture should be able to escape local optima in its characterization of the world. For example, an agent should be able to reclassify data, or replace one explanation by a shorter one. An agent should also be able to remove obsolete or unused concepts from its ontology.

An agent should also be able to use reasoning (and especially hypothetical reasoning) to develop plans for accomplishing goals. We suspect that reasoning, parsing, prediction, classification and explanation in terms of developed concepts, and planning can be implemented as different facets of a common algorithm. For example, planning might simply be the process of "explaining" how a desired situation can come about.

Finally, an agent should be able to combine its abilities to reason, learn rules of inference, and form heterarchical conceptual structures to implement hierarchical reasoning. That is, an agent should be able to create a conceptual structure of rules, and use these rules to quickly reach conclusions. For example, an agent faced with the task of breaking a piece of wood might follow this path: (going up the heterarchy) "Wood is-a rigid object. Breaking is-a change. To change a rigid object requires force applied to it." (and back down) "There are several ways to apply force. Some involve striking with another rigid object. A rock is another rigid object...".

Metacognition It would be useful for an agent to be able to observe and modify aspects of its own cognitive behavior. Such metacognition might allow the agent to cache "lemmas" (and other conclusions) or develop heuristics to speed searches. Metacognition, because it deals with information about information, is useful for deciding which ac-

tions and cognitive actions an agent can take to improve its model. Therefore, metacognition can be used to pose questions and design experiments (action plans) to gain information. Metacognition can also be used to search for inconsistencies in the model, and modify the conceptual structure when contradictions are found.

It's possible for metacognition to be elegant: If the representation schema for an architecture is general enough, then it should be able to encode cognitive actions. If the model-building mechanism of the architecture is powerful enough to blindly work on any system described in its representation framework, then it's conceivable that an agent can characterize its own cognitive actions just as it would characterize any other stream of data (assuming we separate cognitive actions from metacognitive actions, thus avoiding a feedback loop).

Specifying Reward An agent should be able to plan and take actions to attain a (possibly externally specified) goal, and an architecture will have to address *how* goals are specified. This is non-trivial for a robot baby because it's born with a minimal model of the world and therefore, no "language" to express what should cause a reward. Furthermore, we shouldn't rely on any particular sensor modality to specify the reward. That is, the specification of the reward should be invariant to the raw sensor data.

Human brains seem to have solved this problem. For example, the majority of male humans seem to be innately attracted to women, and vice versa. From a computational standpoint, telling the difference between a man and a women is far from trivial. The attraction seems to be invariant to any single modality: most people either blind or deaf from birth still follow this pattern.

An approach to solving the problem is given in the following example: Suppose we wanted a robot's innate goal to be to harvest tomatoes. Furthermore, we wanted the robot to harvest only proper tomatoes (ripe, but not over-ripe, not too small, or insect infested, etc.), and we don't know what its sensor suite will be. (In practice, we might "teach" the robot as we would with a human, but this example is for illustrative purposes.)

To do this, we could build several robots with widely ranging modalities, and have them (over several months) experience the environment of the tomato fields. From this experience, the robots should have built a world model that would include a "tomato taxonomy". Then, we can find an invariant representation of a "good tomato" by noting what parts of their ontologies are "active" when they have experiences with good and bad tomatoes (analogous to what some neuroscientists do to localize various cognitive functions in humans in fMRI studies). Specifically, we could find an invariant representation for the act of harvesting good tomatoes. We can then put this in an agent's "innate" model, and specify that it's a goal. Then, when developing representations, the agent should discover a representation that is similar (or perhaps isomorphic to) the representation of this goal.

Statistical and Symbolic Components (Sun 2004) argues that a cognitive architecture that explains people should contain "an essential dichotomy" that can be roughly paraphrased as a split between traditionally statistical and symbolic methods. Sun's arguments are from a psychological perspective and might not apply to general intelligence. That is, we leave open the possibility that there might be a unified system that captures both the symbolic and statistical elements of cognition. Whether an architecture explicitly has this dichotomy or not, it seems clear that an intelligent agent should have the strengths of both components.

The world is too complex to be modeled completely, and therefore a system must be able to characterize and handle uncertainty. Furthermore, an agent should be able to pick up on subtle statistical patterns such as correlations among events. Statistical methods are useful for this, but standard methods, such as support vector machines or connectionist models, aren't without their downsides. For example, any connectionist system must address the Binding Problem (von der Malsburg 1999), the problem of specifying which concepts are bound to which parameters, which is important for making use of analogies. Solving The Binding Problem is trivial in symbolic representations. A statement as simple as "`bind <symbol1> <symbol2>`" might suffice.

Along similar lines, an intelligent system should also be able to create new grammatical constructions. For example, given the concept of "red" and the concept of "ant", an agent should be able to represent a "red ant". There should be no hard limit to the depth of allowable constructions. An agent should also be able to represent a (new) concept as complex as "raining rocking horses on Pluto's 3rd moon".

Nonessential Human Cognitive Phenomena

Our goal is not to model humans. As the only exemplar of intelligence, research on how human brains work often lends insight as to how an intelligent system can be built (Hawkins & Blakeslee 2004), but we believe that there exists a more direct way to create intelligence than exactly modeling human brains, and therefore we shouldn't be constrained to designs that were myopically stumbled upon by evolution. An analogy might be made to a human eye's lens. Its shape is an approximation of an ideal lens that can be described by an elegant parabolic equation. If a design for intelligence happens to account for certain human cognitive phenomena, we'll be glad, but this goal is secondary.

In fact, some human cognitive phenomena (such as psychotic behaviour or getting tired) are undesirable. There are some cognitive phenomena that occur in humans that we've intentionally omitted from the gauntlet. For example, we've omitted any mention of consciousness. Whether it's possible to have intelligence without consciousness (in the sense of sentience) is open to debate. Regardless, our chief aim is intelligence, and if it's impossible to have intelligence without sentience, then sentience should be a byproduct of our architecture, not a direct goal. Another example is the split between short-term and long-term memory. It's not clear to us that this split is necessary for intelligence. A cognitive architecture should have both concept definitions and statements, but an elegant model might fold these together (so

that it can use the same mechanisms to work with both). It's possible that the long-term/short-term split is only a product of physical mechanisms in brains (Lynch 2004).

Outlook

We feel that it's premature to have a numeric benchmark for a cognitive architecture, since no major architecture is able to accomplish all the gauntlet's tasks at any speed. At this stage, a simple architecture that's unoptimized, yet able to address most of the gauntlet is preferable to a complex architecture that's optimized but able to address just a few items. This is for the same reason that, were we studying aviation in the late 19th century, a Kitty Hawk style flyer would have been preferable to a system containing only the landing gear and navigation system of an F-15. That is, an architecture should be a full design for intelligence. Case studies in various aspects of intelligence may be useful, but an architecture should focus more on the overall design and interactions of these components.

Therefore, elegance of an architecture is important. An architecture should strive to be simultaneously both simple and general. One strategy to accomplish this is to put the bulk of complexity in the *process* of the architecture as opposed to the architecture's *design*. That is, a simple architecture that takes more computational resources to learn and reason should be preferable to a complex architecture that's a constant factor more efficient. For example, an architecture describable in 500 lines should be preferable to an architecture that takes 20,000 lines to describe but runs 10% faster. Optimization should come after generality.

References

Anderson, J. R. 1993. *Rules of the Mind*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Cohen, P.; Oates, T.; Beal, C.; and Adams, N. 2002. Contentful mental states for robot baby. In *Proceedings of the 18th National Conference on Artificial Intelligence*.

Harnad, S. 1990. The symbol grounding problem. *Physica D* 42:335–346.

Hawkins, J., and Blakeslee, S. 2004. *On Intelligence*. Times Books.

Hofstadter, D. R. 2001. Analogy as the core of cognition. *The Analogical Mind: Perspectives from Cognitive Science* 499–538.

Hutter, M. 2004. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Berlin: Springer.

Lynch, M. 2004. Long-term potentiation and memory. *Physiol. Rev.* 84:87–136.

McCarthy, J., and Hayes, P. J. 1969. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence* 4:463–502.

Pearl, J. 2000. *Causality*. Cambridge University Press.

Plato. 360 BC. *The Republic, Book VII*.

Sun, R. 2004. Desiderata for cognitive architectures. *Philosophical Psychology* 17(3):341–373.

von der Malsburg, C. 1999. The what and why of binding: The modeler's perspective. *Neuron* 24:95–104.

Wolff, J. G. 2003. Information compression by multiple alignment, unification and search as a unifying principle in computing and cognition. *Artif. Intell. Rev.* 19(3):193–230.