Leveraging Language into Learning

Jacob Beal

February, 2004

PhD Thesis Proposal

Abstract

I hypothesize that learning a vocabulary to communicate between components of a system is equivalent to general learning. Moreover, I assert that some problems of general learning, such as eliminating bad hypotheses, deepening shallow representations, and generation of near-misses, will become simpler when refactored into communication learning problems.

In support of this hypothesis, I illustrate how several types of general knowledge might be acquired by means of learning to communicate. I then propose a system on which this hypothesis may be tested, and a series of tasks on which it may be evaluated. Finally, I give desiderata for a general theory of learning via learning to communicate.

1 Introduction

"If you've found the right representation, the problem is already almost solved."

That is practically the first thing we teach students in Introduction to Artificial Intelligence here at MIT. Then, for most of the rest of the term, we teach techniques for solving problems forced into the Procrustean beds of particular representations.

This is hardly a unique irony in AI: although the field has had great success in solving nicely represented problems, our systems are still terrible at problems that we can't describe well with a simple representation. We simply have not yet developed practical algorithms that can learn or invent representations to deal with an unprecedented situation. Partly as a result of this lack, computers perform badly at unbounded tasks like understanding pictures, interpreting conversations, and participating in creative brainstorming—all of which people do very well. Accordingly, I turn to people for inspiration on the problem.

How is it that people can deal with confusing, unfamiliar situations? One thing they do is talk with other people! People brainstorm, free-associating ideas and trying to make sense of the detritus they produce. Teachers explain material to their students and end up understanding it better themselves. And being wrong is sometimes more important than being right: there are stories about people going to tell Marvin Minsky about an idea, which he then misunderstood as a much better idea than they'd started with!

Interestingly, there is some evidence that the human brain might operate on the same principle. Cognitive science research finds that the human brain may be composed of modules[7, 8] which learn throughout childhood to communicate and to reason cooperatively.[13, 5] Even in adults, collaboration between regions of the brain may depend on a language faculty. For example, "jamming" language prevents an adult from using color information to reorient[6] and native language biases how vision affects reasoning about time.[3]

For the human brain, the evidence is inconclusive, but it and the example of people learning by talking together lead me to conjecture a novel engineering principle: *a system composed of components which collaborate by constructing a shared vocabulary can adapt to novel or confusing situations.*

Although decomposing a system into independent components and forcing them to build their own communications will generally add significant complexity and overhead to a system, such an architecture provides three useful properties which may prove to greatly outweigh the increased costs:

- **Creative Misunderstanding:** Translating between components with dissimilar representations will tend to mutate the data being translated. Since mistranslation is systematic, however, the mutation tends to produce different sense rather than nonsense.[1]

- **Mistake Filtering:** Components with dissimilar representations will tend to make different mistakes. A group of components comparing possible conclusions can then collectively filter out most incorrect conclusions, greatly reducing the search space.

- **Reified Relations:** Communication by constructing a shared vocabulary means that relations between features are instantiated as vocabulary words. The presence of a vocabulary word can, itself, be a feature, allowing recursive learning on the reified relations.[2]

In my master's thesis, I demonstrated two agents creating a shared vocabulary and inflections, for the purpose of communicating thematic role frames.[1, 2] In that case, the two agents used very similar representations, so the vocabulary contains only identity relations.

There are other relations, however, such as cause and effect, or proximity, which should be nearly as easy to acquire. Incorporating non-identity relations and exploiting the advantages of creative misunderstanding, mistake filtering, and reified relations, a system which builds a shared vocabulary becomes potentially very powerful indeed.

I hypothesize that learning a shared vocabulary to communicate between components of a system is equivalent to general learning. Further, I hypothesize that such a system, given a novel or confusing situation, will be able to quickly learn to solve problems in that situation.

If my hypothesis is correct, it will have both theoretical and practical implications. On the theoretical side, the model can be applied to cooperative reasoning among regions of the human brain and tested by cognitive experiments. Practically, it will be a powerful tool for attacking representation-intensive problems, and a major step towards producing systems which exhibit human-like intelligence.

In the following sections, I outline a program of study aimed at verifying my hypothesis. First, I sketch an example of learning by learning shared vocabulary. I then illustrate some specifics on how learning shared vocabulary can be leveraged into other types of knowledge acquisition in Section 4, and in Section 5 how creative misunderstanding, mistake filtering, and reified relations can be exploited to simplify learning representations. I then describe a system on which this hypothesis may be tested in Section 6, and a series of tasks on which it may be evaluated. Finally, in Section 7, I give desiderata for a general theory of learning via learning to communicate.

2 An Example

Before diving into details, I want to give an illustrative example of learning by learning to communicate, at a high, hand-waving level.

Consider a system with two intelligence domains, vision and motor, playing in a blocks world. What does it mean for this system to be learning to communicate, and how can that give it capabilities which it did not have before?

At first, it flails around like a human infant, waving its manipulator arm randomly. Already, the system can begin learning 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 1)

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

[1] Kirby exploits this in his work on language evolution[9, 10], in which a coincidence in the speaker's utterances becomes a pattern in the listener's interpretation.

[2] Winston's Macbeth system[17] demonstrates the power of reified relations in metaphoric story understanding.

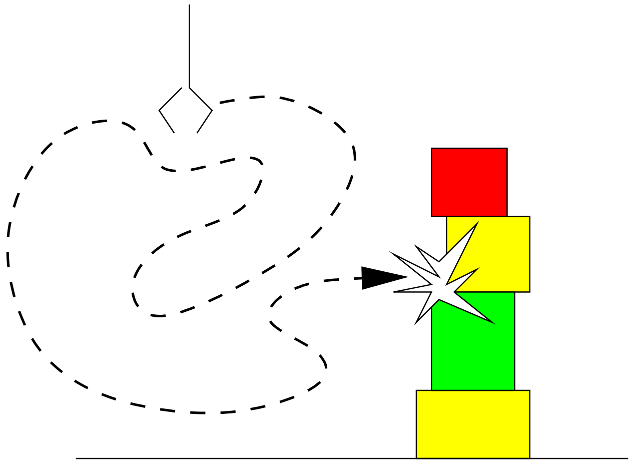

Figure 1: A system of two components, vision and motor, learns hand-eye coordination by waving its arm around randomly. When the arm accidentally crashes into a tower of blocks and knocks it over, the system learns a new symbol meaning "my arm hits something" to motor and "blocks go flying everywhere" to vision. The combination of this symbol across the two systems can be interpreted as a rule.

than the first) 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 2). 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 tower and a 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 the vision system 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 reasoning behavior.

3 Creating Shared Vocabulary

Two components can generate a shared vocabulary of words and inflections based on their shared experiences with the world. Figure 3 shows a base design for a system capable of generating shared vocabulary: the two components are connected together by a thick, tangled bundle of communication lines, and each component receives its own set of perceptions from the same external world. These perceptions may have any degree of similarity, from practically identical (e.g. left eye and right eye) to virtually none (e.g. olfaction and audition).

Learning in this system is inspired by Kirby's work on language evolution,[9, 10] in which the teacher invents nonsense words that are interpreted by the student as though they are intentionally structured. If the student finds a coincidence in the teacher's babbling, then during its turn as a teacher, it will teach the

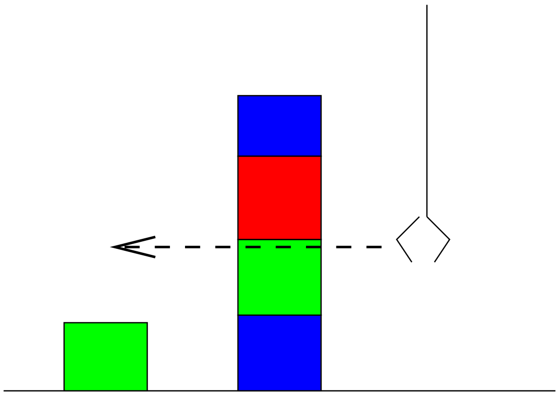

Figure 2: The arm needs to move and pick up the block on the left without knocking over the tower. When the motor system describes its naive motion to the vision system, the vision system says it will encounter the tower, which the motor system redescribes as the “my arm hits something” symbol. The vision system interprets it as “blocks go flying everywhere”, which will thwart the plan.

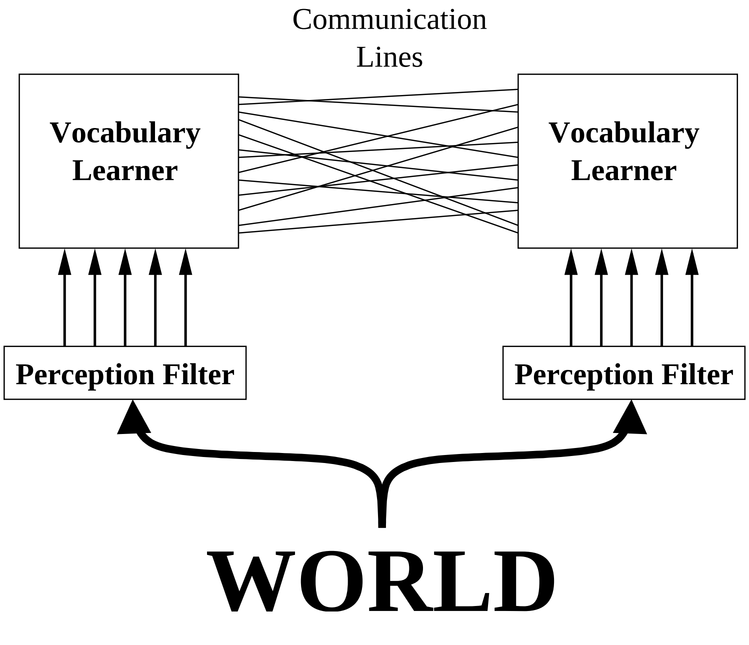

Figure 3: **Learning a shared vocabulary:** Two components which are to learn a shared vocabulary are connected to each other by a twisted bundle of many communication lines, and to a shared external world. Each component perceives the world differently, and correlations between their two perceptions can be learned as vocabulary.

coincidence systematically to its own student. The coincidences accumulate gradually over time, eventually going through a series of phase shifts into a structured grammar.

Although I retain the basic idea of accumulating coincidence to create structure, the specifics of my approach are completely different. My previous work with this system[1, 2] creates a shared word for each separable identity relation between the two components' perceptual worlds.

The components percepts are thematic role frames which pair roles and symbols (e.g. the set of pairs {`subject=mary, object=john, action=kiss`} means "Mary kisses John"). To add a symbol to its vocabulary, a component picks a random set of communication lines as a base, then intersects the sets of active communication lines whenever that symbol appears. After a few repetitions, either the two components' vocabularies assign the same set of lines to the symbol, or else it has been intersected into nothingness and they try again. Similarly, roles are represented as inflections on the symbols — the signal transmitted on the symbol's lines — so once a shared vocabulary of symbols is acquired, roles can be agreed upon quickly as well.

The vocabulary entries are a *bidirectional constraint*, similar to those in Yip and Sussman's phonological learner:[18] whenever a constraint is close to satisfaction, it asserts the nearest set of values that satisfy it completely. Thus, when a component perceives a symbol, that symbol's vocabulary entry is activated and stimulates the associated communication lines, and whenever enough of a vocabulary entry's communication lines are stimulated, the entry asserts its symbol (hallucinating it into its percepts).

Thus, the final product of learning in this system is vocabularies containing shared words and inflections, such that if one component perceives "Mary kisses John", its vocabulary constraints stimulate the communication lines, causing the other component's vocabulary constraints to activate and pop out "Mary kisses John" on the other side of the system.

4 Learning Relations As Vocabulary

The only relations for which learning a shared vocabulary has been demonstrated are identity relations. Building a shared word from repeated co-occurrences, however, should be able to produce a shared word to represent any strong positive correlation. In this section, I illustrate how several types of useful relation might be acquired as words in a shared vocabulary.

It is also worth noting that previous investigation has dealt exclusively with the behavior of a single pair of components. Other topologies, such as a graph of component pairings, or several components sharing a single set of communication lines, will likely be necessary for systems of sufficient complexity.

4.1 Difference

The complement of state is change; thus, in addition to learning words that represent particular percepts, it would be useful to have words that describe transitions between percepts. This is a more complex task, because the number of possible transitions is the square of the number of possible states.

One way in which this might be accomplished is by having a component build shared vocabulary with a time-delayed version of itself. Vocabulary entries for changes would then arise from repeating differences between the two components' expressions. For example, a system learning about stoplights would learn nothing while both sides showed green, because there were no consistent differences, but would discover the green to yellow transition as a difference which repeats often, and build a word which means green in the delayed percepts and yellow in the current percepts.

This vocabulary word could then be used to predict future states. After the light begins showing green, the delayed percepts would contain green, thereby stimulating the green/yellow transition word to be expressed on the communication lines, causing the current percepts to contain a hallucinated yellow — predicting the upcoming transition. For a state with several likely transitions, like an old Massachusetts red light which may change to either a red/yellow walk light or a green light, each transition would have its own word, and perceiving red would cause hallucination of the effects of both possible transitions.

Finally, note that the second component does not have to be a special component created for this purpose. If the second component is simply reflecting back the statement of the first component with some time-lag, it could have the same effect, even if its own percepts are completely different. Thus, differences in visual

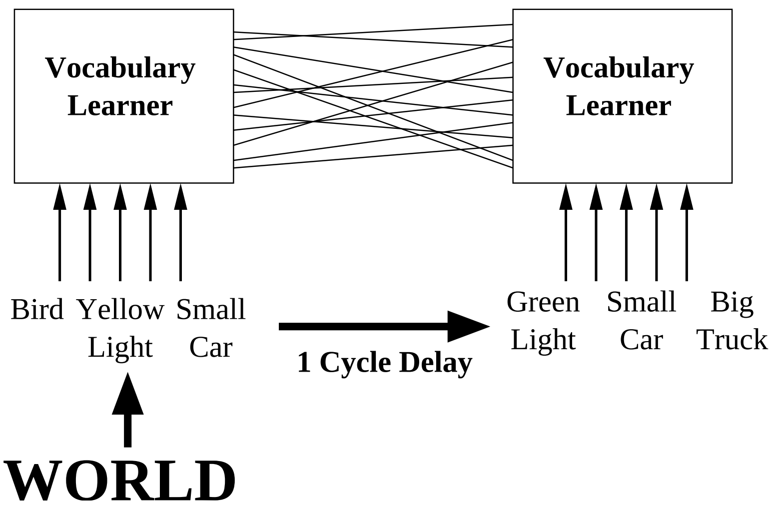

Figure 4: Transitions might be learned by comparing a component's percepts with a time-delayed copy. For example, a stoplight changes from green to yellow consistently, while other things in the scene do not behave consistently from transition to transition.

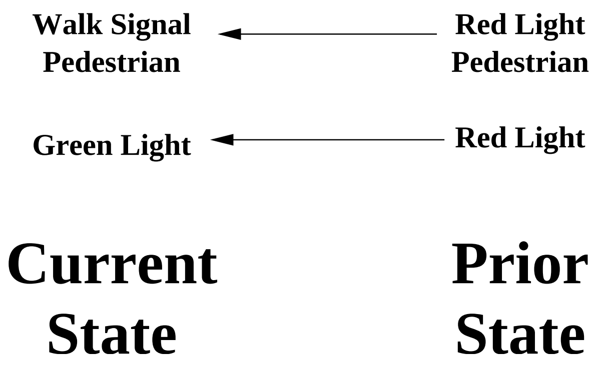

Figure 5: Causality can be synonymous with transition. For example, when a stoplight goes from red to a walk signal, there is almost always a pedestrian at the corner, while the presence of a pedestrian is uncorrelated with a transition from red to green. Thus the word learned for the transition to a walk signal may include as a precedent the pedestrian who caused that transition.

percepts might be learned via reflection off of the audition component, while at the same time differences in auditory percepts are being learned via reflection off of the visual component.

4.2 Causality

Cause and effect is a short jump from transition, especially since a good indicator of causality is temporal sequence. Again, as long as the cause and the effect appear together reliably, it should be possible to create a shared vocabulary word to represent the relationship.

For example, in the stoplight example for learning differences, a red light may change either to a green light or to a red/yellow walk signal. But the walk signal has another correlate besides a red light: there is also a pedestrian at the corner who pressed the walk button.[3] If a difference word is also allowed to include elements which are reliably the same, then the pedestrian will become part of the shared vocabulary word for the red light to walk signal transition. If the light changes from red to green, however, there may or may

[3]Learning that pressing the walk button causes a red/yellow signal is a harder task than simply spotting the pedestrian at the corner.

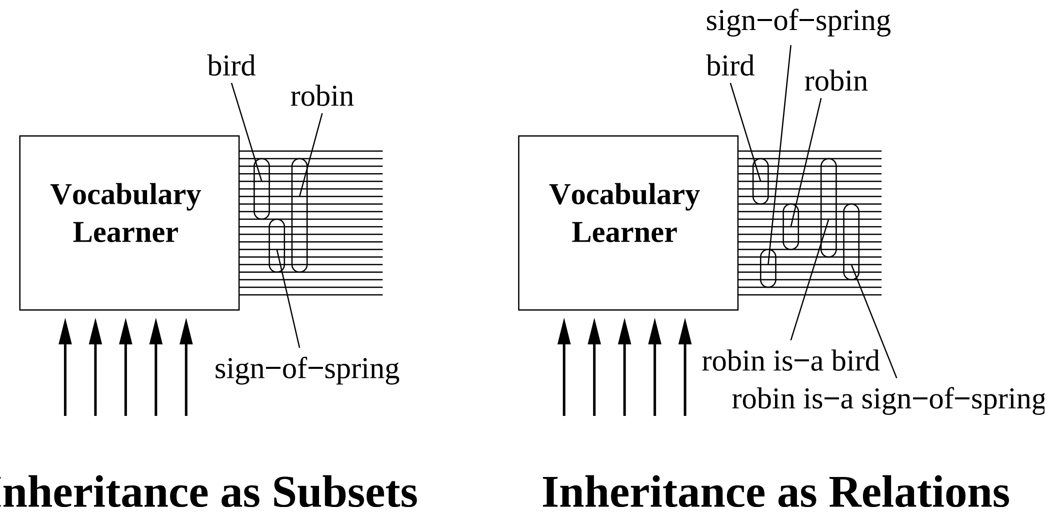

Figure 6: One way to learn an inheritance relation is to have the subclass's shared vocabulary word use communication lines which are a superset of the communication lines for the superclass's shared vocabulary word. This is simpler, but more costly and less flexible than learning an inheritance relation as a new shared vocabulary word linking the two sets of percepts.

not be a pedestrian at the corner, so the red light to green light transition will not include the pedestrian.

Thus this pair of transitions captures the pedestrian's causal effect on the light: for a red light when there is no pedestrian present, only a red to green transition is predicted. When there is a pedestrian present, red to green and red to walk are both predicted.

Learning causal relations which are not simultaneous will require an additional layer of complexity to connect the cause and effect in space and time. This could be done with an invisible piece of state that contacts both cause and effect: for example, pushing the walk button might cause "want walk" which persists and causes the red to walk transition. Bringing things together in time and space will be explored more in Section 4.4.

4.3 Subclass/Superclass

The simplest way to learn inheritance relationships is as subsets. If the percepts for a robin always include the percepts for a bird, then the shared vocabulary word for robin will include the communication lines representing bird, simply because there is no way to filter them out. Thus the word for a superclass is a subset of the word for a subclass. Multiple inheritance, in this scheme, is simply two subsets included in the subclass — so a robin can be both a bird and a sign of spring if both its parents' communication lines are included in its symbol. Thus, transmitting the word for the subclass stimulates the word for all of its superclasses as well.

To use this method, however, there are three requirements. First, the superclass must already be identified as a set of percepts, ruling out invention of new superclasses. Second, the superclass must almost always be perceived when the subclass is, which makes exception classes difficult (e.g. the relation between penguins and birds). Finally, the relation between the subclass and the superclass needs to be obvious when the shared vocabulary word for the subclass is being learned. All of these conditions can be addressed by learning inheritance relationships as shared vocabulary words separate from the classes which they relate.

For this method, the relation being noted would be the simultaneous stimulation of the shared vocabulary words for both superclass and subclass. Unlike an identity relation, it would not be necessary for both to appear, but rather for the subclass's appearance to be conditioned by the appearance of the superclass. Finally, if desired, the different portions of the shared vocabulary word can be weighted such that the presence of the subclass stimulates hallucination of the superclass, but not the other way around.

Thus perceiving a robin would stimulate the shared vocabulary word for robin as well as the shared

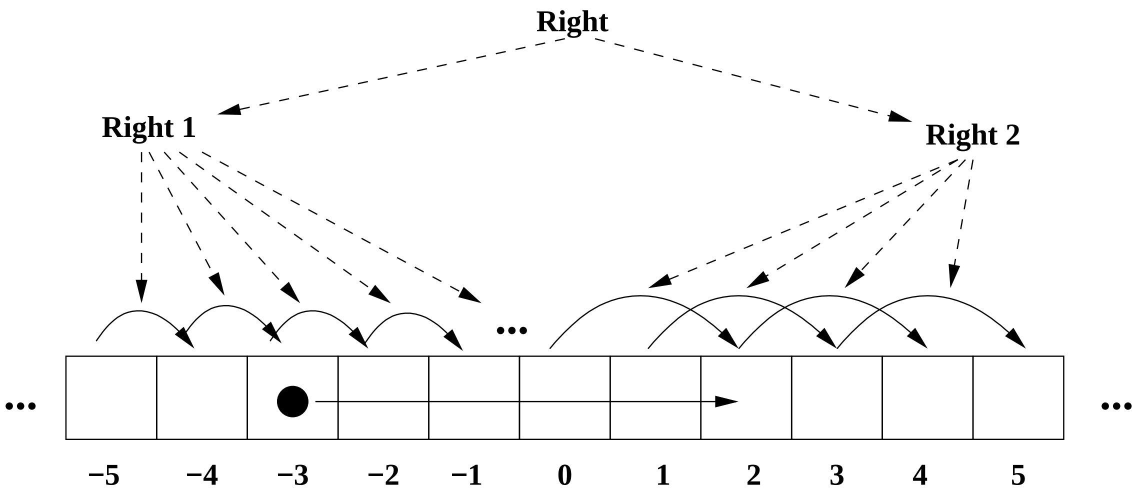

Figure 7: Proximity measures can be built by observing the percept transitions created by moving objects. Local percept transitions chain together into larger transitions, and classes of transition, which can then create a metric for the perceptual space on the basis of how large a motion links any two locations.

vocabulary words for its superclass relations, causing the percepts for birds and signs of spring to be hallucinated.

4.4 Time/Space Proximity

Transition and inheritance relationships, along with an inductance system for postulating new superclasses, can be used to create measures of relative distance between percepts. Consider, for example, objects moving left and right in a one-dimensional visual field (Figure 7).

Objects moving left and right in the visual field create regular percept changes. For example, a red dot moving right from location -3 will produce a reliable transition whereby location -2 starts perceiving a red dot, which can be learned as a shared vocabulary word. This transition, along with the same transition at other locations, can be lumped together under an invented superclass meaning relative motions of one step rightward. Thus -3 to -2 and 4 to 5 are both movements of type “right one.” By compounding single motions (or looking at longer transitions), other relative motions can be learned as well, and all of them accumulated under big categories like “moving right” or “moving left”.

From a scale of relative movements, a distance scale can be constructed as well. Locations -3 and 2, for example, can be added to a relation with movement “right 5”, since the motion “right 5” reliably transitions from -3 to 2. Finally, all locations related by the same movement can inherit from the “five apart” superclass, establishing a metric for the space. Thus, given any two percepts in this visual space, the pair of percepts would belong to a superclass which measured the distance between them.

In addition, spatial maps of various granularities might be constructed using part/whole relations between elements of the finer map (see Section 4.5) and the coarser elements which include them.

4.5 Part/Whole

Part/whole relationships are very similar to inheritance relationships, just as causality relationships are similar to transition relationships. Like an inheritance relationship, a part/whole relationship is one in which the whole and the part are generally both present at the same time. For a part/whole relationship, however, we might expect a proximity map (see Section 4.4) to show the part and the whole occupying different spaces.

Thus a part/whole relationship might be thought of as an inheritance relationship which also includes relative position relationships specifying the location of the part with respect to the whole. This difference

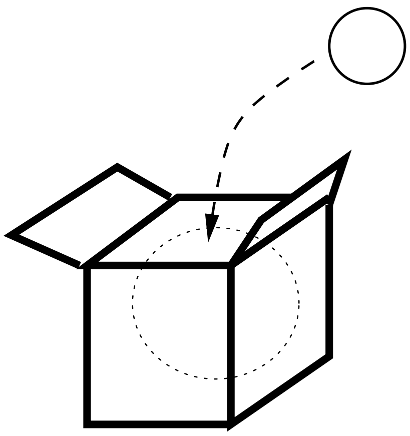

Figure 8: Containment may be viewed as a composition of a part/whole relationship and proximity relationships. For example, the interior of the box may be represented as an invisible part of the box occupying space in a fixed spatial relationship with the rest of the box. When the ball comes in contact with this part of the box, it may be interpreted as being contained by the box.

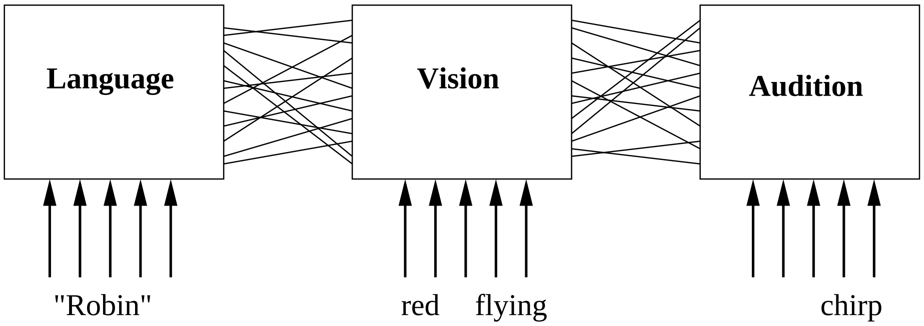

Figure 9: The attributes of an object, such as its color, weight, and other perceivable qualities, are simply the percepts which the shared vocabulary word is associated with in various components. For example, the word “Robin” might be associated with percepts for the color red, flying trajectories of movement and chirping sounds.

of location need not be a literal physical difference in location, but might be composed by metaphor or in abstract spaces as well.

4.6 Containment

Containment can be taken as a combination of a relative spatial relations and part/whole relationship, by conjecturing an invisible “empty space” part inside the container. Containment, then, is being in contact with a container’s space. For example, a box could have a part/whole relationship with its interior, which fills the space inside the box. Then if something is placed inside the box, it contacts this interior object, and may be interpreted as being contained by the box.

proximity to interior

4.7 Attributes

Finally, the attributes of an object are simply learned as its instantiation in different representations. For example, a robin is red, chirps, and moves by flying. Thus, the shared vocabulary word which associated with the percept for the English word “robin” in language would be associated with percepts of flying motion and red color in vision, and percepts of chirping sounds in audition.

Thus, for example, the language component perceiving the English word "Robin" would stimulate the shared vocabulary word which, among other things, causes the vision component to hallucinate the image of a small red bird. This, in turn, could stimulate the vision component to express its shared vocabulary word for red onto the communication lines, causing the language component to hallucinate the English word "Red".

5 Better Learning Via Communication

The utility of a representation lies in what it exposes for easy manipulation, and what it abstracts as implicit processes in order to reduce complexity. Learning via constructing shared vocabulary provides several key representational advantages which should be exploitable to solve hard problems of learning.

Mistake Filtering and Creative Misunderstanding Two advantages of abstraction are consequences of the fact that components in the system communicate imperfectly. This may be due to an inadequate vocabulary, differences in component sensoriums, or differences in learned interpretation across components. Whatever the case may be, this means that structures translated between components will often suffer systematic distortion. But this bug is a feature: since different components will tend to make different mistakes, bad conclusions are less likely to survive translation than good ones, and translation can be used as a collaborative filter. On the other hand, mistranslation can be used as a brainstorming mechanism, where the systematic distortion of mistranslation means it will be more likely to produce useful mutations than random garbage.

Reified Relations An advantage of exposure lies in the fact that the shared vocabulary may itself be used as percepts. This means relations are automatically reified into objects from which more relations may be extracted, greatly increasing the expressiveness of the language of relations, and allowing the learning of abstractions significantly removed from basic sensory percepts.

Finally, learning via communication strips away the illusion of coherence that permeates programming, forcing us to confront the Babel problem directly: the representations which we build tend to be fragile and subject to the disease of microtheories, in which knowledge crystallized for one application is incompatible with others, and therefor useless. Learning via communication, however, is fundamentally a problem of drawing different representations into usable (but not necessarily perfect!) coherence.

Exploiting these advantages, I believe learning via constructing a shared vocabulary can be used to make progress on some hard problem, such as:

5.1 Generating Near Misses

Learning from near misses (e.g. [16, 17, 18]) is extremely fast and effective. Since only examples relevant to a decision boundary need to be considered, every example can contribute to constructing the model being learned.

The problem, however, is that the world doesn't provide many examples of near misses, and when it does they are rarely labelled as such (the exception to both of these being instruction by a teacher). One solution to this, proposed by Yip and Sussman,[18] is to have a system generate its own near misses by applying external constraints: for example, since English words only have one correct pluralization, Yip and Sussman's system could swap endings of words pluralized differently to generate near misses.

Translation of an example between components may offer a similar capability. When an example is translated back and forth between two components, it will generally be distorted by the process. Any requirement not identified may be violated in this distortion (Figure 10) creating a near-miss. Since some translation distortions may still be hits, a test for near-miss is needed, but this is a significantly simpler problem.

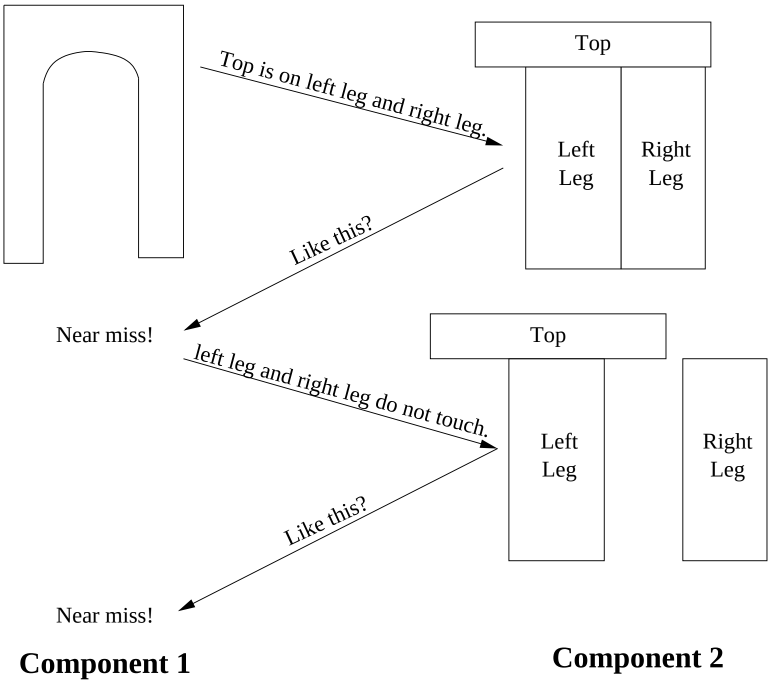

Figure 10: Mistranslation can generate near misses: the left component, perceiving the arch, gives the right component an inadequate description, and it resolves the underconstrained bits incorrectly, creating near misses.

5.2 Fast Hypothesis Searching

Unsupervised learning is often bogged down by the sheer number of reasonable hypotheses which can be made to explain events. For example, a child learning language has such a large menu of grammatical hypotheses to choose from, that it has been used as an argument in favor of an innate universal grammar. This typifies one approach to the problem of hypotheses: finding a set of additional constraints on the problem such that only a few hypotheses can be considered.

Learning via creating shared vocabulary, on the other hand, suggests an approach from the opposite direction: create lots of hypotheses, attempt to transmit them to the other component, and discard those which do not join the shared vocabulary.

For example, to learn that the word “open” refers to opening a door (Figure 11), the system could create hypotheses that open refers to everything even vaguely reasonable in the situation. The only hypotheses that can correlate between components to establish shared vocabulary, however, would be the ones that actually have something to do with opening the door. Note that learning shared vocabulary is an inherently parallel problem, so many hypotheses can be tested at once.

5.3 Deepening Understanding

Something which people are very good at, but that computers are currently very bad at, is understanding a situation on multiple levels at once. People often begin learning with a very shallow understanding of a subject, then deepen their understanding as they continue, transforming their models and building metaphoric connections to other knowledge.

Since relations are learned as vocabulary words, those vocabulary words can be themselves used as elements in a relationship. So from the basic sensory percepts of a component, nested layers of abstraction can arise to explain the same set of actions. For example, a kinaesthetic component playing with blocks might first learn regularities in its motion, then abstract the notion of “blocks” which can be moved around. Blocks lead to relations between blocks like “on” and “next to” and methods for manipulating those relations.

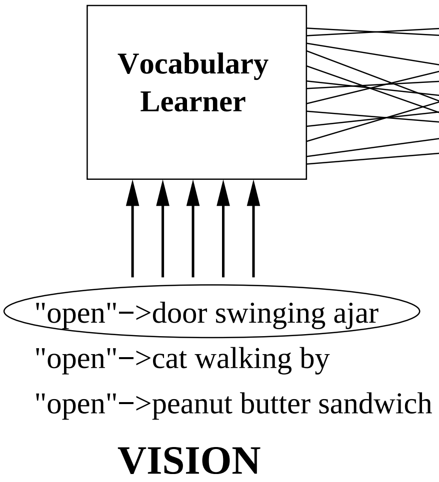

Figure 11: Fast Hypothesis Searching: to learn the meaning of the word "open" from visual and tactile context, hypothesize that all interesting stimuli are related, then eliminate the hypotheses which can't correlate across components to support shared vocabulary words.

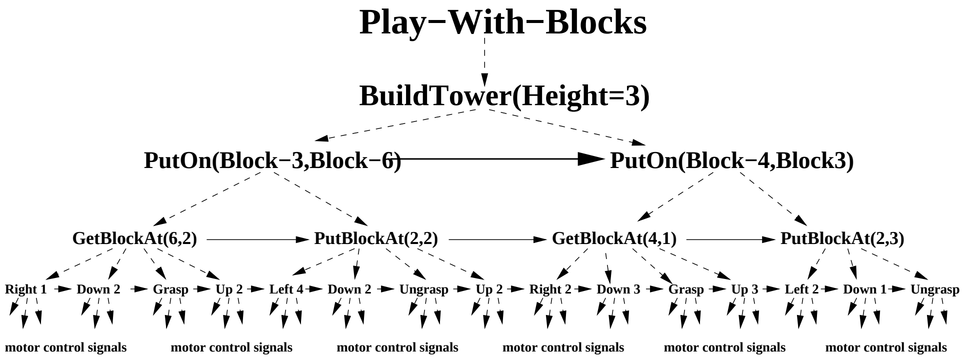

Figure 12: Playing with blocks on many levels, from the most abstract context and the current goal ("Build a three block tower") down through layers of operation to raw motor perception and actuation at the base. Learning proceeds from the base layer upwards, recursively building a new vocabulary for each level.

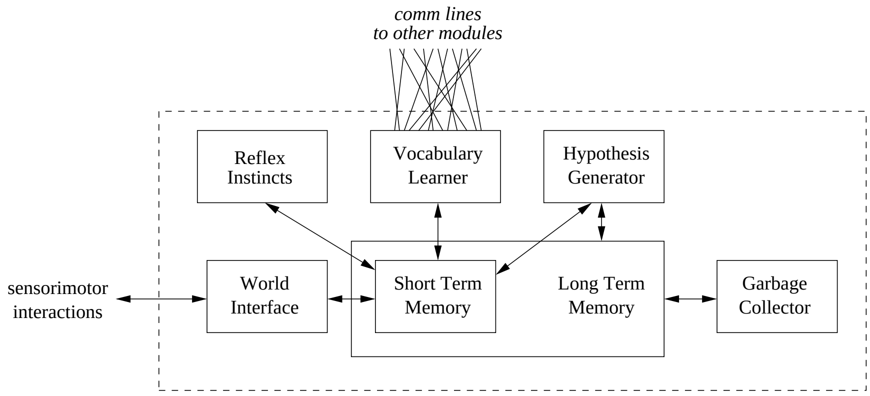

Figure 13: Proposed architecture for a component, coordinating episodic learning (long-term memory), abstraction (hypothesis generator), and shared vocabulary learning to convey complex concepts between very dissimilar representations.

Relations can be grouped into constructions like towers. Finally, at the highest level, all of these activities might be related under the heading, “playing with blocks”.

6 Language Learning Testbed

As a testbed for my hypotheses, I propose to use a system of independent components, each with a monopoly on a portion of sensory input. Placing it in an environment like the venerable Blocks World, I will be able to perform empirical tests of my hypotheses.

6.1 Component Design

Each component is composed of seven parts (Figure 13):

Long Term Memory A component's storage is a collection of perceptual primitives and typed relations. Relations may represent inheritance, containment, sequence, or identity, and connect between primitives or relations.

Short Term Memory Short term memory is a working set of memory elements. This working set is contains the current sensory percepts, a focus of attention pointing to a small set of elements, a cache of recently attended elements, and immediate relations of the focus of attention. The relations of the focus of attention can be used for shallow inference via marker propagation after the fashion of NETL.[4]

World Interface The world interface is a constraint mapping between sensorimotor primitives and short-term memory, which records a history of percepts and turns expectation into actuation. Perceptual primitives are organized into classes (e.g. light levels for a particular pixel in vision, or tactile sensations for a particular location) and each perceptual class is associated with a chain of events (relations associating with a perceptual primitive) that has an insertion-pointer in short-term memory. Events ahead of the insertion pointer are predictions, which the interface attempts to assert on the sensorimotor primitives; events behind the insertion pointer are a record of past percepts. The events may be further categorized into state and change events, giving a dual view of the perceptual class.

Vocabulary Learner The vocabulary learner takes attended elements from short-term memory as its percepts and is connected by a thick twisted bundle of wires to the vocabulary learners of other components. Incoming communication from other components is expressed as changes to short-term memory.

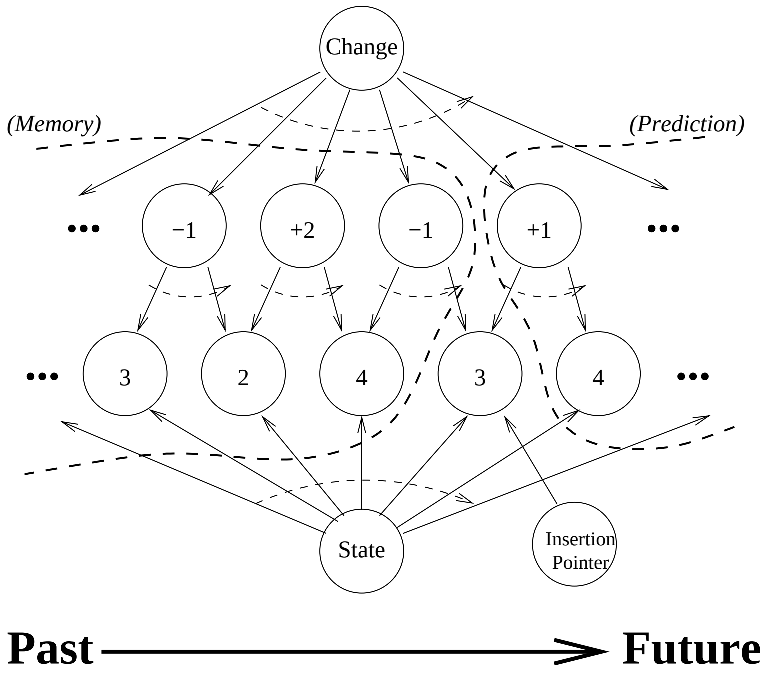

Figure 14: Sensorimotor percepts are connected to a chain of relations in short term memory recording past percepts and predicting/asserting future percepts.

Hypothesis Generator The hypothesis generator reflects on memory to introduce symbols not directly connected to any sensorimotor percepts. It scans short-term memory for patterns matching its heuristics, and then, if enough additional matches can be found in long-term memory, hypothesizes that the pattern is meaningful and binds the pattern to a new relation. In this way, new levels of abstraction can be created. Most hypotheses will be junk, but these can be weeded out as explained in Section 5.2.

Reflex Instincts There are two sets of hardwired reflexes which manipulate short-term memory. The first is a default drive which generates random activity at a low level in order to prevent the system from ever being entirely quiescent. The second is a set of rules shifting focus of attention to "attention-grabbing" events, like flashing lights, fast-moving objects, sudden loud noises, or tactile pain. The attention rules will generally be specialized for any given sensorimotor domain.

Garbage Collector The garbage collector serves to enforce memory limits and to promote generalization by causing specific examples to be forgotten. The garbage collector runs continuously, pruning the least relevant relations from memory on the basis of age, elapsed time since last attended, and number of links.

6.2 Example Problem Domain

To hammer out the details of a system and produce a working proof of concept, one needs a good problem to act as an anvil. Building towers from blocks is one such problem. As an anvil, this has the advantages of being a well-understood domain with simple dynamics, yet still holding much subtle difficulty.

My version of Blocks World has a human teacher and the learning system attempting to communicate about the underlying world of blocks, which is visible to both. There are three components in the system: vision, motor, and language. The components start from tabula rasa, with nothing but reflexes, and develop through interaction with the world and the human teacher.

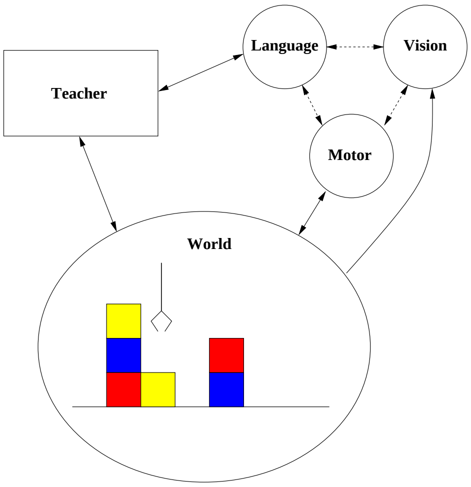

Figure 15: The Blocks World system is composed of the world, a human teacher, and three learning components (language, vision, and motor). Solid lines are sensory/motor interaction, dotted lines are communication bootstrapping channels.

6.2.1 World Dynamics

The Blocks World is two-dimensional, with one horizontal dimension and one vertical dimension. The space is continuous, but bounded in extent. Within this world, there are blocks, a ground plane on which blocks rest, and a manipulator arm. Blocks are rectangular, but vary in size and color and have a weight proportional to their area. The manipulator arm can move up, down, left or right, and can grasp a block on any part of its top.

When the arm grasps a block, it may move it if the block (plus any other blocks resting elsewhere on the top) does not weigh too much. If there is too much weight, then the arm cannot move at all while grasping the block. The arm can also move blocks by crashing into them from the side (or crashing a block which it holds into them), thereby knocking them over. Finally, gravity acts on the blocks, toppling unstable structures, and dropping any ungrasped block downward until it encounters a surface.

6.2.2 Component Sensory/Motor Interfaces

Each of the three components has a sensorimotor interface which simulates interaction with the world as appropriate for that component. The language component deals in word tokens, streaming between the component and the teacher. The vision component receives a low-resolution image of the world as its input, with a movable center of attention. The motor component controls the manipulator arm and can sense its location, contact with objects, grasping status, and weight of an object.

6.2.3 Teacher Interactions

The teacher can see the entire Blocks World, and can interact with the system either by manipulating the world or by talking to its language interface. When manipulating the world, the teacher hijacks the system's motor interfaces, a situation which is indicated to the system by pressure sensed on the top of the arm. (The analogy is of grasping a human student's hand and moving it to show the student how it feels to do something.)

In addition, the teacher can create and destroy blocks, something which the system cannot do — although the system can, of course, use its language interface to ask the teacher to do it for it. When the teacher creates or destroys a block, there is a brief omni-directional pressure wave felt by the motor interface, which should give a way for the system to cope with the "miracles" caused by the teacher.

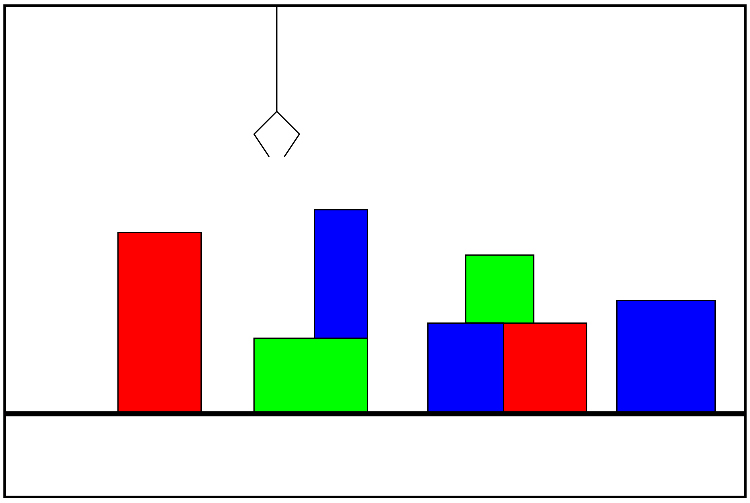

Figure 16: The Blocks World has colored rectangular blocks resting on a table, or on each other, and a manipulator arm capable of grasping one the top of one block at a time.

6.2.4 Problems To Solve

A successful system acting in the Blocks World will be able to solve a wide range of problems illustrating deep understanding of the dynamics of the Blocks World, as well as the abstract concepts used by the teacher to describe it. This learning is demonstrated through its interactions with the human teacher.

More concretely, the system should learn about simple perceptual things like “red”, “blue”, “left”, “down” and “arm”. It should also learn slightly more complex concepts like “block”, “put”, “get”, “move” and “fall”. Moving to even greater complexity, it should learn about compound objects like “tower”, “pyramid” or “wall” and themes like questions, commands and pronouns. Finally, it should learn to introspect into its own operations to learn about mistakes, plans, and predictions.

A mature system, when presented with a novel scenario, should be able to produce interactions such as in this sequence:
Human: “(get (blue block))”
Computer: [Picks up a blue block] *(basic vocabulary and skills)*
Human: “(put (blue block) (on (red block)))”
Computer: [Puts the block on some red block] *(persistent reference)*
Human: “(build tower)”
Computer: [Puts three blocks in a stack] *(abstraction and complex goal)*
Human: “(knock-down it)”
Computer: [Disassembles the tower] *(shared reference [via pronoun])*
Human: “(make tower bigger)”
Computer: “(cannot-comply (knock-down it))” *(failure detection)*
Human: “(rebuild it)”
Computer: [Rebuilds same tower, adds two more blocks] *(recursive dialogue)*
Human: “(build tower (using (green blocks)))”
Computer: “(cannot-comply (is too-few))” *(predictive reasoning)*
Human: “(what-if? (is three))”
Computer: “(then (can (build tower)))” *(counterfactual reasoning)*
Human: “(why-is? different)”
Computer: “(tower requires (at-least three blocks))” *(explanatory reasoning)*

If the system is really successful, maybe we’ll be able to hold a discussion of the Sussman Anomaly with it.

7 Steps Toward a Theory

Assuming that the empirical tests bear out my hypotheses, it should be possible to formulate a general theory of equivalence between learning shared vocabulary and general learning. I look for the form of this theory to be as follows:

Given any system C, there exists a system C' consisting of two or more components which is functionally identical to C. For any execution E of C with input/output trace T, there is an execution E' of C' which has an identical trace. Furthermore, if C acquires a fact F during execution E, then C' acquires an equivalent set of shared vocabulary words F' during E'.

In other words, it should be possible to show that any learning system can be replaced by a learning shared vocabulary system which behaves indistinguishably. If, however, it turns out that it is not equivalent, then it should be possible to show a set of practical conditions under which learning shared vocabulary can, in fact, be substituted for other types of learning.

It should also be possible to quantify the benefits of refactoring into a learning shared vocabulary system. I do not propose a form for that analysis at this time.

8 Contributions

I have proposed an ambitious hypothesis and laid out a course of work for an assault upon it. I hypothesize that learning a shared vocabulary is equivalent to general learning, and give some illustration of how learning shared vocabulary can be leveraged to acquire many types of information and to improve our ability to deal with some hard learning problems. I then sketch a design for components with which to run experiments in this area and obtain evidence for my hypothesis, and sketch a blocks world and some hard tasks to develop the system against. Finally, I sketch the form of a general theory of learning via learning shared vocabulary which should be provable if my hypothesis is correct.

If my hypothesis is correct, it will have both theoretical and practical implications, suggesting directions for cognitive research and giving a powerful set of new tools toward producing systems exhibiting human-like intelligence.

References

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

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

[3] Boroditsky, Lera. "Does Language Shape Thought? English and Mandarin Speakers' Conceptions of Time." *Cognitive Psychology* 43:1-22, 2001.

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

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

[6] 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.

[7] Kanwisher, Nancy. "The Modular Structure of Human Visual Recognition: Evidence from Functional Imaging." Invited address published in Sabourin, M., Craik, F., & Robert, M. (Eds.), Advances in Psychological Science. Volume 2: Biological and Cognitive Aspects. p. 199-214, 1998.

[8] Kanwisher, Nancy and Wojciulik, Ewa. "Visual Attention: Insights from Brain Imaging." Natural Reviews Neuroscience, (1), 91–100, 2000.

[9] 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.

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

[11] Minsky, Marvin. "K-Lines: A Theory of Memory" MIT AI Lab Memo 516, June 1979.

[12] Minsky, Marvin. *The Society of Mind*. Simon & Schuster, Inc, New York, 1985.

[13] 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.

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

[15] Taylor, Michael. Randomly Perturbed Computation Systems. MIT Ph.D. Thesis, 1966

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

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

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