The Toy Box Problem (and a Preliminary Solution)

Benjamin Johnston[1]
Business Information Systems
Faculty of Economics and Business
The University of Sydney
NSW 2006, Australia

Abstract

The evaluation of incremental progress towards 'Strong AI' or 'AGI' remains a challenging open problem. In this paper, we draw inspiration from benchmarks used in artificial commonsense reasoning to propose a new benchmark problem—the Toy Box Problem—that tests the practical real-world intelligence and learning capabilities of an agent. An important aspect of a benchmark is that it is realistic and plausibly achievable; as such, we outline a preliminary solution based on the Comirit Framework.

Introduction

The objective of commonsense reasoning is to give software and robotic systems the broad every-day knowledge and know-how that comes effortlessly to human beings and that is essential for survival in our complex environment. While commonsense is, or at least appears to be, a narrower problem than 'Strong AI' or 'AGI', it shares many of the same representational, computational and philosophical challenges. Indeed, one might view commonsense as the practical, immediate and situated subset of general purpose intelligence.

Thus, commonsense reasoning serves as a useful stepping-stone towards both the *theory* and *practice* of Strong AI. Not only does commonsense provide a broad, deep and accessible domain for developing *theoretical* conceptions of real-world reasoning, but the *practical* experience of developing large scale commonsense provides useful insights into the scalability challenges of general-purpose intelligence.

In much the same way that evaluation challenges AGI researchers today[2], work in commonsense reasoning has long been challenged by the difficulty of finding manageable, but open-ended benchmarks.

The difficulty of evaluation facing both commonsense reasoning and AGI arises from the fact that the problems are so great that we cannot today simply implement *complete* systems and test them in their intended domains; it is necessary to measure *incremental* progress. Unfortunately, incremental progress can be difficult to judge: it is relatively easy to demonstrate that a particular formalism supports or lacks a given feature, but it is much harder to determine whether that feature represents a meaningful improvement. For example, even though a formalism or system appears to offer improvements in expressive power, efficiency and ease-of-use, it may have sacrificed a 'show-stopping' crucial feature that is overlooked in the evaluation.

In the commonsense reasoning community, this problem is addressed by defining non-trivial (but plausibly achievable) reasoning problems, and then analyzing the ability of a formalism to solve the problem and a number of elaborations. While analysis is conducted with respect to the system's performance (rather than 'features'), the formalism itself is also considered in what might be termed a 'grey box analysis' (rather than, say, a black-box comparison such as used at RoboCup). To these ends, Morgenstern and Miller (2009) have collected a set of non-trivial commonsense reasoning problems and a handful of proposed 'solutions'.

With any benchmark or challenge problem there is, of course, the temptation to fine-tune a system to the problem, rather than attempting to design more general and abstract capabilities. As such, the benchmarks used in the commonsense reasoning community are not formally defined, nor are they strictly measurable. Instead, they offer a shared and open-ended scenario to guide qualitative analysis of progress towards our goals. Even though this approach is less objective than a competition or a formal goal, these challenge problems provide a meaningful context for evaluation that helps temper unfounded wild claims, while at the same time avoiding specifics that are readily gamed.

To date, we have been developing a general purpose commonsense-reasoning framework named *Comirit*, and have evaluated the system on two benchmark problems: the Egg Cracking Problem (Johnston and Williams 2007), and the Eating on an Aircraft Problem (Johnston 2009). We have found this methodology useful, but in charting a course to more general intelligence, we found that the current selection of proposed benchmark problems offer little scope for evaluating the ability of an agent to learn on its own and thus demonstrate AGI-like capabilities.

Our objective in this paper is therefore to present an open-ended benchmark, the *Toy Box Problem*, in the style of Morgenstern and Miller (2009), which may be used to evaluate progress towards commonsense reasoning and general intelligence.

In order to briefly illustrate the feasibility of the benchmark problem and the problem may be applied, we then use the *Toy Box Problem* on the Comirit framework. In doing so, we will introduce new capabilities in the framework, and show how the framework may be used to partially solve the Toy Box Problem.

[1] This research supported in part by an ARC Discovery Grant while at the University of Technology, Sydney.

[2] Consider, for example, the 'Developing an AGI IQ Test' workshop that is affiliated with the AGI 2010 conference.

Published by Atlantis Press, © the authors

The Toy Box Problem

As with existing benchmarks used within the commonsense reasoning community (Morgenstern and Miller 2009), we pose the *Toy Box Problem* problem as a hypothetical scenario:

> A robot is given a box of previously unseen toys. The toys vary in shape, appearance and construction materials. Some toys may be entirely unique, some toys may be identical, and yet other toys may share certain characteristics (such as shape or construction materials). The robot has an opportunity to first play and experiment with the toys, but is subsequently tested on its knowledge of the toys. It must predict the responses of new interactions with toys, and the likely behavior of previously unseen toys made from similar materials or of similar shape or appearance. Furthermore, should the toy box be emptied onto the floor, it must also be able to generate an appropriate sequence of actions to return the toys to the box without causing damage to any toys (or itself).

The problem is intentionally phrased as a somewhat constrained (but non-trivial) scenario with open-ended possibilities for increasing (or decreasing) its complexity. In particular, we allow the problem to be instantiated in combinations of four steps of increasing situation complexity and four steps of toy complexity.

That is, the problem may be considered in terms of one of the following environments:

E1. A virtual robot interacting within a virtual 2-dimensional world

E2. A real robot interacting within a real-world planar environment (*e.g.*, a table surface with 'flat' toys and in which no relevant behavior occurs above the table surface)

E3. A virtual robot interacting within a virtual 3-dimensional world

E4. A real robot interacting within the real world, without constraints

Similarly, the complexity of toys is themselves also chosen from the following:

T1. Toys with observable simple structure, formed from rigid solids, soft solids, liquids and gases

T2. Toys with complex, but observable mechanical structure (again, created from rigid solids, soft solids, liquids and gases)

T3. Toys with complex, but observable mechanical structure, created from any material (including magnets, gases and reactive chemicals)

T4. Toys with arbitrary structure and operation (including electronic devices)

In each case, the world and the toys contained within, may only be observed via 'raster' cameras. That is, even in virtual worlds (E1 and E3), the robot is unable to directly sense the type or the underlying model of a virtual toy (*i.e.*, the problem cannot be tested in a world such as Second Life, in which agents can directly 'sense' the symbolic name, type and properties of an object).

In fact, virtual worlds (of E1 and E3) should be as close as possible to a physical world (including the ability for objects to be arbitrarily broken and joined). Virtual worlds are included in the Toy Box Problem not to reduce the conceptual challenge of the problem, but primarily to separate the effort involved in dealing with camera noise, camera/hardware failures, color blurring and other sensory uncertainty.

The Toy Box Problem may be used for evaluating an 'intelligent' system by selecting a combination of environment and toy challenges. For example, the pairing E1&T1 represents the easiest challenge, whereas E4&T4 present the greatest difficulty. Note, however that the pairs do not need to match: the next development step for a system which solves E1&T1 might be either E1&T2 or E2&T1.

The Toy Box Problem is specifically designed as a *stepping stone* towards general intelligence. As such, a solution to the simplest instances of this problem should not require universal or human-like intelligence. While an agent must have an ability to learn or identify by observation (because the toys are new to the agent), it does not *necessarily* require the ability to 'learn to learn'. For example, given a comprehensive innate knowledge-base of naïve physics, it *may* be sufficient for an agent to solve the problem with toys in T1 and T2 by a process of identification rather than true learning. However, the difficulty of the challenge increases with more complex toys of T3 and T4, and it is unlikely that a system would continue to succeed on these challenges without deeper learning capabilities (though, it would be a very interesting outcome with deep implications for AGI research if a system without learning capabilities does continue to succeed even on the most challenging instances of the problem).

While the pairing E1&T1 is the easiest challenge of the Toy Box Problem, we believe that any solution to E1&T1 would be a non-trivial accomplishment, far beyond the reach of standard 'Narrow AI' techniques in use today. Nevertheless, we expect that the pair E1&T1 should lie within reasonable *expectations* of the capabilities of proposals for 'Strong AI' architectures today. One could readily conceive of systems based on methods as diverse as logic, connectionist networks or genetic programming to each be adapted to solving E1&T1 within a short-term project, and thus form the basis of meaningful comparison and analysis between disparate methods.

More difficult combinations, such as the pair E4&T4, are currently far beyond all current technologies. While a system that performs well for such pairings may not have true general intelligence, it would be at the pinnacle of practical real-world physical competence and would have serious real world applications. For example, this level of knowledge would enable a domestic robot or a rescue robot to deal with the many unexpected objects and situations it would encounter during its routine duties: whether cleaning a house or making way through urban rubble.

Finally, it is worthwhile noting a connection here with recent discussions concerning the creation of virtual 'preschools' for human-like AI development (Goertzel and Bugaj 2009). The Toy Box Problem may be viewed as a specific and achievable 'target' for developing and evaluating real systems, rather than simply aiming to provide an enriching environment for robot 'education'.

In this rest of this paper, we further illustrate the problem by outlining a preliminary solution to the Toy Box Problem, and considering (in the first instance) how it may be 'solved' for the pair E1&T1.

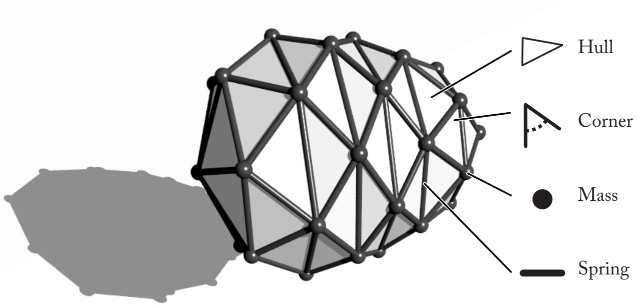

Figure 1: Parts of a simulation of an egg

Comirit Framework

Over the past four years, we have been developing Comirit; a hybrid reasoning framework for commonsense reasoning. At the core of the framework lies a generic graph-based scheme for constructing simulations that implicitly capture practical commonsense knowledge. While the framework is not intended to be biologically plausible, simulation in Comirit may be viewed as a computational analog to human visual imagination. Comirit uses simulations to reason about, predict and speculate about a given situation, by first instantiating a simulation of that situation and then using the simulation as a mental playground for experimenting with possible actions and anticipating reactions.

However, while simulation is a powerful, computationally efficient, and easy-to-engineer scheme for representing commonsense knowledge and predicting the outcome of a situation, the method is constrained to concrete reasoning and to the 'arrow-of-time'. That is, simulation by itself is not well suited to the following kinds of reasoning:

1. Explaining the cause of an outcome ('Why is there split milk on the floor?')
2. Fact-based reasoning ('What is the capital of Russia?')
3. Abstract deduction ('What is 3 plus 7?')
4. Learning about and predicting in novel domains ('How will this new toy behave?')

We have therefore developed Comirit as an open-ended multi-representational framework that combines simulation with logical deduction, machine learning and action selection. This integration is achieved by a uniform mechanism that is based on the automated theorem proving method of analytic tableaux (see *e.g.*, Hähnle 2001). In Comirit, the tableau algorithm is extended so that it searches and ranks spaces of possible worlds, enabling the disparate mechanisms to be uniformly represented and reasoned in a unified tableau.

In the remainder of this section, we provide an overview of simulation, hybrid reasoning and learning in Comirit and show how it relates to the *Toy Box Problem*. More detailed explanations of Comirit may be found in our prior publications (Johnston and Williams 2007; 2008; 2009).

Simulation

In the Comirit framework, simulations are the primary representation of commonsense knowledge. Comirit Simulations are designed as a generalization and formalization of an early proposal by Gardin and Meltzer (1989). In particular, Comirit uses a generic graph-based representation that has been extended to use accurate 3D physics[3].

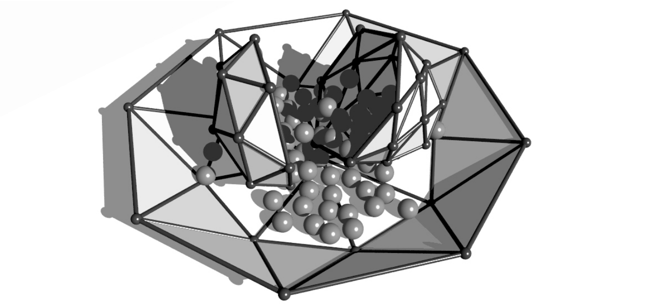

Figure 2: Simulation of an egg cracked into a bowl (the beads represent spilled yolk and white)

Complex objects are modeled by approximating the structure of the object as an annotated graphical structure, and then iteratively updating the annotations according to the laws of physics. That is, if we have an object to simulate—an egg, for example—then a graph is instantiated comprising of vertices that denote interconnected 'masses', 'springs', 'torsion springs' and 'convex hulls' which approximate the structure of an egg. Each such vertex is annotated with attributes to drive appropriate behavior; for example, each 'mass' vertex has a spatial position (*x*, *y*, *z* coordinates) and a local mass density (among other attributes). Newton's laws of motion are then iteratively applied to the graph structure. For example, the effects of springs, liquids and hulls are modeled using Hooke's law. Figure 1 illustrates some of the parts of a simulation for the Egg Cracking Problem. Figure 2 also depicts the result of running such a simulation to observe the effects of dropping an egg into a bowl: the egg has cracked, and its liquid contents have spilled out.

Comirit uses 3D simulations and so is potentially applicable to E3 and E4 in the Toy Box Problem, however the same framework may be trivially adapted to the 2D environments of E1 and E2. The translation of 3D physics into 2D physics is straightforward: one axis is either fixed to be constant, or is entirely removed from the equations of motion.

A system may perform powerful reasoning about any object for which it has a simulation. For example, it may consider safe ways of handling eggs and toys by instantiating a simulation in internal memory and then testing actions against that 'imagined' instance. If the agent uses visualization to determine that an heavy force will cause an egg to crack, it can avoid causing damage to the egg in real life.

Simulation + Reasoning

Recalling that simulation only supports a 'forward chaining' inference mode, we have integrated simulation with logical deduction in a hybrid architecture in order to combine the strengths and complement the weaknesses of each mechanism. That is, we use the deductive power of a general-purpose logic to make up for the inflexibility of simulation.

In combining simulation and logic, our experiences are that

[3] Comirit may also support non-physical domains, such as financial markets or organizational behavior but they are beyond the scope of this paper.

the conceptual mismatch between the mechanisms of simulation and logic prevents the application of traditional integration techniques such as blackboard architectures. Our attempts to use mainstream integration architectures invariably resulted in systems that were unworkably complex and difficult to maintain. Instead, we sought a clear and unifying abstraction to harmonize the semantics of the reasoning mechanisms; by interpreting both simulation and logical deduction as operations that manipulate spaces of possible worlds.

The method of analytic tableaux (see *e.g.*, Hähnle 2001) is an efficient method of automatic theorem proving. Analytic tableaux have been successfully applied to large problems on the semantic web, and there is a vast body of literature on their efficient implementation (*ibid.*). The method involves the automatic construction of search trees (tableaux) through the syntactic decomposition of logical expressions, and then eliminates branches of the tree that contain contradictions among decomposed atomic formulae. Each branch of the resultant tableau may be seen as a partial, disjunction-free description of a model for the input formulae.

Logical deduction *and* simulation can be unified through tableau reasoning. The tableau algorithm is designed for logical deduction, and its algorithm is effectively a search through symbolically-defined spaces of worlds. Simulation is a process that can be used to expand upon symbolic knowledge in a given world (*i.e.*, by forward chaining to future states based on description of the current state), and so simulation can be applied to generate information in the branches of a tableau.

Comirit thereby incorporates a generalization of the tableau method such that a tableau may contain not only standard logical terms and formulas, but also non-logical structures such as simulations, functions, data-structures and arbitrary computer code. With some similarity to the methods of PolyScheme (Cassimatis 2005), integration in Comirit is achieved by translating diverse reasoning mechanisms into the tableau operators for expansion, branching and closing of branches. Traditional logical tableau rules are used unchanged, and simulation is treated as an expansion operator (like the conjunction rule).

More detailed explanation of the workings of Comirit tableau reasoning (including an explanation of how tableau rules, heuristics and meta-rules are also recursively embedded inside the tableau) may be found in our earlier publication (Johnston and Williams 2008). In the following subsection, we will provide an example of a tableau as it is further extended and used for machine learning.

Simulation + Reasoning + Learning + Action

Of course, even with a comprehensive knowledge base, an intelligent system will be of limited use in any complex and changing environment if it is unable to learn and adapt. Indeed, in the Toy Box Problem, the agent has no prior knowledge of the specific toys that it may encounter. The system must autonomously acquire knowledge through interaction and observation of the toys.

It turns out that simulation is ideal for observation-based learning. The laws of physics are generally constant and universal; an agent does not need to learn the underlying laws of behavior of every object. Thus, when the underlying graphical structure of an object can be approximated by direct observation, the learning problem is then reduced to discovering the hidden *parameters* of the object by machine learning.

For example, given a novel toy (e.g., a toy ring), direct observation may be used to directly instantiate a graph-based mesh that approximates the structure. In the 3D case, this would be achieved by using the 3D models extracted from stereographic cameras, laser scanners or time-of-flight cameras; in the 2D case, this might be achieved by simple image segmentation.

Once the shape of an object has been approximated by a graph, machine learning is used to determine underlying values of annotations: mass densities, spring constants, rigidity and breaking points that will result in an accurate simulation. These can be discovered simply by collecting observations, and using these observations as training instances for a parameter search algorithm (where fitness is measured by the accuracy of the simulation given the parameters).

However, while simulation is well suited to learning, it is not necessarily obvious how to reconcile the search for consistency that is fundamental to the tableau method with the hypotheses search and evaluation of learning. A given hypothesis can not be independently declared either true or false (as demanded by tableau reasoning): it is only possible to compare hypotheses against each other and select the 'best'.

Thus, in Comirit, learning is implemented by further extending the tableau reasoning algorithm. Learning in Comirit is treated as a 'generate-and-test' algorithm. Generation of candidate hypotheses is akin to the disjunctive branching of the tableau, however the testing of hypotheses is implemented as a special extension of the tableau algorithm to allow branches to be closed due to sub-optimality.

Learning is therefore implemented by introducing an ordering over branches, and then treating the tableau algorithm as a search for both consistent *and* minimal models. A branch in a tableau is no longer advanced or refined (as though 'open') simply if it is consistent per the traditional tableaux algorithm: it must be consistent and *have no other consistent branch that compares less in the partial order*. A consistent but non-minimal branch is therefore said to be 'weakly closed'.

We define the ordering over branches using symbols that are stored *within* the tableau. The set of propositions rank(*Index*, *Value*) are assumed to be tautologically true in any logical context, but are used for evaluating the order of the branches. The *Index* is an integer indicating the order in which the *Values* are to be sorted: branches are first compared by the values with smallest indexes of any rank term in the branch; equal values are compared using the next smallest rank term; and so on[4].

To illustrate this process, consider a robot upon encountering a novel toy ball. Using just a naïve stochastic hill-climbing strategy[5] for generating hypotheses, it may use observations of the object in order to build an accurate simulation of the

[4] This extension does not affect the consistency or completeness of logical deduction in the framework; the rank terms simply prioritize the search towards minimal branches.

[5] This strategy is effective in our example, but in real-world settings, more powerful algorithms may be used.

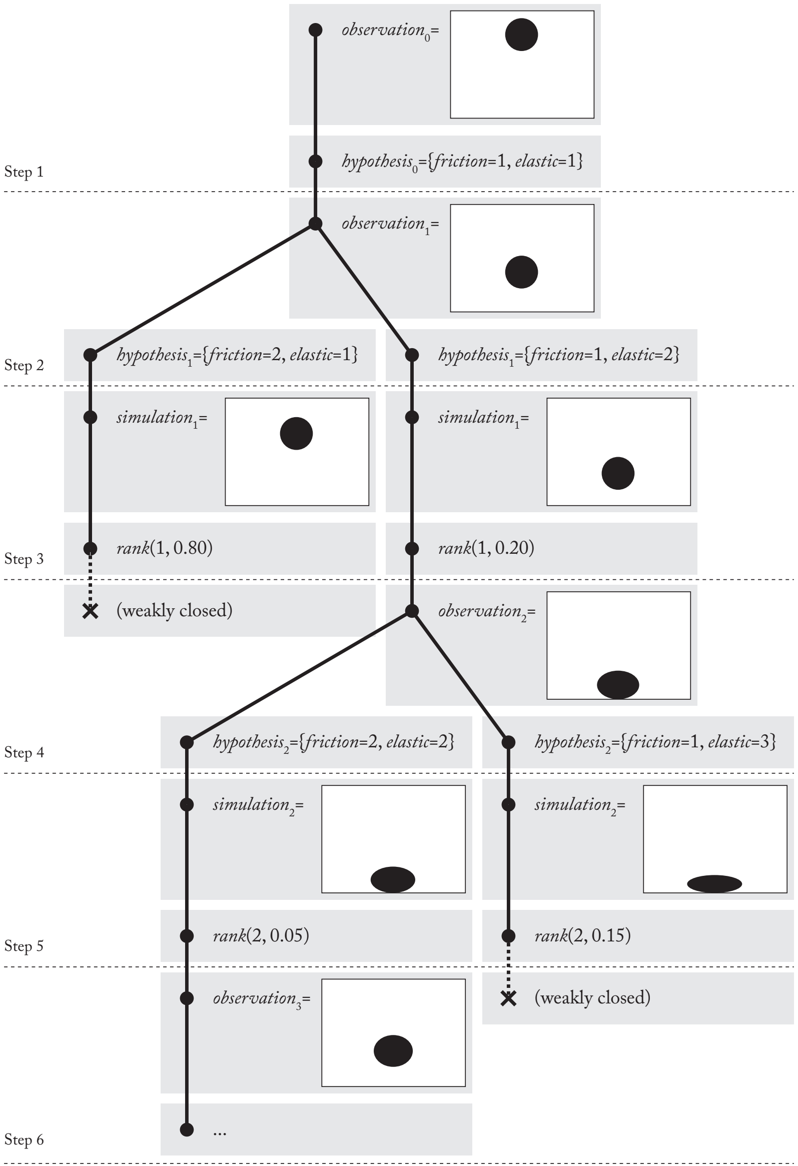

Figure 3: Learning in a tableau

object, as depicted in Figure 3:

Step 1. The tableau initially contains the first observation of a ball and the initial hypothesis generated (many other control objects, meshes, functions and other data will be in the tableau, but these are not shown for simplicity).

Step 2. The system observes movement in the ball. It generates new hypotheses, seeking to find a hypothesis with minimal error.

Step 3. The system simulates each hypothesis. The result of the simulation is compared with observations to determine the error in the hypothesis. The right branch has smaller error so the left branch is 'weakly closed'.

Step 4. As with Step 2, the system observes more movement and generates new hypotheses, further refining the current hypothesis.

Step 5. The system then simulates as with Step 3, but this time the left branch is minimal.

Step 6 and later. The algorithm continues yet again with more new observations and further hypothesizing.

Note also that because learning occurs in an (extended) tableau containing logic, simulations and arbitrary functions, the system may use logical constraints or ad-hoc 'helper functions' even when searching for values in a simulation (*e.g.*, it may use constraint such as *mass* > 0, or a heuristic-driven hypothesis generator to produce better hypotheses faster).

Furthermore, the ordering induced by rank terms finds application not only in driving the search for good hypotheses, but also in selecting between actions. Possible actions are treated as disjunctions in the tableau, and the error between the agent's goals and its simulated expectation is computed, so that the extended tableau algorithm may select the branch with minimal error.

Comirit and the Toy Box Problem

The Comirit Framework combines simulation, logical deduction and machine learning; as such, it is ideally suited to the physical reasoning (well suited to simulation), abstract reasoning (well-suited to tableau-based logical deduction), learning (as parameter search in the tableau) and action selection (as action search in the tableau) in the Toy Box Problem.

There is insufficient space here to provide a detailed analysis of the problem, and indeed, this work is itself ongoing (hence the 'preliminary' nature of the solution), however our early results are encouraging.

We conduct an experiment as depicted in Figure 4. A virtual 2D world is simulated with models of simple toys including boxes, balls, bagels and bananas all of varying, weight, appearance and strengths all of which are subject to 2D physics. The agent may only observe the world through 2D raster images (it cannot observe the underlying models), and it must construct its own internal models and predictions. Accuracy is measured by projecting the agent's belief back into a raster image and performing a pixel-by-pixel comparison (this is a demanding test since it makes no allowances for 'nearly' correct).

In our early experiments we have the system learn two hidden parameters (mass and spring constant). These two parameters are combined in a vector and serve as the hypothesis space for the learning problem. Even though we use an extremely simple machine learning algorithm (stochastic hill-climbing search), a single pair of visual observations is sufficient for the agent to achieve 95% accuracy in predicting an object's future behavior. This astounding learning rate is depicted in Figure 5. The slight improvement from subsequent observations comes from the elimination of minor overfitting—the accuracy is as good as may be achieved given the differences in the underlying models.

This incredible learning rate is possible because a single pair of images (before and after) contains thousands of pixels, serving as thousands of data-points for training. Indeed, this learning rate aligns with the human competence in develop-

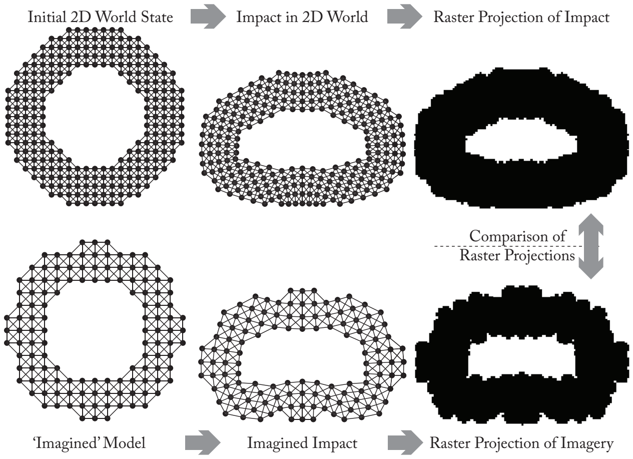
Initial 2D World State
Impact in 2D World
Raster Projection of Impact
Comparison of Raster Projections
'Imagined' Model
Imagined Impact
Raster Projection of Imagery

Figure 4: Setup of initial experiments

ing a 'good intuition' for an object (whether it is heavy, hard, smooth, fragile, *etc.*) after just a fleeting interaction.

We have also begun exploring the use of naïve Self-Organizing Maps (SOM) (Kohonen 1998) for learning the underlying structure of the world (*e.g.*, that 'bagels' are generally light, that balls are often soft, and that metallic objects are usually solid). In this case, the hypothesis becomes an entire SOM, combined with vectors for each visible object. When beliefs about toys are refined, a standard update is applied to the SOM. Our preliminary findings are that the ability for the SOM to generalize across instances roughly doubles the learning rate and provides better initial hypotheses about unknown objects. However, these are early findings and we will report on this in more detail once we have refined the model.

Finally, while the concrete implementation of action selection remains as future work, action selection does not present any theoretical challenge. Given simulation that has been learnt, actions (or sequences of actions) are be selected by searching (in an extended tableau) for a sequence that, when performed in simulation, are closest to the agent's goals (e.g., the goal of tidying-up the toys).

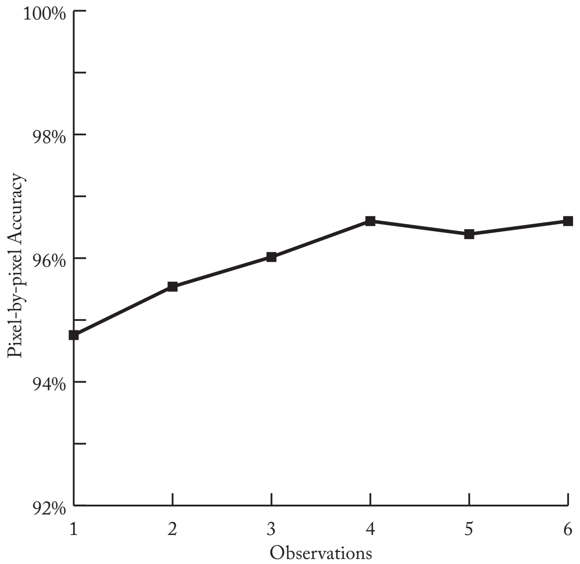

Figure 5: Learning rate in a 2D world

Conclusion

In this paper we have described an open ended benchmark problem that we believe is useful for evaluating and comparing the practical real-world intelligence.

We have also presented a brief overview of the Comirit architecture (with particular emphasis on the recent extensions for learning), and sketched how its capabilities may be applicable to the Toy Box Problem. A comprehensive adaptation and analysis remains, however our early indications (both qualitative and quantitative) suggest that Comirit will be able to 'solve' certain instances of the Toy Box Problem. As such, we believe that the pairing E1&T1 are within the realm of plausibility today.

Of course, much future work remains: comprehensive implementation and evaluation, more challenging environments and toys, and the development of methods for learning the fundamental laws of physics and 'learning to learn'. However, we believe that the Toy Box Problem provides an exciting framework for guiding and evaluating incremental progress towards systems with deep and practical real-world intelligence.

References

Cassimatis, N. (2005) 'Integrating Cognitive Models Based on Different Computational Methods', *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*.

Gardin, F. and Meltzer, B. (1989) 'Analogical Representations of Naive Physics', *Artificial Intelligence*, vol. 38, pp. 139–59.

Goertzel, B. and Bugaj, S.V. (2009) 'AGI Preschool: A Framework for Evaluating Early-Stage Human-like AGIs', *Proceedings of the Second International Conference on Artificial General Intelligence (AGI-09)*.

Hähnle, R. (2001) 'Tableaux and Related Methods', In Robinson, J.A. and Voronkov, A. (eds.), *Handbook of Automated Reasoning Volume 1*, MIT Press.

Johnston, B. and Williams, M-A. (2007) 'A Generic Framework for Approximate Simulation in Commonsense Reasoning Systems', *Proceedings of the AAAI 2007 Spring Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense 2007)*.

Johnston, B. and Williams, M-A. (2008) 'Comirit: Commonsense Reasoning by Integrating Simulation and Logic', *Proceedings of the First International Conference on Artificial General Intelligence (AGI-08)*.

Johnston, B. and Williams, M-A. (2009) 'Autonomous Learning of Commonsense Simulations', *Proceedings of the Ninth International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense 2009)*.

Johnston, B. (2009) *Practical Artificial Commonsense*, PhD Dissertation, Under Examination.

Kohonen, T. (1998) 'The Self-Organizing Map', *Neurocomputing*, no. 21, pp. 1–6.

Morgenstern, L. and Miller, R. (2009) *The Commonsense Problem Page*, <http://www-formal.stanford.edu/leora/commonsense/>, Accessed 13 October 2009.