Getting a Grip: A Computational Model of the Acquisition of Verb Semantics for Hand Actions

David Bailey
Computer Science Division, Univ. of California at Berkeley
and International Computer Science Institute
1947 Center St. Suite 600, Berkeley CA 94704
dbailey@icsi.berkeley.edu

Abstract

We present a computational model of how verbs might be learned within the limited domain of hand actions. We hypothesize that such verbs refer to the activities of underlying motor schemas, and leverage this constraint to build a system with strong enough biases that it can learn from a reasonably small number of examples, while still having adequate flexibility to learn the hand-action verbs of any language. The completed system should demonstrate its knowledge both by labelling its own behavior and by carrying out verbal commands in a simulated world.

1 Overview

How do children learn to describe simple actions, such as lifting a mug or pushing a box? This paper outlines a computational model of children's acquisition of verbs, as well as an activity-based model of their semantics, within the restricted domain of actions that can be performed by one hand upon simple objects on a tabletop. In English, these verbs would include *push*, *pull*, *shove*, *grab*, *hold* and *touch*. Beyond these simple verbs, our model also endeavors to learn verb "complexes", such as *pick up*, *unzip*, or *keep hitting*.

The model will be evaluated and refined by incorporating it into a computer system that can learn the relevant verbs within a context of carrying out actions in a simulated world. The system learns from labels attached to its actions, and must demonstrate its acquisition of the verbs in two ways. First, it must be able to label novel actions—i.e., recognition. Second, it must be able to carry out appropriate actions when given a verb as a command in a novel situation. A human-body animator connected to our system will be used to collect training data and to evaluate the trained system.

This work is part of the L_0 language learning project (Feldman *et al.* 1990), a theme of which is to bring together constraints from linguistics, psychology, neurobiology, and computer science, in order to discover underlying universals in cognitive representations. One such constraint derives from crosslinguistic variation. Requiring our models to be capable of learning the relevant portion of *any* natural language without change, guides us toward the proper amount of innate bias—too little, and learning will be too slow; too much, and we won't be able to represent some languages.

A second constraint is the hypothesis that semantics are often grounded in the body (Lakoff & Johnson 1980; Johnson 1987). Within the particular domain of hand actions, the hypothesis is that the semantics primarily involve aspects of intentional motor behavior, such as goals, primitive motor synergies, their parameters (such as amount of force), and their coordination. Our current model attempts to capture the essence of motor control within a representation called *executing schemas*, from which verb semantics are derived.

We should make clear that our primary aim is to provide a framework for explaining how the processes of language interact with other processes such as motor control. It is unlikely that this model of verb semantics will be either complete or exactly correct. Instead, we hope the model serves to suggest how the larger picture of language use partially determines the *nature* of verb semantics.

At the end of the paper, this work will be set within a larger context by connecting it to other L_0 projects modelling the semantics of spatial terms, the semantics of auxiliary and force-dynamic verbs, and the understanding of discourse and metaphor.

2 Computational Model

Figure 1 shows the top-level view of the model. Essentially, the model consists of separate components for verbs and for motor actions, connected by a bi-directional mapping—the upward direction performs recognition, while the downward direction carries out commands. These two components have very different structures:

- The motor system involves structures whose fundamental character is that they are *executing*. In other words, they are controllers, and their semantic contribution comes from their *activity*, rather than their structure.
- Language involves structures that are fundamentally *descriptive*, in other words passive. Thus, they are operated upon by external processes, e.g. in order to be

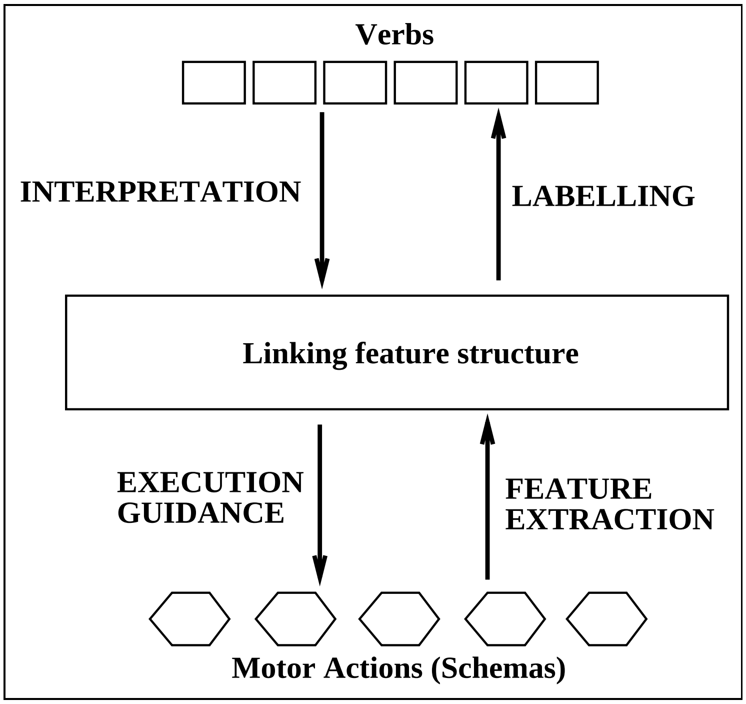

Figure 1: Top-level view of the model, highlighting the processes involved.

composed to represent phrases. Passive representations are also used in other kinds of reasoning and are the mainstay of standard Artificial Intelligence techniques.

As a result, a key design point is how the model *links* these two components. Figure 2 shows the details of our solution, in the form of an example. The next two sections discuss the data structures and their interconnections. Later sections describe the processes which operate on these structures, including the learning algorithm. The reader will want to refer back to Figure 1 and Figure 2 while reading these sections.

2.1 Components

2.1.1 Executing components

The executing components in our model are called *executing schemas* (*schemas* for short), and they are drawn in hexagonal boxes. Schemas are meant to (crudely) capture the character of motor synergies (Bernstein 1967), as well as Piaget's motor schemas (see Drescher (1991) for a computational version), and can be considered an analogue of visual image schemas. Essentially, executing schemas capture the coordination of primitive movements into organized higher-level actions, by encoding sequentiality, concurrency, repetition, hierarchicality, parameterization, and error recovery.

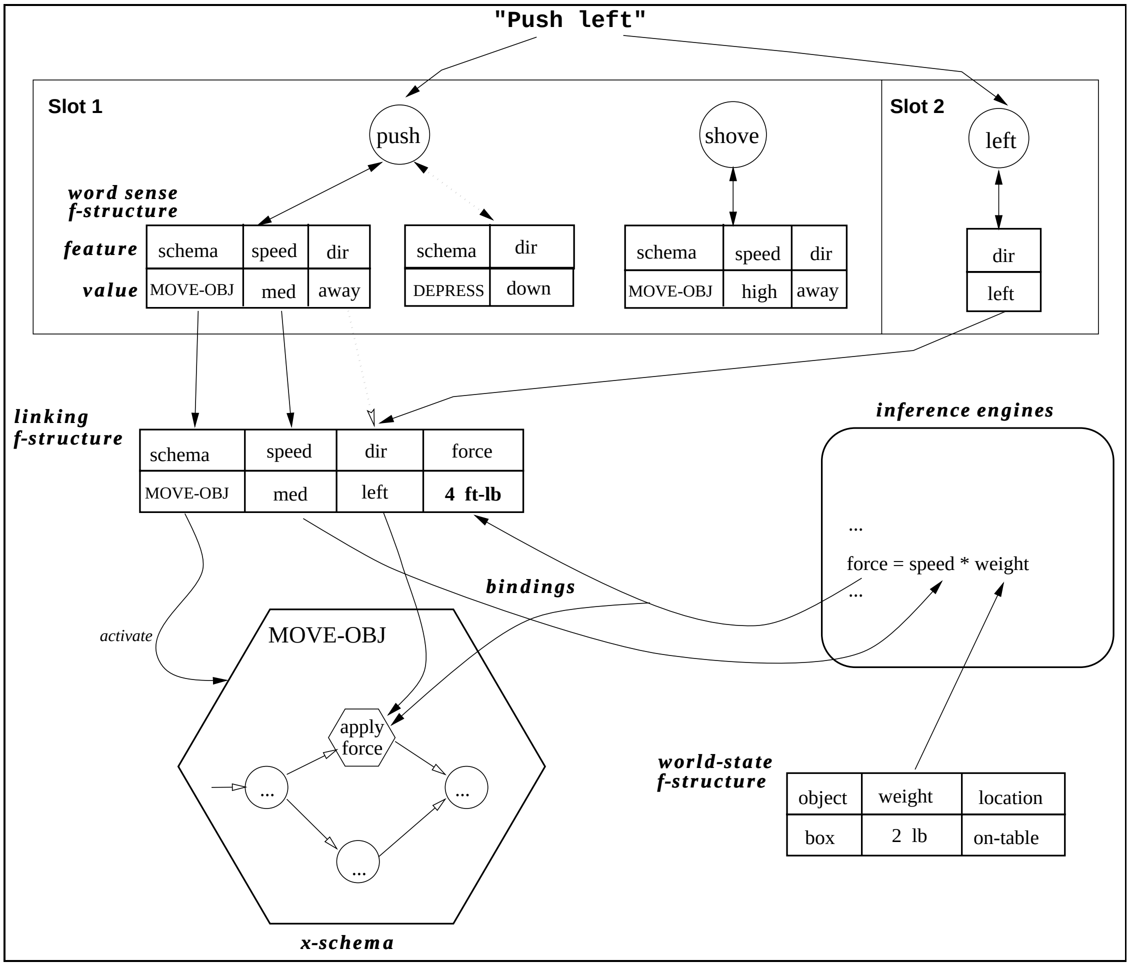

Figure 2: A more detailed view of the model, highlighting data structures and connections involved in carrying out a *push left* command.

Internally, schemas are expressed using a formalism based on Petri nets (Reisig 1985). For the reader unfamiliar with Petri nets, the formalism is like finite state machines, or, more precisely, recursive transition networks, along with extensions to allow multiple "flows of control" which can be initiated either intentionally (to begin multiple simultaneous actions) or by external input (to trigger an error-recovery action).

Note that schemas can both *test* the world state to determine their next action, as well as *affect* the world state via their actions. Consequently, determining a schema's unfolding response to a given input and initial world state is most easily and directly accomplished by simply executing it.

Turning to specifics, our current schema set includes the following. The simplest (i.e., non-decomposable for our purposes) schemas include

GRASP
RELEASE
CONTACT-WITH-PALM
MOVE-ARM

as well as simple calculation schemas such as

COMPUTE-DIRECTION
COMPUTE-FORCE.

From these, complex schemas can be built, such as

MOVE-HAND-TO
GRIP-OBJ
MOVE-HELD-OBJ-TO
OBTAIN-OBJ
RELINQUISH-OBJ
OPPOSE-FORCE
MOVE-OBJ
DEPRESS.

These schemas are generally implementable as a sequence of several primitive schemas, with occasional parallel steps. For an example see the MOVE-OBJ schema in Figure 3.

While an agent may invoke one of the above schemas directly, often the agent has some other, more specific goal in mind. To model this, we have another set of schemas which often simply invoke one of the above schemas, but which serve to indicate the "higher-level" goal. Examples include

INSERT
GET-OBJECT-OUT-OF-WAY

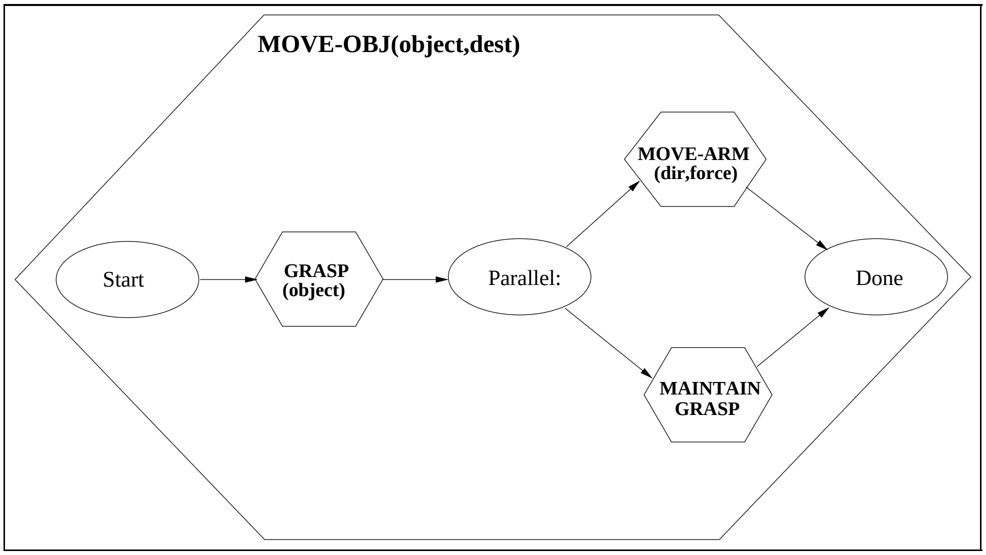

Figure 3: The MOVE-OBJ schema. Note that it is composed of more primitive schemas.

COVER
REMOVE.

2.1.2 Descriptive components

All descriptive components in our model take the form of *feature structures*, or *f-structs* for short. They consist of a fixed set of *features*, each assigned a *value*. F-structs are drawn as a double row of boxes, where each column contains a feature and its corresponding value. Values may be of a variety of types, including boolean, multi-valued, numerical, etc. An f-struct may also assign to a feature a probability distribution over possible values.

F-structs are so named because they play a role similar to feature structures in various linguistic theories. However, it's crucial that our f-structs differ from the conventional variety in the following ways:

1. A relatively large number of features are employed, at a relatively fine granularity.
2. Feature values are not only binary, but also multi-valued, numerical, or probabilistic.
3. Our feature set is bodily grounded via its connection to a model of the motor control system.

Of course, feature-structures are a fairly general and common representation, and that is a strength. Any number of traditional Artificial Intelligence reasoning algorithms employ feature structures and thus they can be used to augment our model as necessary in scaling up to richer domains.

Again turning to specifics, a subset of our current feature set is shown below. The goal is to include only those features which are linguistically relevant, and in this regard we have been motivated by Talmy (1985) and our own informal crosslinguistic analyses of English, Tamil, Cantonese, and Farsi. Importantly, this set is just a starting point, subject to inevitable revision and extension as the model is implemented and tested:

Top-level schema (encodes agent's goal)
Set of sub-schemas executed
Direction of motion or force
Force on finger muscles
Force on arm muscles
Initial/final supportedness of object
Repetition count for looped schemas
Hand posture (flat palm, 5-finger grip, index-finger-extended, etc.)

2.2 Architecture

2.2.1 The linking f-struct

The central structure of the whole model is the *linking f-struct* pictured in the center of Figure 2. Its importance derives from its function as the sole interface between descriptive components such as language or "reasoning", and the executing schemas. It therefore must contain all features which could ever be linguistically relevant in any language, and thereby makes a strong theoretical claim. (In this project, however, it contains only those features potentially relevant to verbs describing hand actions.) The linking f-struct can be used either to *summarize* important results during execution of a schema, or to *guide* schema execution; both uses are described shortly.

2.2.2 The world-state f-struct

The model includes a *world-state f-struct* to hold information about the current state of the world. Generally it can be considered to hold the output of high-level perceptual processes (which we are not concerned with modelling explicitly). The world-state f-struct is used by schemas during execution in order to choose actions and their parameters. Importantly, it is also used when interpreting a verbal command, in order to choose the most applicable amongst multiple senses of a verb.

2.2.3 Word sense f-structs

A *word-sense f-struct* represents the semantics of a word. While it may contain some features which are assigned a simple value, it will generally contain other features which are assigned a probability distribution over possible values. Moreover, a word sense may simply omit those features which have proven to be irrelevant. Note that by taking the mode from each probabilistically defined feature, one obtains what we might call a prototype for the word. Since some words may be most usefully thought of as having multiple senses/prototypes, the model allows multiple word-sense f-structs for each word.

2.2.4 Slots

The complexity of the upper, linguistically-oriented portion of Figure 2 stems from our desire to handle not only simple verbs, but full *verb complexes*—that is, verb roots along with auxiliary verbs, affixes, and satellites (in the sense of Talmy (1985)). For example, we would ideally like to handle expressions such as *keep picking up*.

We assume, rather than model, a mechanism to segment such a verb complex into four *slots* containing the morphemes *keep*, *pick*, *-ing*, and *up*, but importantly we do not assume prior knowledge of the semantic roles played by these slots. After learning, each slot in the system contains a node for each word (or morpheme) which has appeared in it. In turn, each node is attached to its word-sense f-structs. In Figure 2, two slots are shown as large boxes, and word nodes are shown as circles.

While the model does not rely upon prior knowledge of the semantic roles played by slots, we certainly do intend to model *learning* such information as a by-product of word learning, and using it to facilitate learning of new words.

2.3 Processes

The four arrows in Figure 1 represent the four key processes underlying language use once the system is trained. The two upward arrows—feature extraction and labelling—are active when the system is producing an appropriate verb complex for its own actions. The two downward arrows—interpretation and execution guidance—are active when the system is carrying out a verbal command. The learning algorithm is saved for the next section.

2.3.1 Feature extraction

A schema, while executing, may cause certain values to be placed into the linking f-struct. These values may reflect states encountered during execution, the parameters of various actions, or flow-of-control patterns such as looping, and may represent averages over the whole schema execution, or just the value at a particularly salient moment. As a result of feature extraction, the linking f-struct contains a "summary" of a schema's execution once it has completed.

2.3.2 Labelling

The labelling process involves matching the current contents of the linking f-struct against the store of word-sense f-structs. In the case of simple verbs, we simply choose the best-fitting word-sense and emit the corresponding word, where the degree of fit is a function of both the prior probability of the word-sense (i.e., its frequency in the training set) and the likelihood of the word-sense generating the linking features. In the case of verb complexes, a more complex process is required, which chooses words to fill slots until all salient linking features have been communicated by one slot or another.

2.3.3 Interpretation

Interpretation is the process of mapping from linguistic input to a setting for the linking f-structure. This involves choosing senses for each word in the input and then combining them, in a way that minimizes conflicts amongst the words senses and the initial world state. Our current algorithm is a simple one related to unification—its adequacy remains to be tested.

2.3.4 Execution guidance

Once the linking f-struct is set, it should guide schema execution appropriately. This involves choosing the proper schema to execute, transferring values down to parameters of actions (where they may fully specify, or merely modulate, those parameters), and influencing branching decisions.

3 Learning

The real strength of our representation is that it lends itself to the use of powerful learning techniques. The learning process involves three activities:

1. formation of word-sense f-structs as appropriate
2. determination of which features to include in each word-sense f-struct
3. finding the proper value or probability distribution for each included feature.

Our current algorithm is Bayesian model-merging (Omohundro 1992), a common and well tested probabilistic clustering technique. A key feature of this algorithm is its explicit, tunable mechanism for striking a balance between generating too many, very specific senses of a word, and generating too few, excessively general senses. Another advantage is that it is on-line; in other words, sensible results emerge after just one training example. The algorithm also provides a means for biasing—but not forcing—the representation of a new word toward those features which have proven to be relevant for other words in the same slot.

A pseudo-code version of the algorithm follows; the reader may wish to skip ahead to the example in the next section.

```
For each word:
    1.  Collect one or more training-example linking f-structs for the word.
    2.  Create a word-sense for each, by turning each feature value
        into a probability distribution (using priors for word's slot).
    3.  Add these word-senses to the existing set of word-senses.
    4.  LOOP
          Set (ws1,ws2) = the best-matched pair of word senses
          IF (merging ws1 and ws2 would yield a higher a posteriori model)
          THEN Delete ws1 and ws2; Insert Merge(ws1,ws2)
          ELSE terminate loop
        END
    5.  Go back to step 1.
```

3.1 A Learning Example

Let us suppose that there are two senses of the English verb *push*, one of which involves moving an object away from oneself, usually using the palm of the hand, and the other

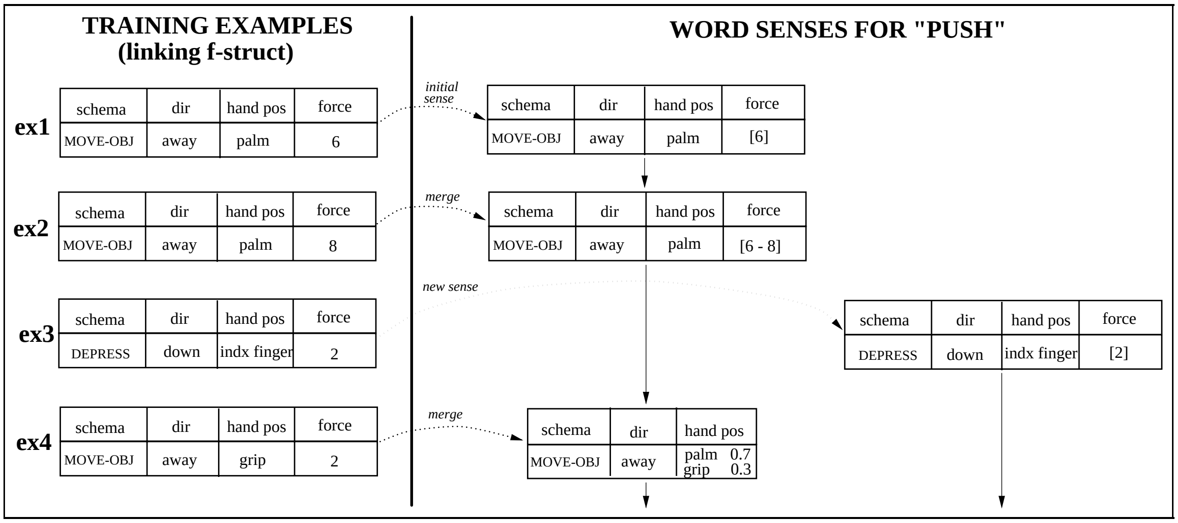
TRAINING EXAMPLES
(linking f-struct)
WORD SENSES FOR "PUSH"
ex1
schema | dir | hand pos | force
MOVE-OBJ | away | palm | 6
ex2
schema | dir | hand pos | force
MOVE-OBJ | away | palm | 8
ex3
schema | dir | hand pos | force
DEPRESS | down | indx finger | 2
ex4
schema | dir | hand pos | force
MOVE-OBJ | away | grip | 2
initial sense
schema | dir | hand pos | force
MOVE-OBJ | away | palm | [6]
merge
schema | dir | hand pos | force
MOVE-OBJ | away | palm | [6 - 8]
new sense
schema | dir | hand pos | force
DEPRESS | down | indx finger | [2]
merge
schema | dir | hand pos
MOVE-OBJ | away | palm 0.7 grip 0.3

Figure 4: Step by step view of learning two senses of *push*. Training examples are incorporated into the model from top to bottom.

being the case of pressing a button. How would the learning algorithm synthesize two word-sense f-structs to represent these senses?

For simplicity, let the linking f-struct contain just four features: top-level schema executed, direction of motion, hand posture, and amount of force used. We'll illustrate the algorithm without worrying about how these features are extracted from schema execution.

Figure 4 shows the model's changing representation of *push*, as four training examples—i.e., settings of the linking f-struct—are processed. Initially, a word-sense f-struct is created which closely matches the first training example. The second training example is deemed close enough to this word-sense that it is merged with it, causing the word sense to generalize a bit—note the widened range of acceptable force values. Training example 3 is deemed too different for merging into the current word sense—it differs on every feature. Instead, a new word sense is created closely matching the training example. Finally, training example 4 most closely matches the first word sense, and so is merged with it. At this point, the force values have varied so much within this word sense that force is dropped as a relevant feature.

4 Relevant Cognitive Linguistics Issues

This project is currently in the implementation phase, and so we do not yet have hard results on the adequacy of the model. However, it is worth reviewing here the linguistic issues which the model purports to address, and the kinds of predictions it would make if successful.

4.1 Semantics

We hope we have presented a plausible account of how semantics for action-oriented words may be bodily grounded via a connection to the motor synergies which drive behavior. An important implication of the model is that schema internals—such as joint angles or detailed patterns of muscle contractions–are not available at the linguistic level, leading to the prediction that no language will contain verbs referring to these details.

4.2 Pragmatics

Our model exemplifies the notion that semantics can be simplified by relying on pragmatics. A single word sense may give rise to a variety of appropriate behaviors, because (a) it is combined with the initial world-state f-struct to produce a variety of possible linking f-structs; and (b) the resulting linking f-struct can give rise to yet further variety of behavior, because the changing world state also affects schema execution.

4.3 Radial categories and prototype theory

Our word representation, viewed as a "concept", exhibits both fuzzy boundaries and multiple prototypes, made possible by allowing multiple word senses with probabilistic distributions. However, we haven't modelled the relational structure of the multiple prototypes composing a radial category, nor inferences which might be derivable therefrom.

4.4 Polysemy and compositional semantics

A version of polysemy and compositional semantics follows from our use of multiple word-senses and the Interpretation Process. When given a command, the Interpretation Process chooses a sense for each word of the command so as to minimize conflicts, demonstrating how linguistic context can determine the intended sense of a polysemous word. Then, the combining of the chosen senses into the linking f-struct—including a mechanism for resolving remaining conflicts—demonstrates how compositional meanings may be formed. If correct, these mechanisms predict that one would not find verb-complexes which use any but the least-conflicting senses of their component words.

4.5 Learning language patterns

Children are able to learn semantic patterns in the language they are acquiring (e.g., learning that verbs encode path vs. manner), as documented in Choi & Bowerman (1991). Our model captures this via the learning algorithm's ability to alter the initial assumption of what features are relevant for a new word, based on the relevant features for words previously learned within the same slot. A resulting prediction is that words which violate their language's patterns will take longer to learn.

5 Related L_0 Work

5.1 Semantics of Locatives

In many ways this project is patterned after earlier L_0 work by Regier (1992), in which a system was built which could learn the semantics of spatial terms from a variety of languages. In both projects, the aim is to arrive at universal semantic features by (1) balancing the dual constraints of efficient learning and crosslinguistic applicability; and (2) grounding these features in an idealized, or "schematic" representation. In Regier's case the semantic features are grounded in an idealization of the human visual system, while in our case it's an idealized motor control system.

5.2 Aspect and Force Dynamics

Jonathan Segal is working on a schematic representation for force-dynamic and aspectual terms such as *keep*, *let*, *hinder*, etc. The key idea is to posit a special schema for controlling other schemas as they progress through their stages, such as start, process, suspend, finish, etc. Then, the semantics of the terms of interest can be encoded by features which map to the behavior of this special schema. This work fits in nicely with the framework presented in this paper.

5.3 Metaphor and Understanding Sentences

Will the architecture presented here scale up to handle arbitrary natural language? We think it might, and this is explored in the thesis work of Narayanan (1995). Since our feature structures are a rather conventional type of representation, one can imagine adding any number of standard Artificial Intelligence inference mechanisms to the architecture, in a position where they can inspect and modify the contents of the linking structure and world state structure, but do not have access to the internals of the schemas. In particular, Narayanan is exploring the use of belief networks to model how feats such as disambiguation might result from probabilistic reasoning over the products of schema execution. Furthermore, he is modelling metaphor within these belief nets, so that metaphorical inferences might be explained via execution of the relevant source-domain schemas.

6 Acknowledgments

This paper represents joint work with Jerry Feldman, Srini Narayanan, and Jonathan Segal of the L_0 group at the International Computer Science Institute. Valuable assistance has also been lent by Collin Baker, Ben Gomes, George Lakoff, D. R. Mani, Terry Regier, Dan Slobin, Robert Wilensky, and quite a few others. This work was partially supported by ONR Grant N00014-93-1-1149.

References

Bernstein, N. A. 1967. *The Co-ordination and Regulation of Movement*. New York: Pergamon Press.

Choi, Soonja, & Melissa Bowerman. 1991. Learning to express motion events in English and Korean: The influence of language-specific lexicalization patterns. In *Lexical and Conceptual Semantics*, ed. by Beth Levin & Steven Pinker, 83–122. Amsterdam: Elsevier Science Publishers.

Drescher, Gary L. 1991. *Made-Up Minds: A Constructivist Approach to Artificial Intelligence*. Cambridge, MA: MIT Press.

Feldman, Jerome A., George Lakoff, Andreas Stolcke, & Susan Weber. 1990. Miniature language acquisition: A touchstone for cognitive science. Technical Report TR-90-009, International Computer Science Institute, Berkeley, CA. also in the Proceedings of the 12th Annual Conference of the Cognitive Science Society, pp. 686–693.

Johnson, Mark. 1987. *The Body in the Mind*. The University of Chicago Press.

Lakoff, George, & Mark Johnson. 1980. *Metaphors We Live By*. University of Chicago Press.

NARAYANAN, SRINI, 1995. One more step: An online model of metaphoric interpretation. Ph.D. Thesis Proposal, Computer Science Division, University of California at Berkeley.

OMOHUNDRO, STEPHEN. 1992. Best-first model merging for dynamic learning and recognition. Technical Report TR-92-004, International Computer Science Institute, Berkeley, CA.

REGIER, TERRY, 1992. *The Acquisition of Lexical Semantics for Spatial Terms: A Connectionist Model of Perceptual Categorization.* Computer Science Division, EECS Department, University of California at Berkeley dissertation. Available as Technical Report TR-92-062, International Computer Science Institute, Berkeley.

REISIG, WOLFGANG. 1985. *Petri Nets: An Introduction.* Berlin: Springer–Verlag.

TALMY, LEONARD. 1985. Lexicalization patterns: semantic structure in lexical forms. In *Language typology and syntactic description: grammatical categories and the lexicon*, ed. by Timothy Shopen, chapter 2. Cambridge University Press.