View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by DSpace@MIT

massachusetts institute of technology — artificial intelligence laboratory

Leveraging Learning and Language Via Communication Bootstrapping

Jacob Beal

AI Memo 2003-007 March 2003

© 2003 massachusetts institute of technology, cambridge, ma 02139 usa — www.ai.mit.edu

Abstract

In a Communication Bootstrapping system, peer components with different perceptual worlds invent symbols and syntax based on correlations between their percepts. I propose that Communication Bootstrapping can also be used to acquire functional definitions of words and causal reasoning knowledge. I illustrate this point with several examples, then sketch the architecture of a system in progress which attempts to execute this task.

1 Introduction

Communication Bootstrapping is a phenomenon in which two language acquisition algorithms converge rapidly to a shared symbol-set and grammar via feedback and shared observations. Knowledge of this phenomenon derives from work by Kirby on language evolution, in which he demonstrated a population of grammar induction machines synthesizing an efficient grammar for expressing a simple shared semantics over the course of many generations. [Kirby1998] Further work by Kirby and others [Kirby2000, Kirby and Hurford2001] in the field of iterated language acquisition has expanded the original work to cover compositional grammars and a more precise understanding of the mechanisms which apply selection pressure to symbols in a grammar.

Communication Bootstrapping systems apply the ideas from iterated language acquisition to the problem of communication between heterogenous components. In particular, I have been interested in how different components in a cognitive system might learn to coordinate. There are strong arguments from neuroscience for the existence of modules in the human brain, such as the fusiform face area [Kanwisher et al.1997] and the parahippocampal place area [Epstein and Kanwisher1998] which are visual recognition specialists. Moreover, phenomena like the late integration of color cues into spatial reorientation [Hermer and Spelke1996] suggest strongly that communication between modules is a learned phenomenon critical to constructing human-level intelligence, which advises attention from any builder of cognitive systems who wishes to replicate human capabilities. Following these ideas, I have previously demonstrated a Communication Bootstrapping system which rapidly acquires a set of shared symbols and inflections capable of communicating thematic role frames between a pair of peer components with similar percepts. [Beal2002a, Beal2002b]

An important note, however, is that the critical requirement for Communication Bootstrapping is not identical percepts, but well correlated percepts (how tightly correlated depends on parameters of the bootstrapping algorithm). This can be leveraged to allow learning of more complex concepts, such as how to recognize a picture of a cup or what "tennis ball" means, in which percepts in the two peers have a relation other than identity.

In this paper, I will briefly review Communication Bootstrapping and show how it can be used to learn simple compound definitions. I will then explore two illustrative examples of learning complex concepts — functional definitions

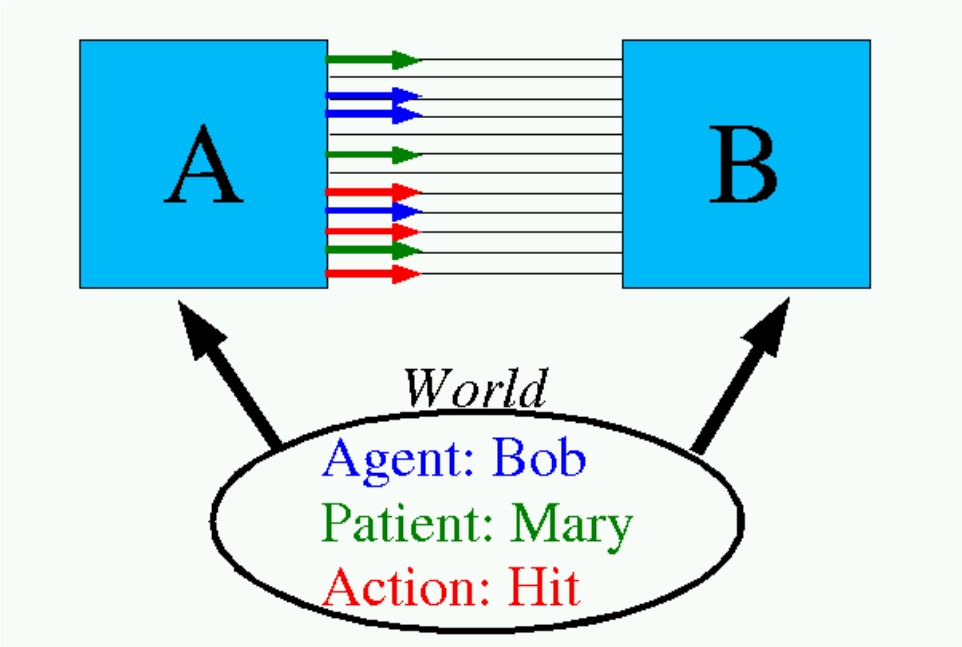


Figure 1: In communication bootstrapping, two agents receiving equivalent inputs from the world invent a shared language allowing them to communicate thematic role frames robustly over a thick bundle of twisted wires. In this illustration, the situation "Bob hit Mary" is being conveyed from A to B.

of words and causal reasoning — then sketch the architecture of a system in progress to realize this.

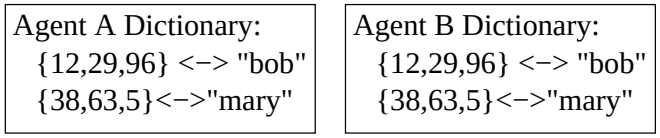
2 Bootstrapping Environment

In Communication Bootstrapping [Beal2002a, Beal2002b], two components are connected by a thick, twisted bundle of wires (i.e. there is some unknown permutation between the components). Each wire can be driven by either component and carries a bit, which reads as 1, 0, undriven, or conflict.

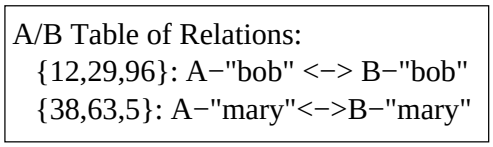
The components are presented with a semantics consisting of a set of $(object, role)$ pairs. For example, "John hit Mary" is represented as:

$$\{(John, agent)(Mary, patient)(hit, predicate)\}$$

Each object is mapped to a symbol expressed as a set of (initially random) wires, while roles are expressed as the proportion of 1s on a symbol's wires (Figure 1). In a system with plentiful wires, sparseness allows a rapid convergence of the two components to an identical set of object and roles mappings, provided that their input semantics are identical.



(a) Dictionary Interpretation



(b) Relation Interpretation

Figure 2: Communication Bootstrapping can be interpreted either as a problem of matching dictionaries or of finding relations. Previous work in Communication Bootstrapping has produced identity relations between the instances of an object in different agents.

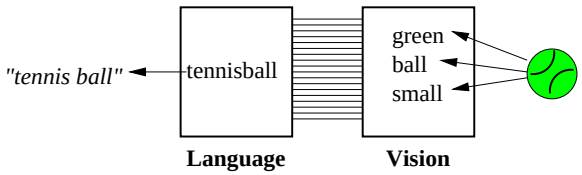
In [Beal2002a], I interpret the vocabulary developed by the two components as a pair of dictionaries, which are considered to be correct when each component's copy is equivalent under mapping between the two perceptual worlds. An alternate way of interpreting the system state, however, is as relations between two sets of objects, which happen to be in different agents. Under this interpretation, Communication Bootstrapping (as implemented in [Beal2002a]) establishes binary identity relations, and each component's interpretation of a relation specifies one member of the identity (Figure 2).

In the following sections, I will discuss Communications Bootstrapping systems where each component represents a major mental faculty corresponding loosely to a section of the mammalian brain: e.g. a vision component, a speech I/O component, a non-linguistic auditory component, a kinesthetic component, etc. The components are, in general, connected to one another pairwise in a complete graph, with a Communications Bootstrapping system running on each paired connection.

3 Learning Simple Definitions

In addition to merely knowing a set of communicable symbols, we would like to have our system learn something of their definitions. For example, we might want to learn that a tennis ball is a small green ball, or that a robin is a red bird.

It turns out that these sorts of definitions are supported very simply by



(a) Recognizing a Tennis Ball

Agent A Dictionary:
{72,99,26,38,27,18}
<–> tennisball

Agent B Dictionary:
{72,26} <–> "ball"
{18,38}<–>"small"
{99,27}<–>"green"

(b) Symbol Dictionaries defining "Tennis Ball"

Figure 3: The conjunction of three visual attributes ("green", "small", and "ball") stimulates the linguistic phrase "tennis ball". Each visual attribute's symbol is mapped to a distinct portion of the wires for the phrase symbol, so that only objects which are small and green and ball will be recognized as tennis balls.

Communication Bootstrapping. In [Beal2002b], I note that when a single object A presented to one component is presented to the other as a pair of objects B_1 and B_2, the system, unsurprisingly, creates a symbol which is interpreted as the A in one and B_1 and B_2 in the other.

This relation can be either an AND or and OR, depending on how the wires are allocated. If the symbol wires are all mapped to both B_1 and B_2, then transmitting either B_1 or B_2 will stimulate reception of A, and the relation is an OR. If, on the other hand, half the wires are allocated to B_1 and the other half to B_2, then if only B_1 or B_2 is transmitted, the stimulus threshold for A will not be met — in other words, the relation is an AND. Between these two extremes are a continuous range of functions, so that one could create a symbol that stimulated A when, for example, 6 out of 8 B_i symbols were transmitted.

These sorts of combinatoric relations can be used to learn about tennis balls and robins. Consider a Communications Bootstrapping with two intelligence domains, vision and language. The vision system is presented with the image of a small green ball in conjunction with the language system being told "tennis ball", but small red balls, big green squares and other near-miss counterexamples are not described as tennis balls. A Communications Bootstrapping system should be able to learn a symbol encapsulating an AND relation (Figure 3).

These combinations of features can be viewed as a propositional logic. For example, our tennis ball symbol may be reinterpreted as a equivalence state-

ment:

$$tennisball \leftrightarrow green \wedge small \wedge ball$$

. This presents obvious difficulties in situations with multiple objects: a small fuzzy mouse next to a big green melon should not be recognized as a tennis ball. This problem can be partially abated through the use of roles, which can transform the situation into something more like a limited universe first order logic. Implications also present difficulties, although they can be created via manipulation of the wire allocations.

With some small modifications, however, the relations can instead be binary relations expressing containment and class — the *is-a* and *has-a* relations forming the basis of frame systems. The tennis ball example could then be described in frame terms as "an instance of ball, with properties size=small and color=green."

4 Learning Functional Structural Descriptions

I will now step forward into a slightly more speculative domain. In this section and the next, I will show scenarios illustrating how difficult concepts can be exposed for capture by Communications Bootstrapping. First, let us consider learning functional descriptions of objects.

Our description of a tennis ball isn't quite satisfactory yet: we would really like to capture the idea that tennis balls are used for playing tennis. In other words, we want to add functional elements to its description, which has advantages demonstrated clearly by the MACBETH system. [Winston1982, Winston and Rao1990] Unfortunately, many useful functional predicates are not easily expressed in English — for example, in the MACBETH cup example, some key properties are "capable of carrying liquids", "upward pointing concavity" and "light enough to be lifted", none of which are easy to define. In a system with visual and motor components, however, these functional properties are exposed for learning via Communication Bootstrapping.

To illustrate this, consider the following scenario, where we are trying to teach a system with three intelligence domains — vision, motor, and language — about cups.

Following in the footsteps of MACBETH, we decide to teach the system about cups by first exposing it to a brick, a glass, a briefcase, and a bowl. Unlike in the MACBETH system, however, we don't provide any linguistic input, and just let the system play with the four objects using a manipulator arm.

Moving things around and feeling with its manipulator arm, plus camera input tracking the arm, gives two sources of correlated world data and stimulates symbol learning in the Communication Bootstrapping system between the vision and motor components.

First the system tries to pick up the brick and finds that it's too heavy to move. It can slide it around though, and the brick, being heavy and flat, doesn't change orientation. The glass, on the other hand, falls over when the brick runs into it, and the system, curious, shifts its attention to it and discovers that it

can pick it up — in the process, learning a new symbol that means "upward motion" to the vision component and "low weight" to the motor — in other words, "light enough to be lifted".

Seizing the moment, we pour water into the bowl and the glass, which is being held upright at the moment. Water is something we've taught the system about earlier — it has a "water" symbol which translates to "cold and pressure everywhere" in motor and a stereotypical texture in vision — so it recognizes that there is water associated with both these objects. When it turns its manipulator, however, the water dumps out of the glass, inciting its curiosity. A few more messy experiments, however, and it has new symbols for "upward pointing concavity" (the tactile sensation from sticking its manipulator inside linked with distance information from vision) and "capable of carrying water" ("upward pointing concavity" and a lack of visible holes).

Finally, playing with the briefcase and the bowl, it discovers that the briefcase is much easier to pick up, provided that it grasps the right part, and thus, through a similar process, learns "handle" and "easy to grasp".

Now, with all the predicates in place, we give the system a coffee cup with a cheesy slogan, a handle-less tin cup for camping, and a disposable styrofoam cup, telling the language component "cup" for each one. The vision and motor systems tell all sorts of symbols to the language component, including "metal", "squishy", "shiny", "breakable", and "#1 Dad". The only ones in common to all three cups, however, are "light enough to be lifted", "upward pointing concavity", "capable of carrying water", and "easy to grasp". Thus, the word "cup" becomes linked with symbols describing it as something easy to grasp, light enough to be lifted, with an upward pointing concavity capable of carrying water — and easily tested predicates in the vision and motor domains that can verify whether an object is, in fact, a cup.

Thus the system learns a functional description of cups, as per [Winston1982], but its description is composed of non-linguistic symbols exposed and easily learned by the combination of vision and motor components.

5 Learning Causal Knowledge

Another relation that can be exposed to learning by Communication Bootstrapping is causal knowledge. In this case, the relation is implication, with one component's interpretation being the cause and the other being the effect.

To illustrate an example of how we could learn causation, consider the following scenario, where a system with two intelligence domains — vision and motor — is playing in a classic blocks world.

At first, it flails around like a human infant, waving its manipulator arm randomly. Already, it can begin learning the simple cause and effect rules of hand-eye coordination: when I make *this* motion, the image changes *that* way. From this it starts to create symbols describing these couplings — symbols that we might interpret as, for example, "go left" or "go up". Then something big happens: the arm hits a tower of blocks and knocks it over! (Figure 4)

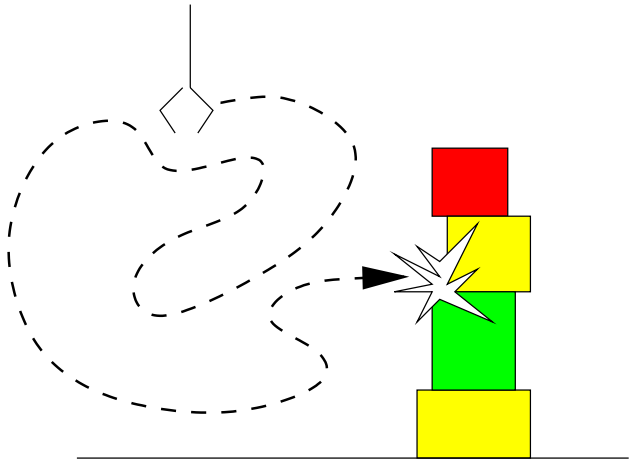


Figure 4: A system of two components, vision and motor, learns hand-eye coordination by waving its arm around randomly. When the arm accidentally crashes into a tower of blocks and knocks it over, the system learns a new symbol meaning "my arm hits something" to motor and "blocks go flying everywhere" to vision. The combination of this symbol across the two systems can be interpreted as a rule.

Hitting the tower wasn't intentional — at this point, the system is still just waving its arms around and seeing what that looks like — but it happens anyway, and now the system learns something very different, a symbol which means "my arm hits something" in motor and "blocks go flying everywhere" in vision. Now correlation becomes causation: this new symbol is, in fact, a constraint describing world dynamics that can be viewed as two complementary rules: "If my arm hits something, then blocks go flying everywhere" and "If blocks go flying everywhere, then my arm hit something." (The second one happens to be less often true than the first, and censors can be added to constrain its application) This piece of knowledge doesn't exist in either the vision or the motor system — it is the result of having a symbol which means something different to the two different systems.

Once the system has a symbol for knocking over a tower with its arm, it has the power to predict when it will happen and choose whether or not it wants that result. To see this, we fast-forward to some time much later in the system's development, when it has developed wants and goals. Now there is a four-block tower, on one side of which is the arm, and on the other side is a fifth block (Figure 5). The motor system has decided it wants a tower five blocks high — more precisely, it wants to have its arm simultaneously five blocks high and resting on something (which requires a five-block tower, but can be stated in purely motor terms).

The motor system, learning from the vision system that there is a four-high

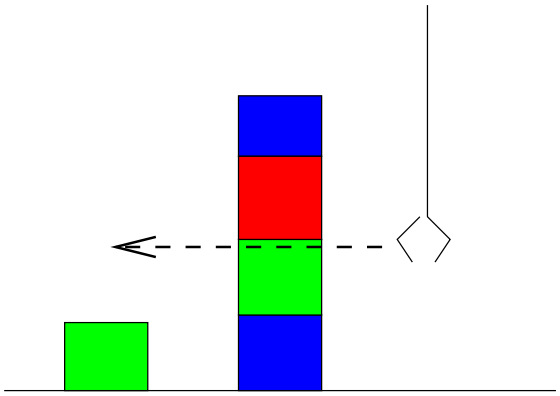


Figure 5: The arm needs to move and pick up the block on the left without knocking over the tower. When the motor system describes its naive motion to the vision system, the vision system says it will encounter the tower, which the motor system re-describes as the "my arm hits something" symbol. The vision system interprets it as "blocks go flying everywhere", which will thwart the plan.

tower and fifth block to its left, goes to pick it up, naively choosing the straight path. As it begins moving, the motor system says "going left." In the vision system, "going left" means the image of the arm moves left into the tower, a situation which it recognizes and describes as "arm contact on right." In the motor system, the combination of "arm contact on right" and "going left" add up to "my arm hits something." That symbol, of course, means "blocks go flying everywhere" in the vision system, which means it has to rescind the four-high tower it earlier told the motor system about — balking the motor system's goal.

At this point the system may become frustrated and start waving its arm around angrily, or it might know enough to negotiate the alternative route over the the top of the tower. In either case, the act of communicating the crashing symbol has produced causal reasoning behavior.

6 Mechanisms for Implementation

As is obvious in the examples above, the Communication Bootstrapping mechanism in [Beal2002b] alone is not sufficient. In addition, a system needs functionality to learn and transmit *is-a* and *has-a* relations, test symbols with near-miss learning, and impel it to explore its environment.

To this end, I am in the process of building a system where each component has a set of six additional mechanisms which, in combination with Communication Bootstrapping, will provide the missing functionality. The six mechanisms are, as follows: first, a general memory system, which serves as a repository for experiences and a provides services for constructing and manipulating *is-a*

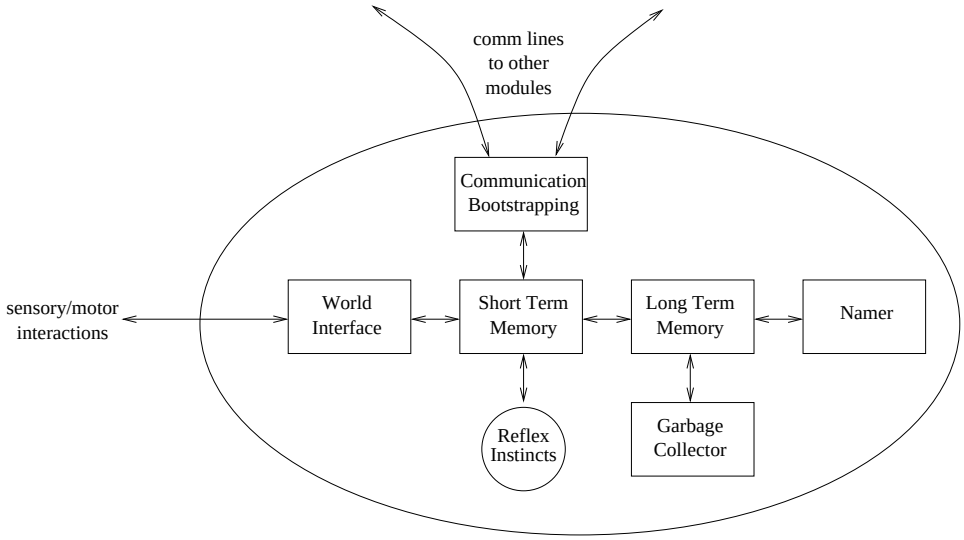


Figure 6: Proposed architecture for a component, coordinating episodic learning (long-term memory), abstraction generation (namer module), and Communication Bootstrapping to convey complex concepts between very dissimilar representations.

and *has-a* relations, which are the basic representation of all other components. Second, short-term memory, which serves as an attention mechanism and holds the current view of the world, and is also the view which Communication Bootstrapping is attempting to transmit. Third, a sensory/motor interface, coupling short-term memory with the system's sensors and actuators. Fourth, a reflex package, prioritizing reaction to stimuli and impelling the system to action when there is nothing to react to. Fifth, a naming mechanism, which uses proximity heuristics to invent new vocabulary, and finally a garbage collector, which handles forgetting. In the following sections, I give a brief sketch of each mechanism, and how it will fit into the larger scheme.

6.1 Relational Memory

All subsystems of a component operate on a common infrastructure of combinatorial atoms composed into NASH (Non-Axiomatic Structureless Heterarchy) diagrams, a loose representational infrastructure derived from standard frames systems and NETL.[Fahlman1979]

The basic element of a NASH diagram is an atom, which is simply an object with a unique identifier. Atoms are linked together to form a graph via directed "is-a" and "has-a" links. Links of type "is-a" create an inheritance web, and links of type "has-a" represent three types of structural relation, with the type determined by the tail atom in the relation: equality, membership, and sequence.

There are no syntactic constraints enforced on the arrangement of atoms within a NASH diagram. Instead, constraints are heuristically imposed by the mechanisms which construct and operate on a NASH diagram. Thus, for example, "is-a" relations will generally be arranged in a heterarchy, because a heterarchy is a good tool for capturing inheritance relationships, but there may be some violations of heterarchal structure.

In effect, NASH diagrams are a fairly standard frames system, with a syntax weakened to preference in order to reduce fragility and consistency maintainence overhead. The price of this weakening is a loss of consistency guarantees, but I believe that, much like the loosening of networking guarantees in amorphous computing systems, accepting a low level of inherent imprecision will enable more robust systems which degrade slowly rather than failing directly.[Abelson et al. 1999]

6.2 Short-Term Memory

Short-term memory has three roles in my system: it represents attention, caches recent knowledge, and performs shallow inference. Of these three, attention is the focal point around which the other functionality is built.

The short-term memory system is composed of three classes: attention, world state, and communication state. Attention is represented by a sequence atom. The attention atom's "has-a" links point to all of the items currently being attended to, in order of preferential attention.

6.3 Sensory/Motor Interface

The interface between the "real world" and a component is a constraint system which maintains correspondence between sequences of events in the NASH diagram (which hold past, present, and predicted events in a special doubly-linked chain structure) and the state of the sensors and actuators which are connected to the component. For sensors, changes in sensory input advance the sequence of events, while for actuators, changes in the sequence of events induce behavior of the actuators.

The event sequences are ladder-like structures of states and changes representing the history and predicted future of a variable, similar to the shift-registers used by Yip and Sussman. [Yip and Sussman1998] The states are atoms in the NASH diagram whose semantic content is determined by the "is-a" links which point to them, and the changes are the other half of the ladder, bridging between successive events. Finally, there are three summary objects which tie up the structure, a list of states, a list of changes, and a pointer to the present moment, which may be either a state or a change.

6.4 Reflex Package

There are two purposes served by the reflex package. First is to provide a default drive which prevents the system from ever being entirely quiescent. The

second is to shift the focus of attention to "attention-grabbing" events. The purpose for the default drive is to allow the system to act and learn in the absence of outside stimulus which would force it to. Reflex attention rules, on the other hand, are a collection of simple rules which describe "attention-grabbing" stimuli like flashing lights, fast-moving objects, sudden loud noises, or tactile pain. When a sensory event sequence matches the stimulus pattern for an attention rule, the rule places the it at the front of the attention sequence.

6.5 Namer

The namer is the abstraction mechanism by which a component reifies portions of its NASH diagram. The purpose of this subsystem is twofold: translation and reflection. The translation function is a necessity created by the highly heterogenous sensory/motor domains of different components. Abstraction creates less sense-specific symbols which should be easier to add to the Communication Bootstrapping shared vocabulary. Reflection, on the other hand, is an intra-component process by which the component can identify patterns in its own actions. This information can then be applied to debugging itself, as in Sussman's HACKER system, [Sussman1973] to produce more "intelligent" behavior, and in combination with the pruning done by Communication Bootstrapping, can implement near-miss learning of the sort described in [Winston1970].

6.6 Garbage Collector

Memory cannot be infinite, and so there needs to be some mechanism for purging un-needed information. Moreover, there is some evidence that forgetting may play a role in human generalization of experience, similar to Kirby's discovery that killing off experience aids in the learning of more general structures. [Kirby1998] In either case, the system needs a garbage collector to prune the long-term memory. The garbage collector runs over the entire NASH diagram contained in the system and garbage collects the portions which are deemed least relevant by criteria of age, elapsed time since last attended and number of incoming and outgoing links.

7 Contributions

I have described how a Communication Bootstrapping system can be leveraged to learn non-trivial information such as causal relationships and functional definitions of words, and sketched a system in development to demonstrate my claims. The ability to acquire these types of relations from multi-modal input is an important step towards building cognitive systems with human-level capabilities.

References

[Abelson et al. 1999] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. Knight, R. Nagpal, E. Rauch, G. Sussman and R. Weiss . "Amorphous Computing" AI Memo 1665, August 1999.

[Beal2002a] Beal, Jacob. "An Algorithm for Bootstrapping Communications" International Conference on Complex Systems (ICCS), June 2002.

[Beal2002b] Beal, Jacob. "Generating Communications Systems Through Shared Context" MIT AI Technical Report 2002-002, January, 2002.

[Epstein and Kanwisher1998] Epstein, R., and Kanwisher, N. "A cortical representation of the local visual environment." Nature 392, 598 601 (1998).

[Fahlman1979] Fahlman, Scott. "NETL: A System for Representing and Using RealWorld Knowledge." MIT Press, 1979.

[Hermer and Spelke1996] Hermer, Linda and Spelke, Elizabeth. "Modularity and development: the case of spatial reorientation." *Cognition*, 61:195–232, 1996.

[Hermer-Vasquez and Spelke1999] Hermer-Vasquez, Linda, Spelke, Elizabeth, and Katznelson, Alla. "Sources of Flexibility in Human Cognition: Dual-Task Studies of Space and Language." *Cognitive Psychology* 39:3-36, 1999.

[Kanwisher et al.1997] Kanwisher, N, McDermott, J, and Chun, M. "The Fusiform Face Area: A Module in Human Extrastriate Cortex Specialized for the Perception of Faces." Journal of Neuroscience, 17, 4302-4311 (1997).

[Kirby1998] Kirby, Simon. "Language evolution without natural selection: From vocabulary to syntax in a population of learners." Edinburgh Occasional Paper in Linguistics EOPL-98-1, 1998. University of Edinburgh Department of Linguistics.

[Kirby2000] Kirby, Simon. "Learning, Bottlenecks and the Evolution of Recursive Syntax." in "Linguistic Evolution through Language Acquisition: Formal and Computational Models" edited by Ted Briscoe. Cambridge University Press, in Press.

[Kirby and Hurford2001] Kirby, Simon and Hurford, James. "The Emergence of Linguistic Structure: An Overview of the Iterated Learning Model." in Parisi, Domenico and Cangelosi, Angelo, Eds. "Computational Approaches to the Evolution of Language and Communication." Springer Verlag, Berlin., 2001.

[Spelke et al.1994] Spelke, Elizabeth, Vishton, Peter, and von Hofsten, Claes. "Object perception, object-directed action, and physical knowledge in infancy." In The Cognitive Neurosciences, M. Gazzaniga, editor. The MIT Press, Cambridge, Massachusetts, 1994.

[Sussman1973] Sussman, Gerald. "A Computational Model of Skill Acquisition." MIT AI Technical Report 297, August, 1973.

[Winston1970] Winston, Patrick. "Learning Structural Descriptions from Examples." MIT AI Technical Report 231, 1970.

[Winston1982] Winston, Patrick. "Learning by Augmenting Rules and Accumulating Censors." MIT AI Lab Memo 678, May 1982.

[Winston et al.1982] Winston, Patrick, Thomas Binford, Boris Katz and Michael Lowry. "Learning Physical Descriptions from Functional Definitions, Examples, and Precedents." MIT AI Lab Memo 679, November 1982.

[Winston and Rao1990] Winston, Patrick and Rao, Satayjit. "Repairing Learned Knowledge Using Experience" MIT AI Lab Memo 1231, May 1990.

[Yip and Sussman1998] Yip, Kenneth and Sussman, Gerald Jay. "Sparse Representations for Fast, One-Shot Learning." MIT AI Lab Memo 1633, May 1998.