

ACQUIRING CORE MEANINGS OF WORDS, REPRESENTED AS JACKENDOFF-STYLE CONCEPTUAL STRUCTURES, FROM CORRELATED STREAMS OF LINGUISTIC AND NON-LINGUISTIC INPUT

Jeffrey Mark Siskind*
M. I. T. Artificial Intelligence Laboratory
545 Technology Square, Room NE43-800b
Cambridge MA 02139
617/253-5659
internet: Qobi@AI.MIT.EDU

Abstract

This paper describes an operational system which can acquire the core meanings of words without any prior knowledge of either the category or meaning of any words it encounters. The system is given as input, a description of sequences of scenes along with sentences which describe the [EVENTS] taking place as those scenes unfold, and produces as output, a lexicon consisting of the category and meaning of each word in the input, that allows the sentences to describe the [EVENTS]. It is argued, that each of the three main components of the system, the parser, the linker and the inference component, make only linguistically and cognitively plausible assumptions about the innate knowledge needed to support tractable learning. The paper discusses the theory underlying the system, the representations and algorithms used in the implementation, the semantic constraints which support the heuristics necessary to achieve tractable learning, the limitations of the current theory and the implications of this work for language acquisition research.

1 Introduction

Several natural language systems have been reported which learn the meanings of new words[5, 7, 1, 16, 17, 13, 14]. Many of these systems (in particular [5, 7, 1]) learn the new meanings based upon expectations arising from the morphological, syntactic, se-

mantic and pragmatic context of the unknown word in the text being processed. For example, if such a system encounters the sentence “I woke up yesterday, turned off my alarm clock, took a shower, and cooked myself two grimps for breakfast[5]” it might conclude that *grimps* is a noun which represents a type of food. Such systems succeed in learning new words only when the context offers sufficient constraint to narrow down the possible meanings to make the acquisition unambiguous. Accordingly, such a theory accounts only for the type of learning which arises when an adult encounters an unknown word while reading a text comprised mostly of known words. It can not explain the kind of learning which a young child performs during the early stages of language acquisition when it starts out knowing the meanings of few if any words.

In this paper, I present a new theory which can account for the language learning which a child exhibits. In this theory, the learner is presented with a training *session* consisting of a sequence of *scenarios*. Each scenario contains both linguistic and non-linguistic (i.e. visual) information. The non-linguistic information for each scenario consists of a time-ordered sequence of *scenes*, each depicted via a conjunction of true and negated atomic formulas describing that scene. Likewise, the linguistic information for each scenario consists of a time-ordered sequence of sentences. Initially, the learner knows nothing about the words comprising the sentences in the training session, neither their lexical category nor their meaning. From the two correlated sources of input, the linguistic and the non-linguistic, the learner can infer the set of possible lexicons (i.e. the possible

*Supported by an AT&T Bell Laboratories Ph.D. scholarship. Part of this research was performed while the author was visiting Xerox PARC as a research intern and as a consultant.

categories and meanings of the words in the linguistic input) which allow the linguistic input to describe or account for the non-linguistic input. This inference is accomplished by applying a compositional semantics linking rule in reverse and then performing some constraint satisfaction.

This theory has been implemented in a working computer program. The program succeeds and is tractable because of a small number of judicious semantic constraints and a small number of heuristics which order and eliminate much of the search. This paper explains the general theory as well as the implementation details which make it work. In addition, it discusses some limitations in the current theory, among which is one which prevents it from converging on a single definition of some words.

2 Background

In [15], Rayner et. al. describe a system which can determine the lexical category of each word in a corpus of sentences. They observe that while in the original formulation, a definite clause grammar[12] normally defines a two-argument predicate `parser(Sentence, Tree)` with the lexicon represented directly in the clauses of the grammar, an alternative formulation would allow the lexicon to be represented explicitly as an additional argument to the parser relation, yielding a three argument predicate `parser(Sentence, Tree, Lexicon)`. This three argument relation can be used to learn lexical category information by a technique summarized in Figure 1. Here, a query is formed containing a conjunction of calls to the parser, one for each sentence in the corpus. All of the calls share a common `Lexicon`, while in each call, the `Tree` is left unbound. The `Lexicon` is initialized with an entry for each word appearing in the corpus where the lexical category of each such initial entry is left unbound. The purpose of this initial lexicon is to enforce the constraint that each word in the corpus be assigned a unique lexical category. This restriction, the *monosemy constraint*, will play an important role in the work we describe later. The result of issuing the query in the above example is a lexicon, with instantiated lexical categories for each lexical entry, such that with that lexicon, all of the words in the corpus can be parsed. Note that there could be several such lexicons, each produced by backtracking.

In this paper we extend the results of Rayner et. al. to the learning of representations of word meanings in addition to lexical category information. Our theory is implemented in an operational computer

program called MAIMRA.¹ Unlike Rayner et. al.'s system, which is given only a corpus of sentences as input, MAIMRA is given two correlated streams of input, one linguistic and one non-linguistic, the later modeling the visual context in which the former were uttered. This is intended to more closely model the kind of learning exhibited by a child with no prior lexical knowledge. The task faced by MAIMRA is illustrated in Figure 2.

MAIMRA does not attempt to solve the perception problem; both the linguistic and non-linguistic input are presented in symbolic form to MAIMRA. Thus, the session given in Figure 2 would be presented to MAIMRA as the following two input pairs:

$$\left\{ \begin{array}{l} (\text{BE}(\text{cup}, \text{AT}(\text{John})) \wedge \\ \neg \text{BE}(\text{cup}, \text{AT}(\text{Mary}))); \\ (\text{BE}(\text{cup}, \text{AT}(\text{Mary})) \wedge \\ \neg \text{BE}(\text{cup}, \text{AT}(\text{John}))) \\ \textit{The cup slid from John to Mary.} \end{array} \right\}$$

$$\left\{ \begin{array}{l} (\text{BE}(\text{cup}, \text{AT}(\text{Mary})) \wedge \\ \neg \text{BE}(\text{cup}, \text{AT}(\text{Bill}))); \\ (\text{BE}(\text{cup}, \text{AT}(\text{Bill})) \wedge \\ \neg \text{BE}(\text{cup}, \text{AT}(\text{Mary}))) \\ \textit{The cup slid from Mary to Bill.} \end{array} \right\}.$$

MAIMRA attempts to infer both category and meaning information from input such as this.

3 Architecture

MAIMRA operates as a collection of modules which mutually constrain various mental representations. The organization of these modules is illustrated in Figure 3. Conceptually, each of the modules is non-directional; each module simply constrains the values which may appear concurrently on each of its inputs. Thus the parser enforces a relation between a time-ordered sequence of sentences and a corresponding time-ordered sequence of syntactic structures or parse trees which are licensed by the lexical category information from a lexicon. The linker imposes compositional semantics on the parse trees produced by the parser, relating the meanings of individual words found in the lexicon, to the meanings of entire utterances, through the mediation of the syntactic structures consistent with the parser. Finally, the inference component relates a time-ordered sequence of observations from the non-linguistic input, to a time-ordered sequence of semantic structures which in some sense explain the non-linguistic input. The non-directional collection of modules can

¹ MAIMRA, or מַימְרָא, is the Aramaic word for *word*.

```

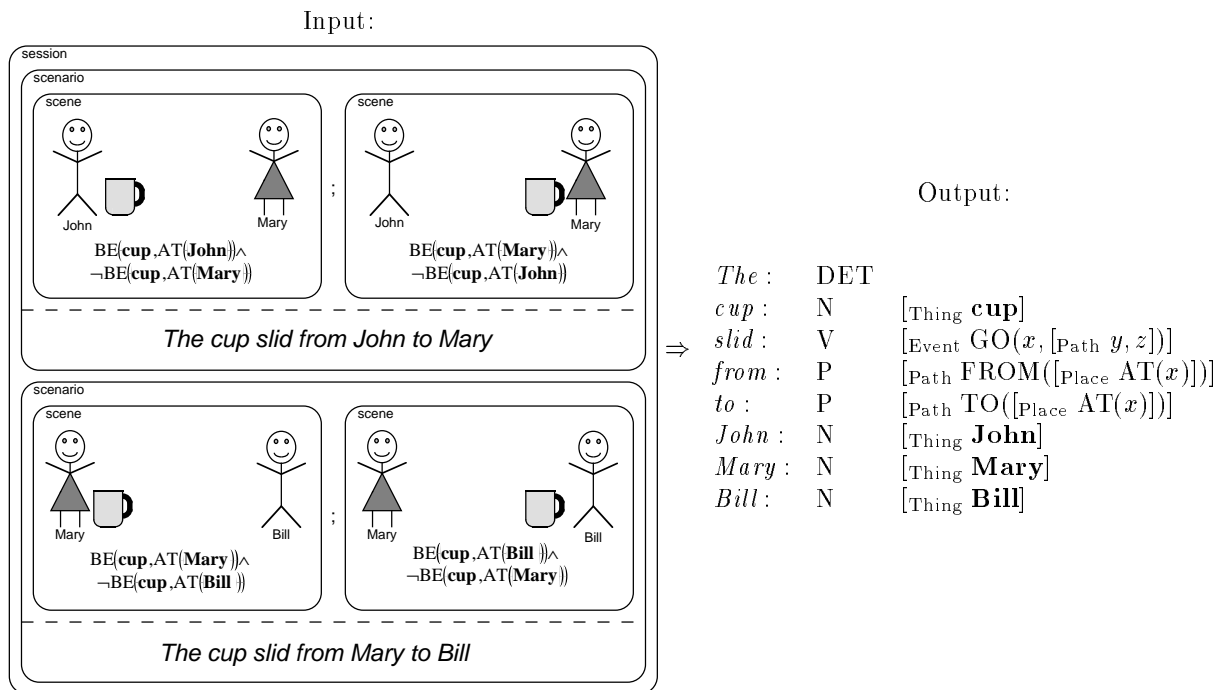
?- Lexicon = [entry(the,_),
              entry(cup,_),
              entry(slid,_),
              entry(from,_),
              entry(john,_),
              entry(to,_),
              entry(mary,_),
              entry(bill,_)],
   parser([the,cup,slid,from,john,to,mary],_,Lexicon),
   parser([the,cup,slid,from,mary,to,bill],_,Lexicon),
   parser([the,cup,slid,from,bill,to,john],_,Lexicon).

```

\Rightarrow

Lexicon = [entry(the,det),
 entry(cup,n),
 entry(slid,v),
 entry(from,p),
 entry(john,n),
 entry(to,p),
 entry(mary,n),
 entry(bill,n)].

Figure 1: The technique used by Rayner et. al. in [15] to acquire lexical category information from a corpus of sentences.


 \Rightarrow

Output:

The : DET

cup : N [Thing **cup**]

slid : V [Event GO(*x*, [Path *y*, *z*]])]

from : P [Path FROM([Place AT(*x*)])]

to : P [Path TO([Place AT(*x*)])]

John : N [Thing **John**]

Mary : N [Thing **Mary**]

Bill : N [Thing **Bill**]

Figure 2: A sample learning session with MAIMRA. MAIMRA is given the two scenarios as input. Each scenario comprises linguistic information, in the form of a sequence of sentences, and non-linguistic information. The non-linguistic information is a sequence of conceptual structure [STATE] descriptions which describe a sequence of visual scenes. MAIMRA produces as output, a lexicon which allows the linguistic input to explain the non-linguistic input.

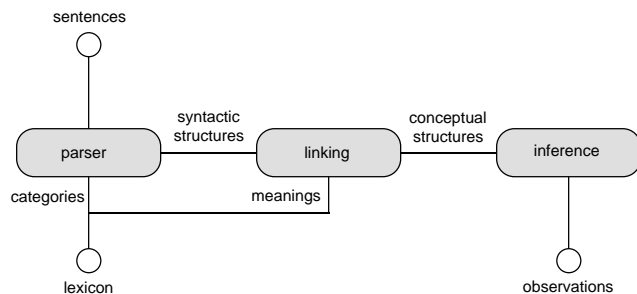


Figure 3: The cognitive architecture used by MAIMRA.

be used in three ways. Given a lexicon and a sequence of sentences as input, the architecture could produce as output, a sequence of observations which are predicted by the sentences. This corresponds to language understanding. Likewise, given a lexicon and a sequence of observations as input, the architecture could produce as output, a sequence of sentences which explain the observations. This corresponds to language generation. Finally, given a sequence of observations and a sequence of sentences as input, the architecture could produce as output, a lexicon which allows the sentences to explain the observations. This last alternative, corresponding to language acquisition, is what interests us here.

Of the five mental representations used by MAIMRA, only three are externally visible, namely the linguistic input, the non-linguistic input and the lexicon. Syntactic and semantic structures exist only internal to MAIMRA and are not externally visible. When using the cognitive architecture from Figure 3 for learning, the values of two of the mental representations, namely the sentences and the observations, are deterministic, since they are fixed as input. The remaining three representations may be nondeterministic; there may be multiple lexicons, syntactic structure sequences and semantic structure sequences which are consistent with the fixed input. In general, each of the three modules alone provides only limited constraint on the possible values for each of the mental representations. Thus taken alone, significant nondeterminism is introduced by each module in isolation. Taken together however, the modules offer much greater constraint on the mutually consistent values for the mental representations, thus reducing the amount of nondeterminism. Much of the success of MAIMRA hinges on efficient ways of representing this nondeterminism.

Conceptually, MAIMRA could have been imple-

mented using techniques similar to Rayner et. al.'s system. Such a naive implementation would directly reflect the architecture given in Figure 3 and is illustrated in Figure 4. The predicate `maimra` would represent the conjunction of constraints introduced by the `parser`, `linker` and `inference` modules, ultimately constraining the mutually consistent values for sentence and observation sequences and the lexicon. Learning a lexicon would be accomplished by forming a conjunction of queries to `maimra`, one for each scenario, where a single `Lexicon` is shared among the conjoined queries. This lexicon is a list of lexical entries, each of the form `entry(Word,Category,Meaning)`. The monosemy constraint is enforced by initializing the `Lexicon` to contain a single entry for each word, each entry having unbound `Category` and `Meaning` slots. The result of processing such a query would be bindings for those `Category` and `Meaning` slots which allow the `Sentences` to explain the `Observations`.

The naive implementation is too inefficient to be practical. This inefficiency results from two sources: inefficient representation of nondeterministic values and non-directional computation. Nondeterministic mental representations are expressed in the naive implementation via backtracking. Expressing nondeterminism this way requires that substructure shared across different alternatives for a mental representation be multiplied out. For example, if MAIMRA is given as input, a sequence of two sentences $S_1; S_2$, where the first sentence has n parses and the second m parses, then there would be $m \times n$ distinct values for the parse tree sequence produced by the parser for this sentence sequence. Each such parse tree sequence would be represented as a distinct backtrack possibility by the naive implementation. The actual implementation instead represents this nondeterminism explicitly as AND/OR trees and additionally factors out much of the shared common substructure to reduce the size of the mental representations and the time needed to process them. As noted previously, the individual modules themselves offer little constraint on the mental representations. A given sentence sequence corresponds to many parse tree sequences which in turn corresponds to an even greater number of semantic structure sequences. Most of these are filtered out, only at the end by the inference component, because they do not correspond to the non-linguistic input. Rather than have these modules operate as non-directed sets of constraints, direction-specific algorithms are used which are tailored to producing the factored mental representations in an efficient order. First, the inference component is called to produce all semantic

```

maimra(Sentences, Lexicon, Observations):-
  parser(Sentences, SyntacticStructures, Lexicon),
  linker(Trees, ConceptualStructures, Lexicon),
  inference(ConceptualStructures, Observations).
?- Lexicon = [entry(the,_,_),
              entry(cup,_,_),
              entry(slid,_,_),
              entry(from,_,_),
              entry(john,_,_),
              entry(to,_,_),
              entry(mary,_,_),
              entry(bill,_,_)],
  maimra([[the,cup,slid,from,john,to,mary]],
         Lexicon,
         be(cup,at(john))&~be(cup(at(mary))),
         be(cup,at(mary))&~be(cup(at(john)))),
  maimra([[the,cup,slid,from,mary,to,bill]],
         Lexicon,
         be(cup,at(mary))&~be(cup(at(bill))),
         be(cup,at(bill))&~be(cup(at(mary)))).

```

⇒

```

Lexicon = [entry(the,det,noSemantics),
           entry(cup,n,cup),
           entry(slid,v,go(x,[from(y),to(z)])),
           entry(from,p,at(x)),
           entry(john,n,john),
           entry(to,p,at(x)),
           entry(mary,n,mary),
           entry(bill,n,bill)].

```

Figure 4: A naive implementation of the cognitive architecture from Figure 3 using techniques similar to those used by Rayner et. al. in [15].

structure sequences which correspond to the observation sequence. Then, the parser is called to produce all syntactic structure sequences which correspond to the sentence sequence. Finally, the linking component is run in reverse to produce meanings of lexical items by correlating the syntactic and semantic structure sequences previously produced. The details of the factored representation, and the algorithms used to create it, will be discussed in Section 5.

Several of the mental representations used by MAIMRA require a method for representing semantic information. We have chosen Jackendoff's theory of *conceptual structure*, presented in [6], as our model for semantic representation. It should be stressed that although we represent conceptual structure via a decomposition into primitives much in the same way as does Schank[18], unlike both Schank and Jackendoff, we do not claim that any particular such decompositional theory is adequate as a basis for expressing the entire range of human thought and the meanings of even most words in the lexicon. Clearly, much of human experience is well beyond formalization within the current state of the art in knowledge representation. We are only concerned with representing and learning the meanings of words describing simple spatial movements of objects within the visual field of the learner. For this limited task, a primitive decompositional theory such as Jackendoff's seems adequate.

Conceptual structures appear within three of the mental representations used by MAIMRA. First, the semantic structures produced by the linker, as meanings of entire utterances, are represented as either conceptual structure [STATE] or [EVENT] descriptions. Second, the observation sequence comprising the non-linguistic input is represented as a conjunction of true and negated [STATE] descriptions. Only [STATE] descriptions appear in the observation sequence. It is the function of the inference component to infer the possible [EVENT] descriptions which account for the observed [STATE] sequences. Finally, meaning components of lexical entries are represented as fragments of conceptual structure which contain variables. The conceptual structure fragments are combined by the linker, filling in the variables with other fragments, to produce the variable free conceptual structures representing the meanings of whole utterances from the meanings of their constituent words.

4 Learning Constraints

Each of the three modules implements some linguistic or cognitive theory, and accordingly, makes some assumptions about what knowledge is innate and what can be learned. Additionally, each module currently implements only a simple theory and thus has limitations on the linguistic and cognitive phenom-

ena that it can account for. This section discusses the innateness assumptions and limitations of each module in greater detail.

4.1 The Parser

While MAIMRA can learn lexical category information required by the parser, the parser is given a fixed context-free grammar which is assumed to be innate. This fixed grammar used by MAIMRA is shown in Figure 5. At first glance it might seem unreasonable to assume that the grammar given in Figure 5 is innate. A closer look however, reveals that the particular context-free grammar we use is not entirely arbitrary; it is motivated by \bar{X} -theory[2, 3] which many linguists take to be innate. Our grammar can be derived from \bar{X} -theory as follows. We start with a version of \bar{X} -theory which allows non-binary branching nodes and where maximal projections carry bar-level one (i.e. XP is \bar{X}). First, fix the parameters HEAD-first and SPEC-first to yield the prototype rule:

$$XP \rightarrow \{X_{\text{SPEC}}\} X \text{ complement}^*.$$

Second, instantiate this rule for each of the lexical categories N, V and P viewing N_{SPEC} as DET, V_{SPEC} as AUX and making P_{SPEC} degenerate. Third, add the rules for S and \bar{S} stipulating that \bar{S} is a maximal projection.² Fourth, declare all maximal projections to be valid complements. Finally, add in the derivation for the English auxiliary system. Thus, our particular context-free grammar is little more than instantiating \bar{X} -theory with the English lexical categories N, V and P, the English parameters HEAD-first and SPEC-first and the English auxiliary system.

We make no claim that the syntactic theory implemented by MAIMRA is complete. Many linguistic phenomena remain unaccounted for in our grammar, among them agreement, tense, aspect, adjectives, adverbs, negation, coordination, quantifiers, *wh*-words, pronouns, reference and demonstratives. While the grammar is motivated by GB theory, the only components of GB theory which have been implemented are \bar{X} -theory and θ -theory. (θ -theory is enforced via the linking rule discussed in the next subsection.) Although future work may increase the scope and accuracy of the syntactic theory incorporated into MAIMRA, even the current limited grammar offers a sufficiently rich framework for investigating language acquisition. It's most severe limitation is a lack of subcategorization; the grammar allows nouns,

²A more principled way of deriving the rules for S and \bar{S} from \bar{X} -theory is given in [4]

$$\begin{aligned} S &\rightarrow \text{NP } \boxed{\text{VP}} \\ \bar{S} &\rightarrow \{\text{COMP}\} \boxed{\text{S}} \\ \text{NP} &\rightarrow \{\text{DET}\} \boxed{\text{N}} \{\bar{S}|\text{NP}|\text{VP}|\text{PP}\}^* \\ \text{VP} &\rightarrow \{\text{AUX}\} \boxed{\text{V}} \{\bar{S}|\text{NP}|\text{VP}|\text{PP}\}^* \\ \text{PP} &\rightarrow \boxed{\text{P}} \{\bar{S}|\text{NP}|\text{VP}|\text{PP}\}^* \\ \text{AUX} &\rightarrow \{\text{DO}|\text{BE}|\{\text{MODAL}|\text{TO}|\{\{\text{MODAL}|\text{TO}\}\} \text{HAVE}\} \{\text{BE}\}\} \end{aligned}$$

Figure 5: The context free grammar used by MAIMRA. This grammar is motivated by \bar{X} -theory. The head of each rule is enclosed in a box. This head information is used by the linker.

verbs and prepositions to take any number of complements of any kind. This causes the grammar to severely overgenerate and results in a high degree of non-determinism in the representation of syntactic structure. It is interesting that despite the use of a highly ambiguous grammar, the combination of the parser with the linker and inference component, together with the non-linguistic context, provide sufficient constraint for the system to learn words quickly with few training scenarios. This gives evidence that many of the constraints normally assumed to be imposed by syntax, actually result from the interplay of multiple modules in a broad cognitive system.

4.2 The Linker

The linking component of MAIMRA implements a single linking rule which is assumed to be innate. This rule is best illustrated by way of the example given in Figure 6. Linking proceeds in a bottom up fashion from the leaves of the parse tree towards its root. Each node in the parse tree is annotated with a fragment of conceptual structure. The annotation of leaf nodes comes from the meaning entry for that word in the lexicon. Every non-leaf node has a distinguished daughter called the *head*. Knowledge of which daughter node is the head for any given phrasal category is assumed to be innate. For the grammar used by MAIMRA, this information is indicated in Figure 5 by the categories enclosed in boxes. The annotation of a non-leaf node is formed by copying the annotation of its head daughter node, which may contain variables, and filling some of its variable slots with the annotation of the remaining non-head daughters. Note that this is a nondeterministic process; there is no stipulation of which variables get

linked to which complements. Because of this non-determinism, there can be many linkings associated with any given lexicon and parse tree. In addition to this linking ambiguity, existence of multiple lexical entries with different meanings for the same word can cause meaning ambiguity.

A given variable may appear multiple times in a fragment of conceptual structure. The linking rule stipulates that when a variable is linked to an argument, all instances of the same variable get linked to that argument as well. Additionally, the linking rule maintains the constraint that the annotation of the root node, as well as any node which is a sister to a head, must be variable free. Linkings which violate this constraint are discarded. There must be at least as many distinct variables in the conceptual structure annotating the head as there are sisters of the head. Again, if there are insufficient variables in the head the partial linking is discarded. There may be more, however, which means that the annotation of the parent will contain variables. This is acceptable if the parent is not itself a sister to a head.

MAIMRA imposes two additional constraints on the linking process. First, meanings of lexical items must have some semantic content; they can not be simply a variable. Second, the functor of a conceptual structure fragment can not be a variable. In other words, it is not possible to have a fragment $\text{FROM}(x(\text{John}))$ which would link with AT to produce $\text{FROM}(\text{AT}(\text{John}))$. These constraints help reduce the space of possible lexicons and support search pruning heuristics which make learning faster.

In summary, the linking component makes use of six pieces of knowledge which are assumed to be innate.

1. The linking rule.
2. The head category associated with each phrasal category.
3. The requirement that the root semantic structure be variable free.
4. The requirement that conceptual structure fragments associated with sisters of heads be variable free.
5. The requirement that no lexical item have empty semantics.
6. The requirement that no conceptual structure fragment contain variable functors.

There are at least two limitations in the theory of linking discussed above. First, there is no attempt to

give an adequate semantics for the categories DET, AUX and COMP. Currently, the linker assumes that nodes labeled with these categories have no conceptual structure annotation. Furthermore, DET, AUX and COMP nodes which are sisters to a head are not linked to any variable in the conceptual structure annotating the head. Second, while the above linking rule can account for predication, it cannot account for the semantics of adjuncts. This shortcoming results not just from limitations in the linking rule but also from the fact that Jackendoff's conceptual structure is unable to represent adjunct information.

4.3 The Inference Component

The inference component imposes the constraint that the linguistic input must "explain" the non-linguistic input. This notion of explanation is assumed to be innate and comprises four principles. First, each sentence must describe some subsequence of scenes. Everything the teacher says must be true in the current non-linguistic context of the learner. The teacher cannot say something which is either false or unrelated to the visual field of the learner. Second, while the teacher is constrained to making only true statements about the visual field of the learner, the teacher is not required to state everything which is true; some non-linguistic data may go undescribed. Third, the order of the linguistic description must match the order of occurrence of the non-linguistic [EVENTS]. This is necessary because the language fragment handled by MAIMRA does not support tense and aspect. It also adds substantial constraint to the learning process. Finally, sentences must describe non-overlapping scene sequences. Of these principles, the first two seem very reasonable. The third is in accordance with the evidence that children acquire tense and aspect later in the language learning process. Only the fourth principle is questionable. The motivation for the fourth principle is that it enables the use of the inference algorithm discussed in Section 5. More recent work, beyond the scope of this paper, suggests using a different inference algorithm which does not require this principle.

The above four learning principles make use of the notion of a sentence "describing" a sequence of scenes. The notion of description is expressed via the set of inference rules given in Figure 7. Each rule enables the inference of the [EVENT] or [STATE] description on its right hand side from a sequence of [STATE] descriptions which match the pattern on its left hand side. For example, Rule 1 states that if there is a sequence of scenes which can be divided into two concatenated subsequences of scenes, such

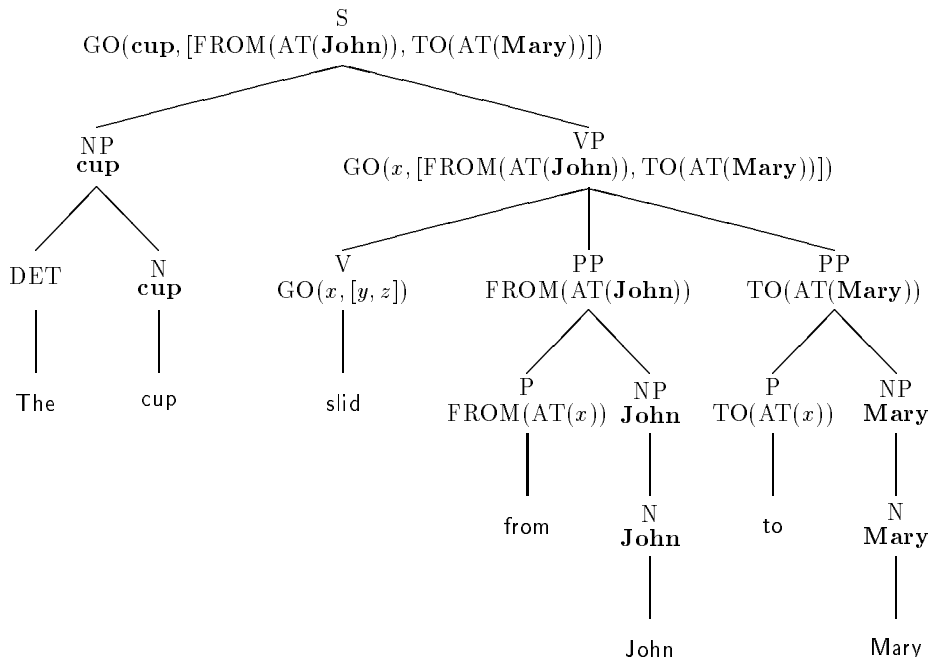


Figure 6: An example of the linking rule used by MAIMRA showing the derivation of conceptual structure for the sentence *The cup slid from John to Mary* from the conceptual structure meanings of the individual words, along with a syntactic structure for the sentence.

that each subsequence contains at least one scene, and in every scene in that first subsequence, x is at y and not at z , while in every scene in the second subsequence, x is at z but not at y , then we can describe that entire sequence of scenes by saying that x went on a path from y to z . This rule does not stipulate that other things can't be true in those scenes embodying an [EVENT] of type GO, just that at a minimum, the conditions on the right hand side must hold over that scene sequence. In general, any given observation may entail multiple descriptions, each describing some subsequence of scenes which may overlap with other descriptions.

MAIMRA currently assumes that these inference rules are innate. This seems tenable as these rules are very low level and are probably implemented by the vision system. Nonetheless, current work is focusing on removing the innateness requirement of these rules from the inference component.

One severe limitation of the current set of inference rules is the lack of rules for describing the causality incorporated in the CAUSE and LET primitive conceptual functions. One method we have considered is to use rules like:

$$\text{CAUSE}(w, \text{GO}(x, [\text{FROM}(y), \text{TO}(z)])) \leftarrow$$

$$(\text{BE}(w, y) \wedge \text{BE}(x, y) \wedge \neg \text{BE}(x, z))^+;$$

$$(\text{BE}(x, z) \wedge \neg \text{BE}(x, y))^+.$$

This states that w caused x to move from y to z if w was at the same location y , as x was, at the start of the motion. This is clearly unsatisfactory. One would like to incorporate a more accurate notion of causality such as that discussed in [9]. Unfortunately, it seems that Jackendoff's conceptual structures are not expressive enough to support the more complex notions of causality. This is another area for future work.

5 Implementation

As mentioned previously, MAIMRA uses directed algorithms, rather than non-directed constraint processing, to produce a lexicon. When processing a scenario, MAIMRA first applies the inference component to the non-linguistic input to produce semantic structures. Then, it applies the parser to the linguistic input to produce syntactic structures. Finally, it applies the linking component in reverse, to both the syntactic structures and semantic structures, to produce a lexicon as output. This process is best

$$\begin{aligned}
\text{GO}(x, [\text{FROM}(y), \text{TO}(z)]) &\leftarrow (\text{BE}(x, y) \wedge \neg \text{BE}(x, z))^+; (\text{BE}(x, z) \wedge \neg \text{BE}(x, y))^+ & (1) \\
\text{GO}(x, \text{FROM}(y)) &\leftarrow (\text{BE}(x, y) \wedge \neg \text{BE}(x, z))^+; (\text{BE}(x, z) \wedge \neg \text{BE}(x, y))^+ & (2) \\
\text{GO}(x, \text{TO}(z)) &\leftarrow (\text{BE}(x, y) \wedge \neg \text{BE}(x, z))^+; (\text{BE}(x, z) \wedge \neg \text{BE}(x, y))^+ & (3) \\
\text{GO}(x, []) &\leftarrow (\text{BE}(x, y) \wedge \neg \text{BE}(x, z))^+; (\text{BE}(x, z) \wedge \neg \text{BE}(x, y))^+ & (4) \\
\text{STAY}(x, y) &\leftarrow \text{BE}(x, y); (\text{BE}(x, y))^+ & (5) \\
\text{STAY}(x, []) &\leftarrow \text{BE}(x, y); (\text{BE}(x, y))^+ & (6) \\
\text{GO}_{\text{Ext}}(x, [\text{FROM}(y), \text{TO}(z)]) &\leftarrow (\text{BE}(x, y) \wedge \text{BE}(x, z) \wedge y \neq z)^+ & (7) \\
\text{GO}_{\text{Ext}}(x, \text{FROM}(y)) &\leftarrow (\text{BE}(x, y) \wedge \text{BE}(x, z) \wedge y \neq z)^+ & (8) \\
\text{GO}_{\text{Ext}}(x, \text{TO}(z)) &\leftarrow (\text{BE}(x, y) \wedge \text{BE}(x, z) \wedge y \neq z)^+ & (9) \\
\text{BE}(x, y) &\leftarrow \text{BE}(x, y)^+ & (10) \\
\text{ORIENT}(x, [\text{FROM}(y), \text{TO}(z)]) &\leftarrow \text{ORIENT}(x, [\text{FROM}(y), \text{TO}(z)])^+ & (11) \\
\text{ORIENT}(x, \text{FROM}(y)) &\leftarrow (\text{ORIENT}(x, [\text{FROM}(y), \text{TO}(z)]) \vee \text{ORIENT}(x, \text{FROM}(y)))^+ & (12) \\
\text{ORIENT}(x, \text{TO}(y)) &\leftarrow (\text{ORIENT}(x, [\text{FROM}(y), \text{TO}(z)]) \vee \text{ORIENT}(x, \text{TO}(y)))^+ & (13)
\end{aligned}$$

Figure 7: The inference rules used by the inference component of MAIMRA to infer [EVENTS] from [STATES].

illustrated by way of an example.

Consider the following input scenario.

$$\left(\begin{array}{l}
(\text{BE}(\text{cup}, \text{AT}(\text{John}))); \\
(\text{BE}(\text{cup}, \text{AT}(\text{Mary})) \wedge \\
\neg \text{BE}(\text{cup}, \text{AT}(\text{John}))); \\
(\text{BE}(\text{cup}, \text{AT}(\text{Mary}))); \\
(\text{BE}(\text{cup}, \text{AT}(\text{Bill})) \wedge \\
\neg \text{BE}(\text{cup}, \text{AT}(\text{Mary}))); \\
\textit{The cup slid from John to Mary.}; \\
\textit{The cup slid from Mary to Bill.}
\end{array} \right)$$

This scenario contains four scenes and two sentences. First, frame axioms are applied to the scene sequence, yielding a sequence of scene descriptions containing all of the true [STATE] descriptions pertaining to those scenes, and only those true [STATE] descriptions.

$$\begin{array}{l}
\text{BE}(\text{cup}, \text{AT}(\text{John})); \\
\text{BE}(\text{cup}, \text{AT}(\text{Mary})); \\
\text{BE}(\text{cup}, \text{AT}(\text{Mary})); \\
\text{BE}(\text{cup}, \text{AT}(\text{Bill}))
\end{array}$$

Given a scenario with n sentences and m scenes, find all possible ways of partitioning the m scenes into sequences of n partitions, where the partitions each contain a contiguous subsequence of scenes, but where the partitions themselves do not overlap and need not be contiguous. If we abbreviate the above sequence of four scenes as $a; b; c; d$, then partitioning for a scenario containing two sentences produces the

following disjunction:

$$\begin{aligned}
&\{[a]; ([b] \vee [c] \vee [d] \vee [b; c] \vee [c; d] \vee [b; c; d])\} \vee \\
&\{([b] \vee [a; b]); ([c] \vee [d] \vee [c; d])\} \vee \\
&\{([c] \vee [b; c] \vee [a; b; c]); [d]\}.
\end{aligned}$$

Next, apply the inference rules from Figure 7 to each partition in the resulting disjunctive formula, replacing each partition with a disjunction of all [EVENTS] and [STATES] which can describe that partition. For our example, this results in the replacements given in Figure 8.

The disjunction that remains after these replacements describes all possible sequences comprised of two [EVENTS] or [STATES] that can explain the input scene sequence. Notice how non-determinism is managed with a factored representation produced directly by the algorithm.

After the inference component produces the semantic structure sequences corresponding to the non-linguistic input, the parser produces the syntactic structure sequences corresponding to the linguistic input. A variant of the CKY algorithm[8, 19] is used to produce factored parse trees. Finally, the linker is applied in reverse to each corresponding parse-tree/semantic-structure pair.

This inverse linking process is termed *fracturing*. Fracturing is a recursive process applied to a parse tree fragment and a conceptual structure fragment. At each step, the conceptual structure fragment is assigned to the root node of the parse tree fragment. If the root node of the parse tree has n non-head daughters, then compute all possible ways of extracting n variable-free subexpressions from the conceptual

$$\begin{aligned}
[a] &\Rightarrow \text{BE}(\mathbf{cup}, \text{AT}(\mathbf{John})) \\
[b], [c] &\Rightarrow \text{BE}(\mathbf{cup}, \text{AT}(\mathbf{Mary})) \\
[d] &\Rightarrow \text{BE}(\mathbf{cup}, \text{AT}(\mathbf{Bill})) \\
[a; b], [a; b; c] &\Rightarrow (\text{GO}(\mathbf{cup}, [\text{FROM}(\text{AT}(\mathbf{John})), \text{TO}(\text{AT}(\mathbf{Mary}))]) \vee \\
&\quad \text{GO}(\mathbf{cup}, \text{FROM}(\text{AT}(\mathbf{John})))) \vee \\
&\quad \text{GO}(\mathbf{cup}, \text{TO}(\text{AT}(\mathbf{Mary}))) \vee \\
&\quad \text{GO}(\mathbf{cup}, []) \\
[b; c] &\Rightarrow (\text{BE}(\mathbf{cup}, \text{AT}(\mathbf{Mary})) \vee \\
&\quad \text{STAY}(\mathbf{cup}, \text{AT}(\mathbf{Mary}))) \\
[c; d], [b; c; d] &\Rightarrow (\text{GO}(\mathbf{cup}, [\text{FROM}(\text{AT}(\mathbf{Mary})), \text{TO}(\text{AT}(\mathbf{Bill}))]) \vee \\
&\quad \text{GO}(\mathbf{cup}, \text{FROM}(\text{AT}(\mathbf{Mary})))) \vee \\
&\quad \text{GO}(\mathbf{cup}, \text{TO}(\text{AT}(\mathbf{Bill}))) \vee \\
&\quad \text{GO}(\mathbf{cup}, []).
\end{aligned}$$

Figure 8: The replacements resulting from the application of the inference rules from Figure 7 to the example given in the text.

structure fragment and assigning them to the non-head daughters, leaving distinct variables behind as place holders. The residue after subexpression extraction is assigned to the head daughter. Fracturing is applied recursively to the conceptual structures assigned to daughters of the root node of the parse tree fragment, along with their annotations. The results of these recursive calls are then conjoined together. Finally, a disjunction is formed over each possible way of performing the subexpression extraction. This process is illustrated by the following example. Consider fracturing the conceptual structure fragment

$$\text{GO}(x, [\text{FROM}(\text{AT}(\mathbf{John})), \text{TO}(\text{AT}(\mathbf{Mary}))])$$

along with a VP node with a head daughter labeled V and two sister daughters labeled PP. This produces the set of possible extractions shown in Figure 9. The fracturing recursion terminates when a lexical item is fractured. This returns a lexical entry triple comprising the word, its category and a representation of its meaning. The end result of the fracturing process is a monotonic Boolean formula over definition triples which concisely represents the set of all possible lexicons which allow the linguistic input from a scenario to explain the non-linguistic input. Such a factored lexicon (arising when processing a scenario similar to the second scenario of the training session given in Figure 2) is illustrated in Figure 10.

The disjunctive lexicon produced by the fracturing process may contain lexicons which assign more than one meaning to a given word. We incorporate a

monosemy constraint to rule out such lexicons. Conceptually, this is done by converting the factored disjunctive lexicon to disjunctive normal form and removing lexicons which contain more than one lexical entry for the same word. Computationally, a more efficient way of accomplishing the same task is to view the factored disjunctive lexicon as a monotonic Boolean formula Φ whose propositions are lexical entries. We conjoin Φ with all conjunctions of the form $\overline{\alpha_i} \wedge \alpha_j$ where the α_i and α_j are both distinct lexical entries for the same word that appear in Φ . The resulting formula is no longer monotonic. Satisfying assignments for this formula correspond to conjunctive lexicons which meet the monosemy constraint. The satisfying assignments can be found using well known constraint satisfaction techniques such as truth maintenance systems[10, 11]. While the problem of finding satisfying assignments for a Boolean formula (i.e. SAT) is NP-complete, our experience is that in practice, the SAT problems generated by MAIMRA are easy to solve and that the fracturing process of generating the SAT problems takes far more time than actually solving them.

The monosemy constraint may seem a bit restrictive. It can be relaxed somewhat by allowing up to n alternate meanings for a word by conjoining in conjunctions of the form

$$\bigwedge_{j=1}^{n+1} \alpha_{i_j}$$

where each of the α_{i_j} are distinct lexical entries for

GO(x, [y, z])	FROM(AT(John))	TO(AT(Mary))	} <i>disjunction.</i>
GO(x, [y, z])	TO(AT(Mary))	FROM(AT(John))	
GO(x, [FROM(y), z])	AT(John)	TO(AT(Mary))	
GO(x, [FROM(y), z])	TO(AT(Mary))	AT(John)	
GO(x, [FROM(AT(y)), z])	John	TO(AT(Mary))	
GO(x, [FROM(AT(y)), z])	TO(AT(Mary))	John	
GO(x, [y, TO(z)])	FROM(AT(John))	AT(Mary)	
GO(x, [y, TO(z)])	AT(Mary)	FROM(AT(John))	
GO(x, [FROM(y), TO(z)])	AT(John)	AT(Mary)	
GO(x, [FROM(y), TO(z)])	AT(Mary)	AT(John)	
GO(x, [FROM(AT(y)), TO(z)])	John	AT(Mary)	
GO(x, [FROM(AT(y)), TO(z)])	AT(Mary)	John	
GO(x, [y, TO(AT(z))])	FROM(AT(John))	Mary	
GO(x, [y, TO(AT(z))])	Mary	FROM(AT(John))	
GO(x, [FROM(y), TO(AT(z))])	AT(John)	Mary	
GO(x, [FROM(y), TO(AT(z))])	Mary	AT(John)	
GO(x, [FROM(AT(y)), TO(AT(z))])	John	Mary	
GO(x, [FROM(AT(y)), TO(AT(z))])	Mary	John	

⏟ *conjunction*

Figure 9: A recursive step of the fracturing process illustrating all possible subexpression extractions from the conceptual structure fragment given in the text, and their assignments to non-head daughters. The center column contains fragments annotating the first PP while the rightmost column contains fragments annotating the second PP. The leftmost column shows the residue which annotates the head. Each row is one distinct possible extraction.

```

(AND (DEFINITION CUP N CUP)
  (OR (AND (OR (AND (DEFINITION MARY N (AT MARY))
    (DEFINITION TO P (TO ?0)))
    (AND (DEFINITION MARY N MARY)
      (DEFINITION TO P (TO (AT ?0))))))
    (OR (AND (OR (AND (DEFINITION JOHN N (AT JOHN))
      (DEFINITION FROM P (FROM ?0)))
      (AND (DEFINITION JOHN N JOHN)
        (DEFINITION FROM P (FROM (AT ?0))))))
      (DEFINITION SLID V (GO ?0 (PATH ?1 ?2))))
    (AND (DEFINITION JOHN N JOHN)
      (DEFINITION FROM P (AT ?0))
      (DEFINITION SLID V (GO ?0 (PATH ?1 (FROM ?2))))))))
  (AND (DEFINITION MARY N MARY)
    (DEFINITION TO P (AT ?0))
    (OR (AND (OR (AND (DEFINITION JOHN N (AT JOHN))
      (DEFINITION FROM P (FROM ?0)))
      (AND (DEFINITION JOHN N JOHN)
        (DEFINITION FROM P (FROM (AT ?0))))))
      (DEFINITION SLID V (GO ?0 (PATH ?1 (TO ?2))))))
    (AND (DEFINITION JOHN N JOHN)
      (DEFINITION FROM P (AT ?0))
      (DEFINITION SLID V (GO ?0 (PATH (FROM ?1) (TO ?2))))))))))

```

Figure 10: A portion of the disjunctive lexicon which results from processing a scenario similar to the second scenario of the training session given in Figure 2.

the same word that appear in Φ , instead of the pairwise conjunctions used previously.

6 Discussion

When presented with a training session³ much like that given in Figure 2, MAIMRA converges to a unique lexicon within six scenarios and several minutes of CPU time. It is not however, able to converge to a unique meaning for the word *enter* when given scenarios of the form:

$$\left\{ \begin{array}{l} (\text{BE}(\mathbf{John}, \text{AT}(\mathbf{outside})) \wedge \\ \neg \text{BE}(\mathbf{John}, \text{IN}(\mathbf{room}))); \\ (\text{BE}(\mathbf{John}, \text{IN}(\mathbf{room})) \wedge \\ \neg \text{BE}(\mathbf{John}, \text{AT}(\mathbf{outside}))) \\ \text{John entered the room.} \end{array} \right\}.$$

It turns out that there is no way to force MAIMRA to realize that the sentence describes the entire scenario and not just the first or last scene alone. Thus MAIMRA does not rule out the possibility that *enter* might mean “to be somewhere.” The reason MAIMRA is successful with the session from Figure 2 is that the empty semantics constraint rules out associating the sentences with just the first or last scene because the semantic structures representing those scene subsequences have too little semantic material to distribute among the words of the sentence. One way around this problem would be for MAIMRA to attempt to choose the lexicon which maximizes the amount of non-linguistic data which is accounted for. Future work will investigate this issue further.

We make three claims as a result of this work. First, this work demonstrates that the combination of syntactic, semantic and pragmatic modules, each incorporating cognitively plausible innateness assumptions, offers sufficient constraint for learning word meanings with no prior lexical knowledge in the context of non-linguistic input. This offers a general framework for explaining meaning acquisition. Second, appropriate choices of representation and algorithms allow efficient implementation within the general framework. While no claim is being made

³Although not strictly required by either the theory or the implementation, we currently incorporate into the training session given to MAIMRA, an initial lexicon telling it that ‘John,’ ‘Mary’ and ‘Bill’ are nouns, ‘from’ and ‘to’ are prepositions and ‘the’ is a determiner. This is to reduce the combinatorics of generating ambiguous parses. Category information is not given for any other words, nor is meaning information given for any words occurring in the training session. In theory it would be possible to efficiently bootstrap the categories for these words as well, via a longer training session containing a few shorter sentences to constrain the possible categories for these words. We have not done so yet, however.

that children employ the mechanisms described here, they nonetheless can be used to construct useful engineered systems which learn language. The third claim is more bold. Most language acquisition research operates under a tacit assumption that children acquire individual pieces of knowledge about language by experiencing single short stimuli in isolation. This is often extended to an assumption that knowledge of language is acquired by discovering distinct cues in the input, each cue elucidating one parameter setting in a parameterized linguistic theory. We will call this assumption the *local learning hypothesis*. This is in contrast to our approach where knowledge of language is acquired by finding data consistent across longer correlated sessions. Our approach requires the learner to do some puzzle solving or constraint satisfaction.⁴ It is normally believed that the latter approach is not cognitively plausible. The evidence for this is that children seem to have short “input buffers.” The limited size of the input buffers is taken to imply that only short isolated stimuli can take part in inferring each new language fact. MAIMRA demonstrates that despite a short input buffer with the ability of retaining only one scenario at a time, it is nonetheless possible to produce a disjunctive representation which supports constraint solving across multiple scenarios. We believe that without cross scenario constraint solving, it is impossible to account for meaning acquisition and thus the local learning hypothesis is wrong. Our approach offers a viable alternative to the local learning hypothesis consistent with the observed short input buffer effect.

7 Related Work

While most prior computational work on meaning acquisition focuses on contextual learning by scanning texts, some notable work has pursued a path similar to that described here attempting to learn from correlated linguistic and non-linguistic input. In [16, 17], Salveter describes a system called MORAN. The non-linguistic component of each scenario presented to MORAN consists of a sequence of exactly two scenes, where each scene is described by a conjunction of atomic formula. The linguistic component of each scenario is a parsed case frame analysis of a single sentence describing the state change occurring between those two scenes. From each scenario in isolation, MORAN infers what Salveter calls

⁴We are not claiming that such puzzle solving is conscious. It is likely that constraint satisfaction, if done by children or adults, is a very low level subconscious cognitive function not subject to introspective observation.

a *Conceptual Meaning Structure* (CMS) which attempts to capture the essence of the meaning of the verb in the sentence. This CMS is a subset of the two scenes identifying the portion of the scenes referred to by the sentence, with the arguments of the atomic formula linked to noun phrases replaced by variables labeled with the syntactic positions those noun phrases fill in the sentence. The process of inferring CMSs involves two processes reminiscent of tasks performed by MAIMRA, namely the figure/ground distinction whereby the inference component suggests possible subsets of the non-linguistic input as being referred to by the linguistic input (as distinct from the part which is not referred to) and the fracturing process whereby verb meanings are constructed by extracting out arguments from whole sentence meanings. MORAN’s variants of these tasks are much simpler than the analogous tasks performed by MAIMRA. First, the figure/ground distinction is easier since each scenario presented to MORAN contains but a single sentence and a pair of scenes. MORAN need not figure out which subsequence of scenes corresponds to each sentence. Second, the linguistic input comes to MORAN preparsed which relies on preexisting knowledge of the lexical categories of the words in the sentence. MORAN does not acquire category information, and furthermore does not deal with any ambiguity that might arise from the parsing process or the figure/ground distinction. Finally, the training session presented to MORAN relies on a subtle implicit link between the objects in the world and linguistic tokens used to refer to them. Part of the difficulty faced by MAIMRA is discerning that the linguistic token *John* refers to the conceptual structure fragment **John**. MORAN is given that information *a priori* by lacking a formal distinction between the notion of a linguistic token and conceptual structure. Given this information, the fracturing process becomes trivial. MORAN therefore, does not exhibit the cross-scenario correlational behavior attributed to MAIMRA and in fact learns every verb meaning with just a single training scenario. This seems very implausible as a model of child language acquisition. In contrast to MAIMRA, MORAN is able to learn polysemous senses for verbs; one for each scenario provided for a given verb. MORAN focuses on extracting out the common substructure for polysemous meanings attempting to maximize commonality between different word senses and build a catalog of higher level conceptual building blocks, a task not attempted by MAIMRA.

In [13, 14], Pustejovsky describes a system called TULLY, which also operates in a fashion similar to MAIMRA and MORAN, learning word meanings from

pairs of linguistic and non-linguistic input. Like MORAN, the linguistic input given to TULLY for each scenario is a single parsed sentence. The non-linguistic input given along with that parsed sentence is a predicate calculus description of three parts of a single event, its beginning, middle and end. From this input, TULLY derives a Thematic Mapping Index, a data structure representing the θ -roles borne by each of the arguments to the main predicate. Like MORAN, the task faced by TULLY is much simpler than that faced by MAIMRA, since TULLY is presented with unambiguous parsed input, is given the correspondence between nouns and their referents and is given the correspondence between a single sentence and the semantic representation of the event described by that sentence. TULLY does not learn lexical categories, does not have to determine figure/ground partitioning of non-linguistic input and implausibly learns verb meanings from single scenarios without any cross-scenario correlation. Multiple scenarios for the same verb cause TULLY to generalize to the least common generalization of the individual instances. TULLY however, goes beyond MAIMRA in trying to account for the acquisition of a variety of markedness features for θ -roles including [\pm motion], [\pm abstract], [\pm direct], [\pm partitive] and [\pm animate]

8 Conclusion

The MAIMRA system successfully learns word meanings with no prior lexical knowledge of any words. It works by applying syntactic, semantic and pragmatic constraints to correlated linguistic and non-linguistic input. In doing so, it more accurately reflects the type of learning performed by children, in contrast to previous lexical acquisition systems which focus on learning unknown words encountered while reading texts. Although, each module implements a weak theory, and in isolation offers only limited constraint on possible mental representations, the collective constraint provided by the combination of modules is sufficient to reduce the nondeterminism to a manageable level. It demonstrates that with a reasonable set of assumptions about innate knowledge, combined with appropriate representations and algorithms, tractable learning is possible with short training sessions and limited processing. Though there may be disagreement as to the linguistic and cognitive plausibility of some of the innateness assumptions, and while the particular syntactic, semantic and pragmatic theories currently incorporated into MAIMRA may be only approxima-

tions to reality, nonetheless, the general framework shows promise of explaining how children acquire word meanings. In particular, it offers a viable alternative to the local learning hypothesis which can explain how children acquire meanings that require correlation of experience across many input scenarios, with only limited size input buffers. Future work will attempt to address these potential shortcomings and will focus on supporting more robust acquisition of a broader class of word meanings.

ACKNOWLEDGMENTS

I would like to thank Peter Szolovitz, Patrick Winston and Victor Zue for giving me the freedom to embark on this project and encouraging me to elaborate on it; AT&T Bell Laboratories for supporting this work through a Ph.D. scholarship; Johan deKleer, Kris Halvorsen and everybody at Xerox PARC for listening to half-baked versions of this work prior to completion; Bob Berwick, Barbara Grosz, David McAllester and George Lakoff for many interesting discussions; and Ron Rivest for pushing me to complete this paper.

References

- [1] Robert C. Berwick. Learning word meanings from examples. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 459–461, 1983.
- [2] Noam Chomsky. *Lectures on Government and Binding*, volume 9 of *Studies in Generative Grammar*. Foris Publications, 1981.
- [3] Noam Chomsky. *Some Concepts and Consequences of the Theory of Government and Binding*, volume 6 of *Linguistic Inquiry Monographs*. The M. I. T. Press, Cambridge, Massachusetts and London, England, 1982.
- [4] Noam Chomsky. *Barriers*, volume 13 of *Linguistic Inquiry Monographs*. The M. I. T. Press, Cambridge, Massachusetts and London, England, 1986.
- [5] Richard H. Granger, Jr. FOUL-UP a program that figures out meanings of words from context. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 172–178, 1977.
- [6] Ray Jackendoff. *Semantics and Cognition*. The M. I. T. Press, Cambridge, Massachusetts and London, England, 1983.
- [7] Paul Jacobs and Uri Zernik. Acquiring lexical knowledge from text: A case study. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 739–744, August 1988.
- [8] T. Kasami. An efficient recognition and syntax algorithm for context-free languages. Scientific Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford MA, 1965.
- [9] George Lakoff and Mark Johnson. *Metaphors We Live By*. The University of Chicago Press, 1980.
- [10] David Allen McAllester. Solving SAT problems via dependency directed backtracking. Unpublished manuscript received directly from author.
- [11] David Allen McAllester. An outlook on truth maintenance. A. I. Memo 551, M. I. T. Artificial Intelligence Laboratory, August 1980.
- [12] Fernando C. N. Pereira and David H. D. Warren. Definite clause grammars for language analysis—a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13(3):231–278, 1980.
- [13] James Pustejovsky. On the acquisition of lexical entries: The perceptual origin of thematic relations. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 172–178, July 1987.
- [14] James Pustejovsky. Constraints on the acquisition of semantic knowledge. *International Journal of Intelligent Systems*, 3(3):247–268, 1988.
- [15] Manny Rayner, Åsa Hugosson, and Göran Hagert. Using a logic grammar to learn a lexicon. Technical Report R88001, Swedish Institute of Computer Science, 1988.
- [16] Sharon C. Salveter. Inferring conceptual graphs. *Cognitive Science*, 3(2):141–166, 1979.
- [17] Sharon C. Salveter. Inferring building blocks for knowledge representation. In Wendy G. Lehnert and Martin H. Ringle, editors, *Strategies for Natural Language Processing*, chapter 12, pages 327–344. Lawrence Erlbaum Associates, 1982.
- [18] Roger C. Schank. The fourteen primitive actions and their inferences. Memo AIM-183, Stanford Artificial Intelligence Laboratory, March 1973.
- [19] D. H. Younger. Recognition and parsing of context-free languages in time $O(n^3)$. *Information and Control*, 10(2):189–208, 1967.