ABSTRACT

AUTOMATIC ACQUISITION OF WORD MEANING FROM CONTEXT

by
Peter Mark Hastings

Chair: Assistant Professor Steven Lytinen

This thesis presents an automatic, incremental lexical acquisition mechanism that uses the context of example sentences to guide inference of the meanings of unknown words. The goal of this line of research is to allow a Natural Language Processing (NLP) system to cope with words that it does not know — not just to gloss over them, but to try to infer what they mean. The environment within which this system operates is epitomized by the information extraction task: from virtually unconstrained text, elicit certain information that is deemed interesting. The knowledge acquisition bottleneck inherent in this task imposes constraints on the type of knowledge available for lexical inference. The main objective in this work is to infer as much information as possible about unknown words from context without requiring special-purpose knowledge. This was accomplished by extending the underlying NLP system to search its domain-specific concept representation for an appropriate concept to denote the meaning of the unknown word. The learning method is incremental, so every time the system encounters an example of an unfamiliar word, it adjusts its hypotheses. The basic system evolved through several different stages in order to improve its inferences. Then several variations to the basic system were made to capture especially difficult aspects of the acquisition task and to take advantage of discourse context. The approach was tested in two different domains. Target words were removed from the lexica and sentences containing them were processed by the system. The results were evaluated using measures taken from the field of Information Retrieval.

When humans learn language, they are faced with a similar task: from a set of examples of a word's use, they must infer what that word means and how it is used. Not only is the task similar, but many of the behaviors and difficulties that the computational acquisition mechanism have encountered have also been described in the psycholinguistic literature. Although the system was not intended as a cognitive model, these parallels indicate strong constraints from the task itself, and therefore lend credence to viewing the system as a cognitive model.

AUTOMATIC ACQUISITION OF WORD MEANING FROM CONTEXT

by
Peter Mark Hastings

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
1994

Doctoral Committee:

Assistant Professor Steven L. Lytinen, Chair
Associate Professor John E. Laird
Associate Professor Robert Lindsay
Professor William C. Rounds
Professor Marilyn Shatz

Before my teacher came to me, I did not know that I am. I lived in a world that was a no-world. I cannot hope to describe adequately that unconscious, yet conscious time of nothingness.... Since I had no power of thought, I did not compare one mental state with another.

Helen Keller
The World I Live In
Century Company, New York, New York (1908), pages 113, 116.

One might learn as much of value to psychology or epistemology from a *particular* but highly *un*realistic AI model as one could learn from a detailed psychology of, say, Martians. A good psychology of Martians, however unlike us they might be, would certainly yield general principles of psychology or epistemology applicable to human beings.

Daniel Dennett
Brainstorms
MIT Press, Cambridge, Massachusetts (1978), page 113.

ACKNOWLEDGEMENTS

There are many people who have influenced my work and made this dissertation possible. First, I would like to thank my advisor, Steve Lytinen for consistently helpful guidance even in the face of adversity. He has helped me not just in writing this thesis, but in becoming a researcher. Thank you Steve, you were a great advisor. I would also like to thank the members of my committee with whom I've had many fruitful conversations about my thesis work and other topics. Bill Rounds helped me think about what I was doing in more theoretical terms. John Laird made me consider the Machine Learning aspects of my work. Bob Lindsay made sure I kept the science in cognitive science. Marilyn Shatz introduced me to the human side of language acquisition and gave me many excellent suggestions for finding out more.

The other faculty and students at the AI lab deserve much credit for the providing stimulating environment to work in. In particular, the NLP group, consisting at various times of Bill, Steve, Carol, Paul, Jeff, Mark, Mark, Craig, Scott, Chris, Rob, Sayan, and Karen, provided a rich environment for exploring the intricacies of NLP.

Most of all, I would like to thank my family: my parents who raised me right and my sisters, especially now that I no longer have to bribe them to play with me. Thanks also to my Aunt Joan and Uncle Jim who took care of me when I moved east. I owe much to my fellow squash and ultimate players, for helping me retain a modicum of sanity. I would also like to warmly thank the many friends that I have met here for making my time enjoyable as well as educational. Thank you Clare, Randy, Pradeep, Kate, Mike, Renee, Craig, Miriam, Suz, Steve, Jeff, Jody, Kathleen, Dina, Joyann, and Lina.

Claire Cardie helped me understand her lexical acquisition system and shared helpful comments about word learning in general.

Chris Huyck, Clare Congdon, and Randy Jones gave me very helpful comments on the developing thesis.

TABLE OF CONTENTS

LIST OF APPENDICES

Appendix

CHAPTER 1

THE LEXICAL ACQUISITION TASK

This thesis describes a mechanism for learning word meanings and lexical categories. From the context of example sentences containing one or more unknown words, the mechanism uses what it knows about other constituents of the sentences to constrain interpretations of unknown words. It is an incremental learning mechanism, so each time it encounters an example of an unfamiliar word, it refines its hypotheses. Thus it is able to (and often does) make an incorrect initial guess about the meaning of the word and then recover based on additional examples of the word's use.

The mechanism is implemented as an extension of the LINK Natural Language Processing (NLP) system [Lytinen and Roberts, 1989b; Lytinen, 1991]. The specifics of LINK which relate to the lexical acquisition task are described in the next chapter. LINK has been applied to a wide variety of tasks, and it is particularly well-suited to the information extraction task, in which text is processed with the goal of pulling out specific "interesting" pieces of the text or concepts derived from it.[1] This type of task provides both one of the basic motivations for this thesis and one of the constraints for its implementation. The most time-consuming aspect of developing an information extraction system is giving it all of the linguistic knowledge that is unique to a particular domain. Another complicating factor is that because the input text is unconstrained, the NLP system is virtually guaranteed to encounter linguistic formulations that are not in its knowledge base. Therefore, in order to perform the information extraction task robustly, an NLP system must contain some mechanism to handle unexpected input.

Furthermore, because the knowledge acquisition problem for any domain is so large, it is important to limit the depth of knowledge that the system requires. Moreover, it is not clear that adding some types of domain knowledge would further increase the power of an information extraction system. For example, the system need not know the motivations of the agent that performed a particular action, only that the action has occurred. Therefore, it is a specific goal of the word-learning mechanism that it rely primarily on the linguistic information required for standard processing, that is, basic knowledge about words, syntax, basic domain knowledge. Requiring significant additional domain knowledge would, in essence, make the lexical acquisition task infeasible.

There have been several other implementations of word-learning mechanisms built in the last 15 years. The approach described here differs from them in the depth of knowledge used and the extent to which it is used. On the lowest end of the knowledge-use spectrum are the statistics-based methods described in [Brent, 1991; Brent, 1993a; Brent, 1993b; Church and

[1] For a general description of an information extraction task, see [Sundheim, 1992]. For a description of the LINK implementation for MUC, see [Lytinen *et al.*, 1992a; Lytinen *et al.*, 1992b; Lytinen *et al.*, in press].

Hanks, 1990; Hindle, 1990; Resnik, 1992; Yarowsky, 1992; Zernik, 1991]. These use virtually no world knowledge whatsoever and instead rely on large corpora to allow them to categorize words into broad classes. Although this categorization could be construed as semantic information, it does not constitute word meaning as it is used in this thesis, therefore these systems will not be further addressed herein.

Another set of systems, [Salveter, 1979; Cardie, 1993; Riloff, 1993; Selfridge, 1986; Siskind, 1990; Siskind, 1991], uses slightly more complex concept representations. These systems have rudimentary concept classes, like Physical-Object, Human, and Move. Such categories allow the systems to make broad constraints, for example that the actor of an action must be Human, and to learn the meanings of words as mappings to these categories or to specific instances of the categories. These systems use a variety of mechanisms to infer the meanings. One is case-based, one uses a graph construction method, and one proposes patterns that a human must check.

Some researchers have taken a knowledge-intensive approach to lexical acquisition [Granger, 1977; Zernik, 1987a]. These systems have somewhat coarse-grained semantic hierarchies along with additional information regarding causes and motives behind actions. This allows these systems to make powerful inferences about word meanings. Because there is so much knowledge required, however, these approaches have only been applied to very limited domains and tasks.

The system described in this thesis, Camille (Contextual Acquisition Mechanism for Incremental Lexeme Learning), takes a middle ground on the depth of knowledge issue, motivated in large part by the tasks for which it is intended. In fact, all of the decisions about which objects to represent and how to structure them are based on task and linguistic differentiability.[2] Objects are broken into a rich hierarchy, based on their similarity and function. Actions have constraints on the constituents that can be attached to them as role-fillers. These constraints determine the hierarchy for action concepts. This semantic information, which is used by the NLP system, is also exactly the information which is used in inferring the meanings of unknown words. This makes Camille ideally suited for the information extraction task.[3]

Camille is further distinguished by the type of inference mechanism used. It is the only non-trained, incremental word-learning system. Unlike some of the other systems [Salveter, 1979; Selfridge, 1986; Siskind, 1990], Camille does not rely on a trainer to feed it sentences and give it a representation for the meanings of those sentences. It learns automatically, using only the linguistic information that is found in the text.

Given the nature of the knowledge representation and the available evidence, Camille makes the best hypotheses possible. The inference task can be naturally viewed as a graph-search problem. To learn a word, the system must infer a mapping from the word to some node in the concept representation structure. Many different nodes are likely to provide reasonable interpretations for the unknown word. Thus the lexical acquisition mechanism has a choice. It can maintain a large set of possible meanings, or it can inductively choose a single node or a small set from among the consistent concepts. Camille takes the latter approach and prefers more specific nodes to general ones. This results in two advantages. First, because the hypothesized meaning is more specific, it has more information content, that is it specifies more

[2] Appendix A contains a discussion of the role of knowledge representation in lexical acquisition.

[3] Interestingly, none of the systems has a very deep knowledge representation. Without exception, the systems use concepts like Block and Stack as atomic, and is there is no further delineation of their meaning.

precisely what the word means. If the hypothesis is correct, the system knows more about that word. Second, the more specific a hypothesis is, the more falsifiable it is. Additional examples are likely to either contradict the initial hypothesis (reinvoking the search for an appropriate concept) or confirm it.

Camille wasn't implemented in a day. It evolved through a series of stages. Learning nouns was relatively easy. The constraints from the verbs on their slot fillers naturally limits the interpretations of the nouns that fill the slots. The verbs were more difficult to pin down. Initially, Camille considered each concept that was consistent with the evidence given it. A second version of the system retained only the most specific consistent concepts in the hypothesis sets. The final version of the basic system ranked the hypotheses and eliminated all but the most falsifiable. It relied on multiple examples of the word's use to select the appropriate candidate. Only at this point, after it had fully exploited all available information from within a sentence to infer word meanings, was Camille extended to allow it to make conclusions based on multiple sentences. A simple script mechanism was implemented that allows Camille to further refine its hypotheses.

In order to ensure that Camille was performing adequately, empirical testing was performed after each new addition. To test the system, some word definitions were removed, and the system was given randomly chosen sentences from which to derive the meanings. Details of the test procedures, the results, and some analyses are in Appendix B.

Camille was developed with the goal of automatically making the best inferences possible about word meaning from context using the knowledge available for parsing. While it was being developed, however, it became apparent that there were interesting similarities between Camille's behavior and that of children when they learn language. Although the system was never intended to model human behavior, it appeared to be learning in a similar fashion. This is despite the fact that there are some major differences between a child's word-learning task and Camille's. For one thing, children are not limited to linguistic input. They receive visual and other sensory information to help them interpret an utterance. So the fact that these parallels were found led to the hypothesis that the constraints of the word-learning task were so strict that they would force any reasonable implementation into similar behavior. Given this possibility, there are interesting implications for using Camille to predict particular behaviors of children when they face the same task.

The evolution of Camille, testing results, and its implications for cognitive modelling are presented in this thesis as follows:

- **Chapter 2** gives a brief description of the LINK system which forms the foundation upon which Camille is built. The basic knowledge bases that it uses will be described along with the parsing mechanism.

- **Chapter 3** presents the lexical acquisition mechanism. First the basic mechanism will be described, and then the evolution of the mechanism to enable it to make use of various knowledge sources. The chapter includes an analysis of the limitations of this mechanism and ways that it could be extended.

- **Chapter 4** describes the variations that were implemented on top of Camille's basic graph-search mechanism. One was a technique taken from psycholinguistic theory. Another used scripts to enable Camille to use context from multiple sentences. A third allowed the system to recognize ambiguous words. The final variation allowed Camille to add concepts to its knowledge representation. The chapter ends with analyses of the

system's evolution and of the aspects of the tested domains which made them more or less amenable to lexical acquisition.

- **Chapter 5** examines previous learning programs and how they relate to Camille. A framework is built for categorizing the different methods of learning. The systems are then organized according to that framework so that they can be easily compared with each other and with Camille.

- **Chapter 6** analyzes the connections between the lexical acquisition mechanism and research in how humans learn language. The extent to which the constraints of the task affect the solutions (human or computer) is also addressed.

- **Chapter 7** contains the conclusions drawn from this work. It also includes a summary of the cognitive implications of this work and some ways in which the model could be extended.

CHAPTER 2

THE FOUNDATION – LINK

The lexical acquisition techniques described herein have been implemented as an extension of the LINK NLP system [Lytinen, 1990; Lytinen, 1991]. LINK uses a unification grammar (described later in this section) and extends the mechanism of chart parsing[1] by integrating syntactic and semantic processing. LINK has been used in many different prototype domains in which the conceptual knowledge can be fairly completely specified, but the textual input is entered by a large number of users and is therefore subject to wide variations in terminology. The examples in this thesis come from one of these domains, in which the texts consist of newswire reports of terrorist activity.[2] This chapter contains a description of the knowledge bases and process involved in normal parsing where all the input words are known to the system.

2.1 Domain knowledge

The domain knowledge for LINK consists of a hierarchy of nodes which are the atomic meaning units. The definition:

```
(define-sem school
   is-a (building))
```

specifies that there is a semantic node called School[3] which is a type of Building. Figure 2.1 gives a sampling of the 156 object concepts that are defined in the terrorism domain and their relations to each other.

Semantic nodes can also specify constraints on relationships between nodes. For example, the definition:

```
(define-sem Arson
   is-a (Terrorist-Act)
   formulae (((Object) = Building)))
```

specifies that Arson is a type of Terrorist-Act, and constrains its OBJECT to be a type of Building.

---

[1] See [Winograd, 1987] for a description of chart parsing.

[2] Appendix B contains a description of a second domain and complete testing results for cross-domain verification.

[3] Throughout this document, the Sans Serif type style will be used to display the names of concepts. The SMALL CAPS type style will be used to display grammar literals, syntactic markers, and function names.

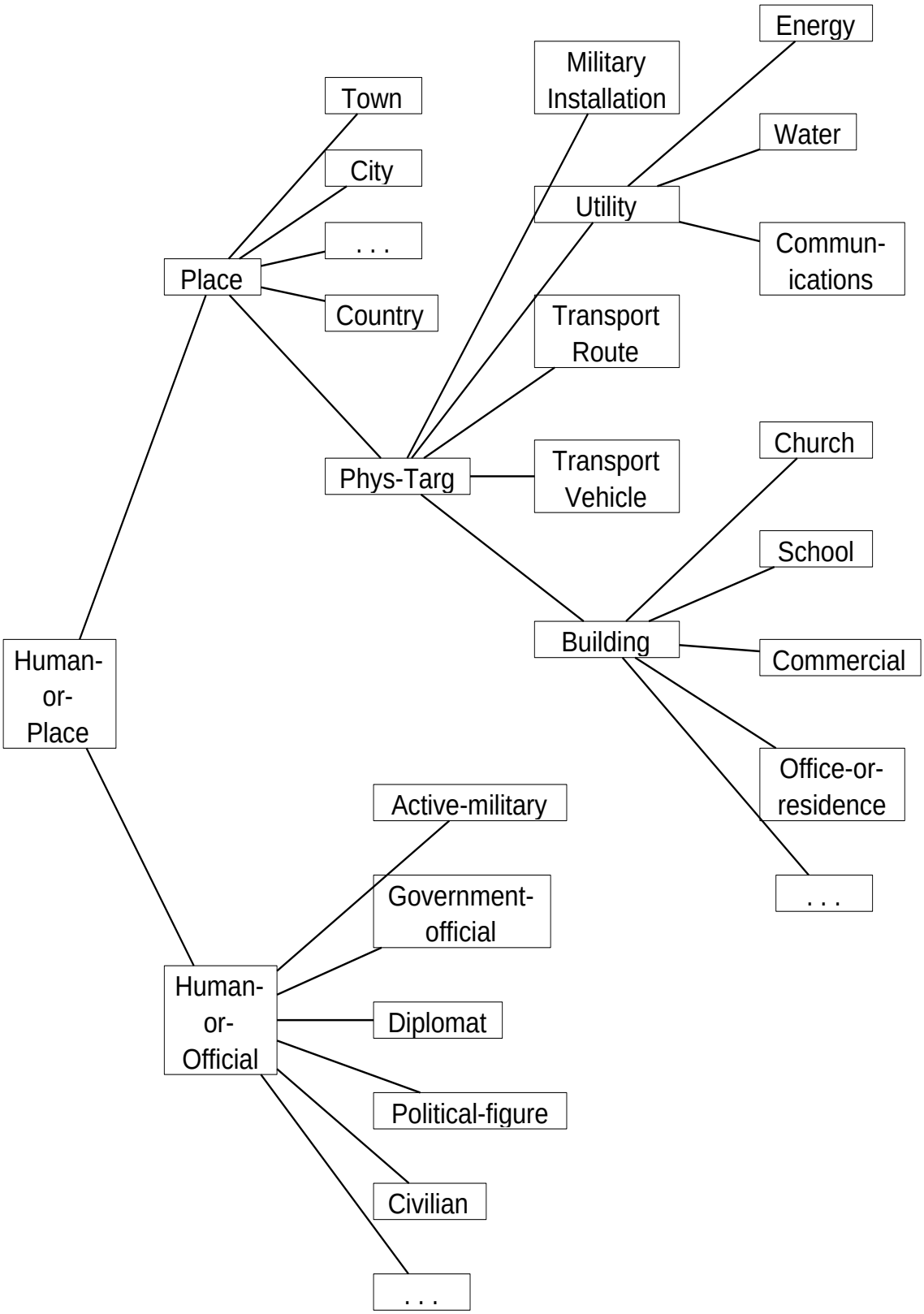


Figure 2.1: An object concept hierarchy for LINK

Verbs normally act as the *head* (center of the representation) of sentences and the parsing process attaches *slot-fillers* (e.g. OBJECT) to them. Because of this, the semantic constraints occur almost entirely on the nodes which represent actions. There are many different slot fillers that can be filled by a parse. The definition for the Bombing act includes constraints not only on the OBJECT of the action, but on the ACTOR and INSTRUMENT as well.

```
(define-sem Bombing
   is-a (Nasty-Action)
   formulae (((Actor) = Human-or-Org
              (Object) = Human-or-Place
              (Instrument) = Explosive)))
```

Figure 2.2 displays the portion of the semantic hierarchy that represents the actions in the domain. Each node includes its slot-filler constraints. The constraints are inherited by the descendants of each node, and any constraint on a child must be at least as specific as the constraints of its ancestors. For example, because Attack takes an OBJECT that is a Human-or-Place, this restriction also implicitly holds for actions like Terrorist-Act and Robbery. Destroy is an example of a concept which makes a further restriction on a previously constrained slot. Phys-Targ, the OBJECT of this action, must be a descendant of Human-or-Place.

The structure of the object hierarchy is fairly obvious. It is based on a hierarchy of subsumption categories, for example, a Phys-Targ is a type of Place, a Building is a type of Phys-Targ, and a School is a type of Building. The ordering of the nodes in the action hierarchy is somewhat more complicated (as will be further discussed in Chapter 6). The organization is mostly determined by the generality of the slot-filler constraints, but a grouping of similar concepts (e.g. Strans and its descendants) also comes into play. This structure is crucial for the word learning mechanism and will be further explored later.

2.2 Grammar

Three different ways of constructing a verb phrase are specified in the following portion of the VP rule:

```
(define-gram VP                              ; 1
   (((1) = Verb                              ; 2
     (Head) = (1 Head)                       ; 3
     (Head Syn Vtype) = Intrans)             ; 4

    ((1) = Verb                              ; 5
     (2) = NP                                ; 6
     (Head) = (1 Head)                       ; 7
     (Head Syn Vtype) = Trans                ; 8
     (Head Syn Vform) = Simple-Past          ; 9
     (Head Sem Object) = (2 Head Sem))       ; 10

    ((1) = VP                                ; 11
     (2) = Particle                          ; 12
     (Head) = (1 Head)                       ; 13
     (Head Part) = (2 Head))))               ; 14
```

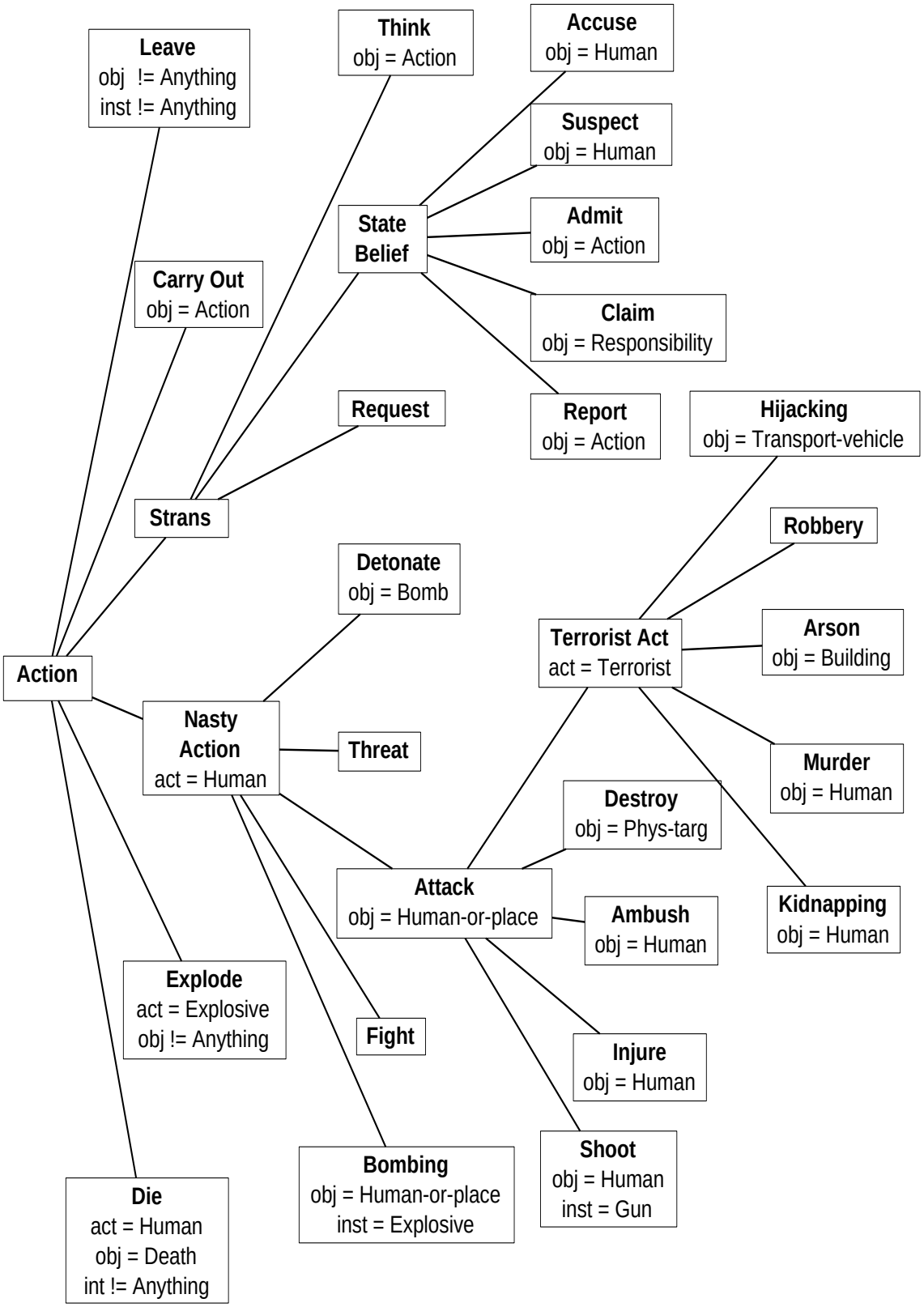


Figure 2.2: An action concept hierarchy for LINK

The numbered equations at the beginning of each of the three segments specify (in order) the constituents of the phrase. The first segment makes a verb phrase out of a single verb and specifies that the type of the verb is intransitive. The second segment has two constituents, a verb and a noun phrase. The equation in line 7 specifies that the head (again, the central point of the representation) of the verb phrase comes from the head of the verb. Line 8 specifies that the type of the verb is transitive, line 9 states that the form of the verb is simple past, and line 10 states that the semantic object of the verb phrase is the semantics of the head of the noun phrase. The third segment is for verb/particle combinations like, "take the lid off".

Two types of equations are shown in this rule. "Labelling equations" in lines 2, 4 – 6, 8, 9, 11, and 12 specify the label of the constituent to which they refer. INTRANS and TRANS are literals that specify the transitivity of the verb phrase. VERB, NP, VP, and PARTICLE refer to other constituents that are either lexical entries or built up by other grammar rules. The other equations are called Unifying Equations. They link together two different parts of the structures that are created during the parsing process. These structures are defined more completely in section 2.4.

2.3 Lexicon

```
(define-word torched                          ; 1
   (Verb (Head Sem) = Arson                   ; 2
         (Head Syn Vform) = Simple-Past       ; 3
         (Head Syn Vtype) = Trans))           ; 4
```

The preceding definition shows the dictionary entry for the word "torched". The syntax of lexical entries is roughly the same as that for grammar rules. The translation of this dictionary entry to English is:

> The meaning representation (SEM) for the word "torched" is the concept Arson.
> The syntactic verb form is simple past and the syntactic verb type is transitive.

There is a spelling checker and morphology component in LINK that finds the root of each word in the input sentence. It returns the definition for the root as well as any affixes the word might have.

This uniformity of representation of the constraint definitions across the semantics, grammar, and lexicon enables the integrated application of a wide variety of constraints as described in the next section.

2.4 Parsing

The basic mechanism underlying processing in LINK is a bottom-up chart parser. The chart is simply a repository for storing the partial parses of subconstituents. This allows the parser to avoid redundant processing. As a parse progresses, the parser can use the stored constituent representations instead of starting from scratch. The goal of the parser is to combine the word constituents into phrase constituents and then into a sentence constituent which will include the entire parse tree for that sentence.

As mentioned above, one of LINK's strengths is its use of a uniform representation format for all of its information. The definitions, such as those that have been seen earlier in this chapter, are all translated into the form of a directed acyclic graph, or DAG. Figure 2.3 shows a simple lexical definition with the corresponding DAG structure.

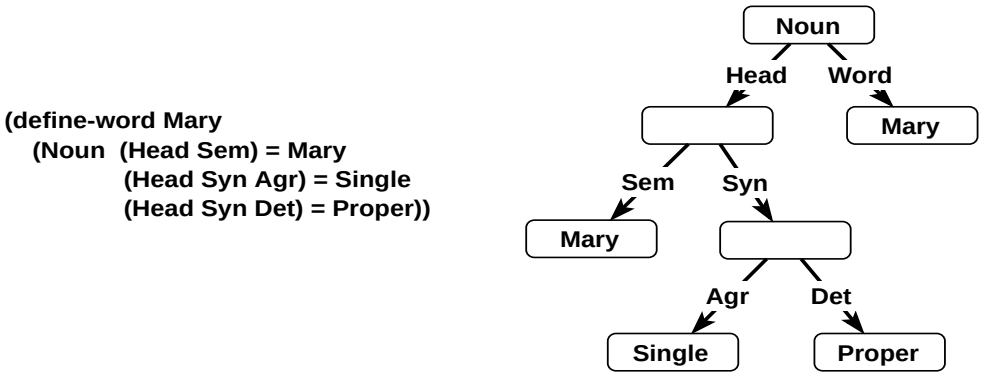


Figure 2.3: A lexical definition and its DAG

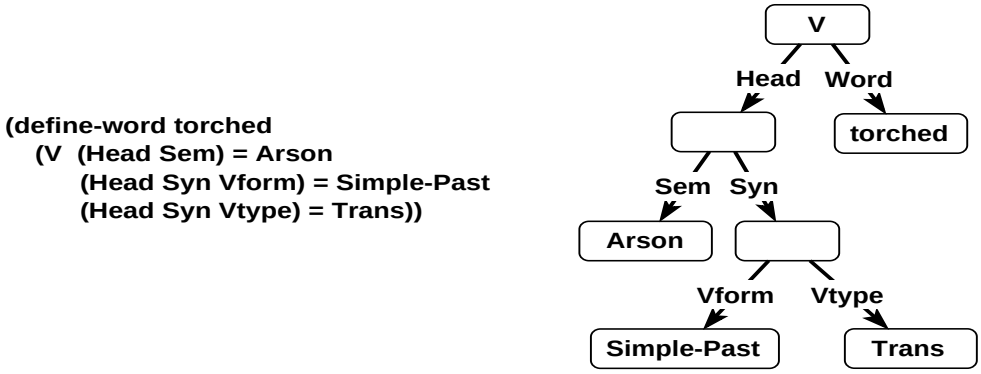


Figure 2.4: Another lexical definition and its DAG

Figure 2.4 shows the lexical definition for the verb, "torched". At the start of the parsing process, LINK enters all the definitions for the input words into the chart (graphically represented in the following figures by a box). Figure 2.5 depicts the chart as it would stand after lexical definitions are entered for the sentence, "Mary torched the headquarters."

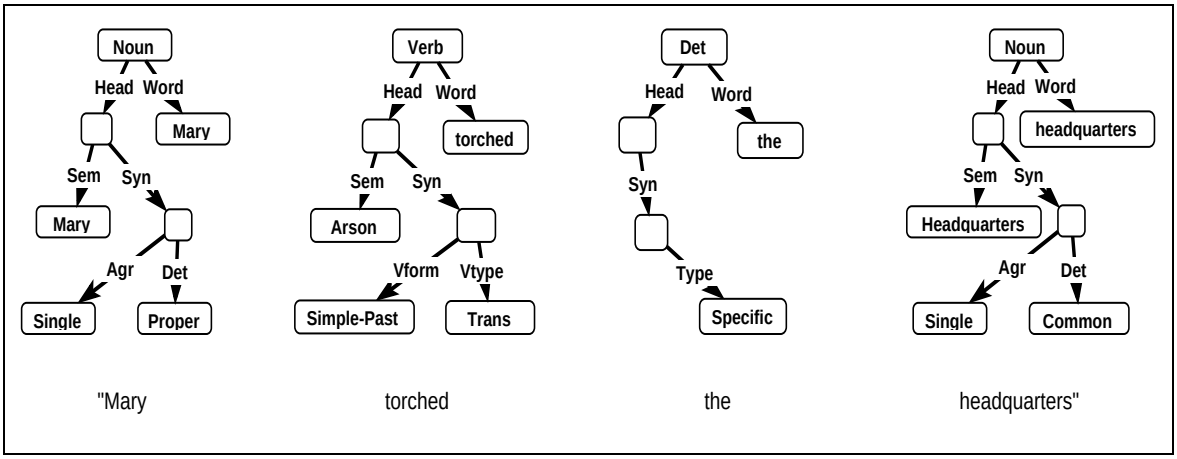


Figure 2.5: The chart with lexical definitions

When the words are added, the parser also brings in any associated semantic con-

straints. Because the SEM node of "torched" is the Arson concept, the parser brings in its constraints, namely ((OBJECT) = Building) and ((ACTOR) = Terrorist) (inherited from Terrorist-Act). The DAG form of these constraints is unified with the corresponding node of the DAG for "torched", leaving the chart as shown in figure 2.6.

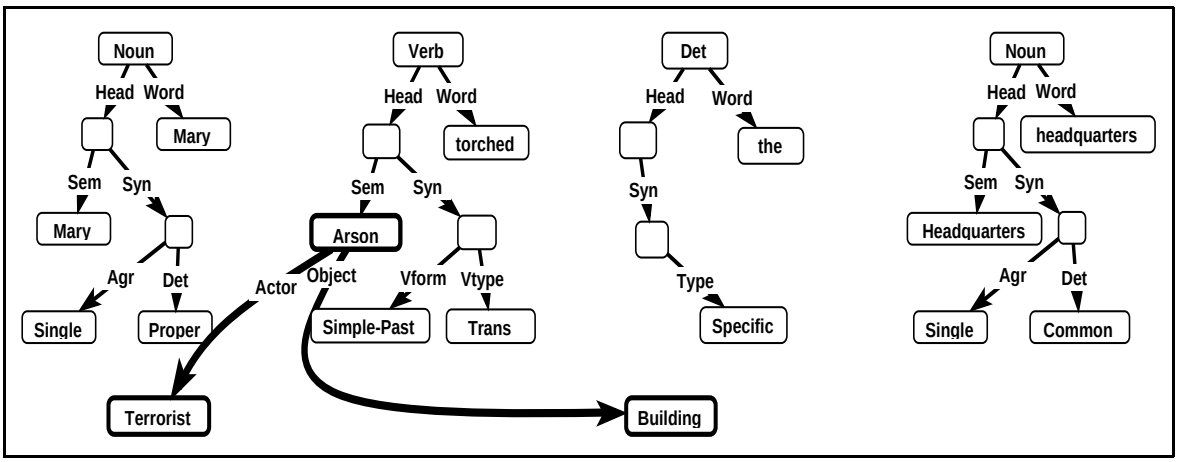


Figure 2.6: The chart with semantic information

At this point, the parsing process starts. LINK repeatedly examines the constituents in the chart to see which can be combined using the grammar rules to make larger constituents. (Only the successful rule applications will be described here.) The first step for the parser will be to apply the NP rules to create two new noun phrases, one from the single proper noun, "Mary", and the other from the determiner and noun, "the headquarters". The portion of the NP rule that covers these situations is:

```
(define-gram NP
   ((1) = Noun
    (Head) = (1 Head)
    (Head Syn Det) = Proper)
   ((1) = Det
    (2) = Noun
    (Head) = (2 Head)
    (Head Syn Det) = Common)))
```

After the structures from the NP rules are unified into the chart, the result is as shown in figure 2.7. Note that the HEAD arcs of the NP rules link to the HEADs of the nouns.

Next, the parser applies the verb phrase rule described in section 2.2 to the verb and the new NP constituent. Several important things happen at this point. The head of the verb phrase is linked to the head of the verb in the structure which represents "torched". Also, the OBJECT arc from the Arson node, which pointed to a DAG labelled Building before, is now unified with the semantic node for the object NP, Headquarters. (If the object was of a type that was incompatible with Building, the unification would have failed, and therefore, the rule would not have fired.) The results of these latest additions are shown in figure 2.8.

Now that the chart contains an NP followed by a VP, this S rule can fire:

```
(define-gram S
   ((1) = NP
```

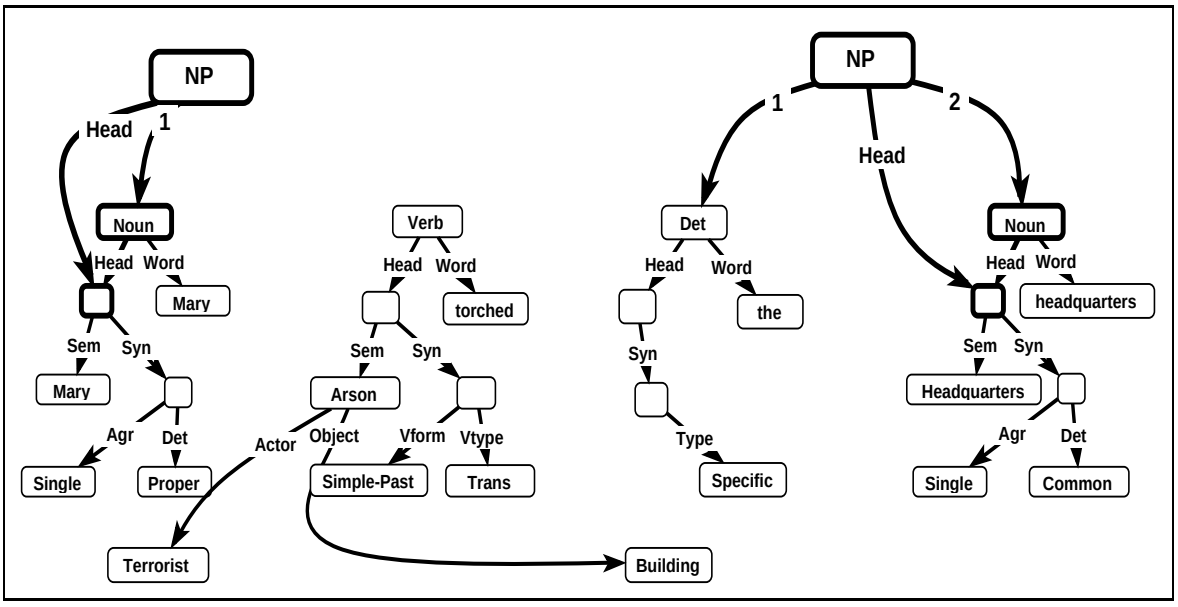


Figure 2.7: The chart after the firing of Noun Phrase rules

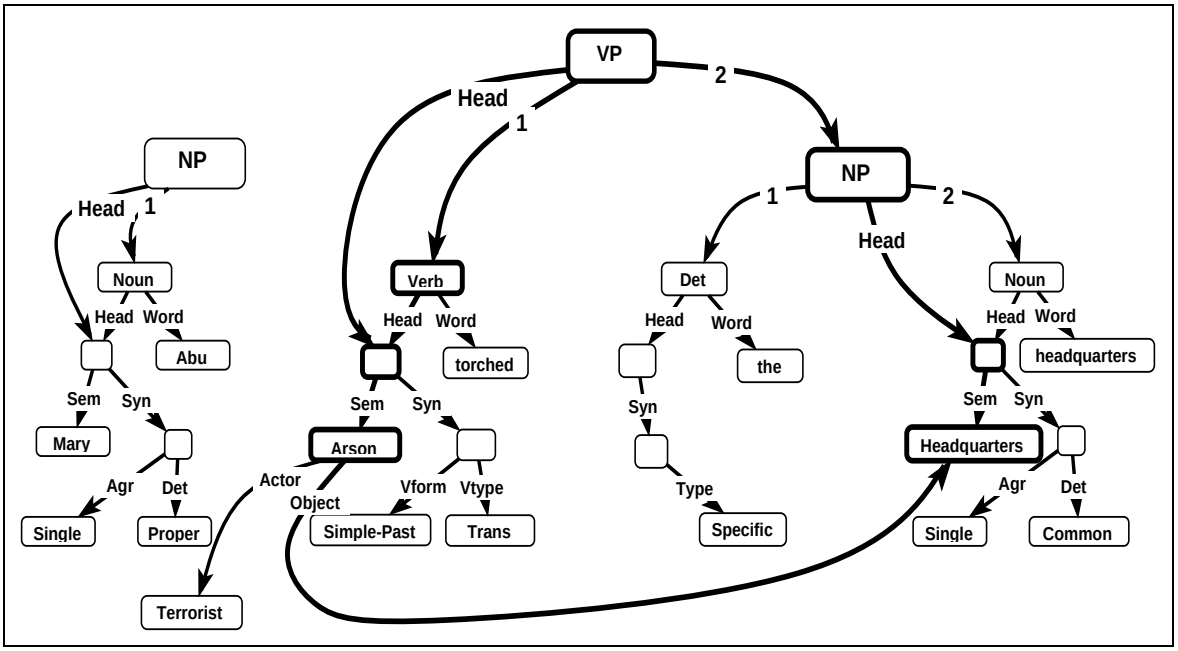


Figure 2.8: The chart after the firing of the Verb Phrase rule

```
(2) = VP
(Head) = (2 Head)
(Head Sem Actor) = (1 Head Sem)
(Head Syn Agr) = (1 Head Syn Agr)))
```

This links the head of the sentence to the head of the VP (which is also the head of the verb). Now the semantics of the completed sentence parse will come from the semantics of the verb. Firing the S rule also causes unification of the Terrorist node on the ACTOR arc with the semantics of "Mary" and the linking of the AGR (agreement) nodes of the subject and verb. The final structure is shown in figure 2.9. The semantic representation of this sentence is taken from the (HEAD SEM) path, namely that there was an Arson action and the actor was Mary and the object was Headquarters.

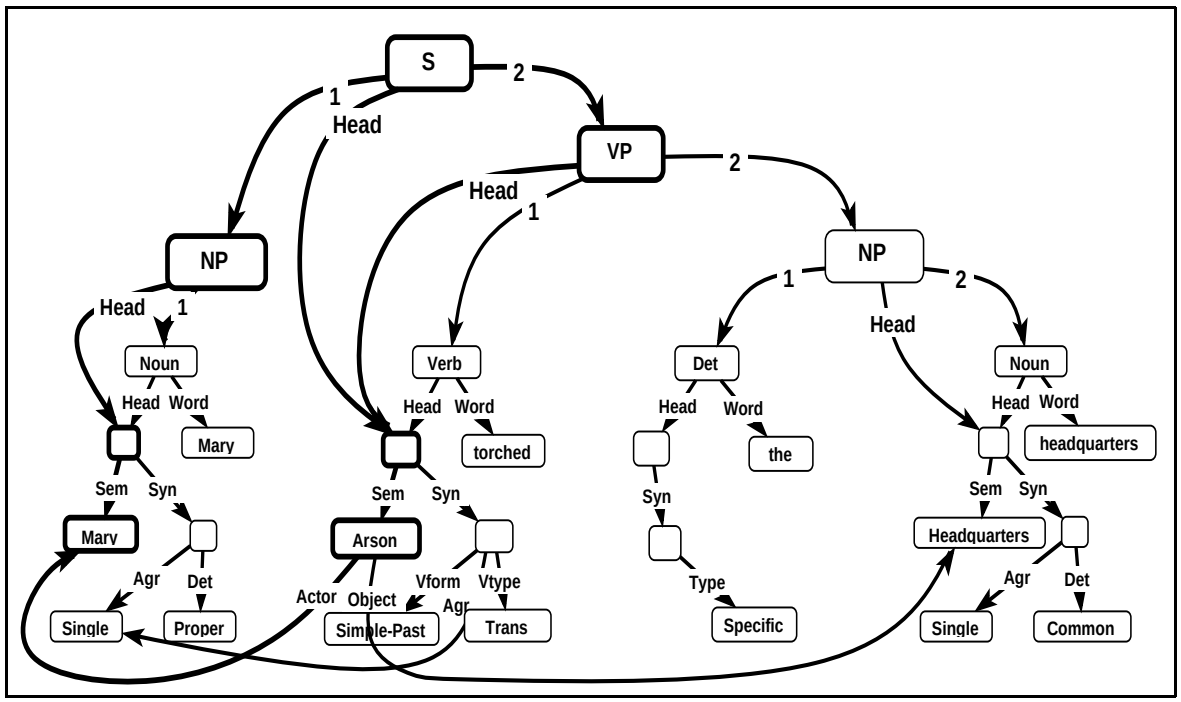


Figure 2.9: The complete parse

The main advantage of the LINK parsing mechanism over traditional parsing is that it integrates the application of syntactic and semantic constraints. For example, consider the creation of a verb phrase from the verb "torched" and the noun phrase "the headquarters". The grammar rule specifies the syntactic category for each constituent. The syntactic constraints within the VP grammar rule are used to ensure number and tense agreement between constituents.[4] At the same time, the semantic constraints are applied. The rule which creates the verb phrase also links the semantic OBJECT slot of the verb with the noun phrase's semantics. The OBJECT slot already contains the semantic constraint, Building. When the rule fires, the unification between Headquarters and Building (resulting in the more specific label, Headquarters) takes place, ensuring that the semantic constraints are met. This integration of

[4]These syntactic constraints alone are normally sufficient to disambiguate between the various senses of a word. If not, the semantic constraints provide further discriminating information.

syntactic and semantic processing ensures that the constraints are brought to bear as soon as possible, allowing the parser to avoid wasting time on dead-end parses.

2.5 Importance of the parser for lexical acquisition

There are several implications of the general parsing technology for lexical acquisition. In the parsing example, it was the verb to which the semantic constraints were attached. As will be discussed in the next chapter, verbs tend to play a central organizing role in both the syntactic and semantic structure of sentences. The head of the verb phrase comes from the head of the verb, and the head of the sentence comes from the head of the verb phrase. The other constituents of the sentence are attached to the verb, and in so doing are added to the sentence structure. Then, the semantic representation of the verb gives the overall meaning of the sentence.

One specific advantage of LINK for lexical acquisition is its uniform representation scheme. The basic unit of representation for syntactic, semantic, and pragmatic information is the DAG. The next chapter describes how the lexical acquisition mechanism exploits this uniformity to infer word meanings within the scope of normal processing.

Finally, the integration of syntactic and semantic constraints is important in that it allows for incremental inference of word meaning. As each constituent is attached during the parse, all of its syntactic and semantic components are already complete, and the additional evidence can be used by the inference procedure. Thus, the same mechanism that performs the inference when new slots are filled in the parse applies to refining the inferred meaning when new examples of a given word's use are encountered.

CHAPTER 3

CAMILLE: A LEXICAL ACQUISITION MECHANISM

This chapter describes the basic lexical acquisition mechanism. It starts with an examination of logical properties of the acquisition task and how these theoretical constraints influenced Camille's architecture. The rest of the chapter lays out the evolution of the basic word-learning mechanism and the empirical testing that was performed. The chapter ends with an analysis of the strengths and weaknesses of the approach.

In Chapter 4, variations on the Camille's basic graph search mechanism will be presented. Each of the variations explores a particular aspect of the lexical acquisition task.

In order to improve the readability of the thesis, the details of all of the tests and their results are presented in Appendix B. Brief summaries of the tests are included here with analyses of their significance for lexical acquisition.

3.1 The nature of the knowledge

There is much discussion in the Philosophy literature about the essence of conceptual knowledge. These issues will be discussed in Appendix A because although they provide the underlying foundation of the research, they are not the main focus of the research. One general point about knowledge representation, however, serves as an important introduction to this chapter. Intuitively it is clear (and it has been addressed by [Katz and Fodor, 1963] among others) that there are many different aspects of conceptual knowledge. Consider what the word "Arson" brings to mind: techniques, instruments, likely targets, motivations of the actors. LINK, on the other hand, knows only this about the concept called Arson[1]: that it is an action, specifically, a Terrorist-Act, and that it is perpetrated by a Terrorist on a Building. Any additional information is implicit, and it is left to the application to decide the rest of the meaning of this concept.

This fundamental limitation[2] has strong implications for what type of inference is possible without relying on additional information. In particular, inferences based on results, preconditions, and goals are not possible because the system knows nothing about them. Within the confines of a system that learns automatically using no special-purpose knowledge, only certain types of knowledge can be acquired. The remainder of this section addresses abstractly the boundaries of learning circumscribed by LINK's concept representation.

---

[1] Figure 2.2 displays this concept and other actions in LINK's domain knowledge for the Terrorism domain.

[2] It is clear that every computer implementation must have this limitation to some extent because computers lack humans' sensory apparatus. Even the CYC project [Lenat, 1990], which attempts to build a huge knowledge base of common sense information, cannot hope to represent low-level "features" that are very salient to humans like the sound of a robin's song or the color of a sunrise.

The nodes in LINK's concept hierarchy serve as its basic units of meaning. With the assumption of a static, complete knowledge base,[3] learning the meaning of an unknown word reduces to finding the appropriate node in the domain representation — in other words, a search problem.[4] The constraints of known constituents in a sentence define a subspace of the semantic knowledge base within which the referent of an unknown word must lie. Although the concepts within this space each constitute a plausible definition for the word, a *space* of possible meanings is of limited use for an NLP system. A useful hypothesis[5] would contain very few nodes, preferably one. Thus the interesting part of this problem is how to inductively select a part of the consistent space as a guess for what the unknown word can mean. The structure of the knowledge representation prescribes the manner in which that can be done.

LINK's semantic constraints are used primarily to limit the ways that parses can be formed. For example, in the sentence, "John made a deposit at the bank," the semantic constraint on "deposit" that the destination be a Financial-Institution disallows the construction of a parse with River-Edge sense of "bank". Camille uses these same constraints to direct the inference of unknown words.

The context of example sentences can provide two types of restrictions on the set of candidate hypotheses for an unknown word's meaning. First, the word may appear as the filler of a thematic role of another word, as in, "Terrorists destroyed a flarge." Because "flarge" is assigned as the direct object of "destroy", LINK's grammar suggests that it is the semantic OBJECT of Destroy. This condition places an upper bound on the generality of the word's meaning: "flarge" must be a Phys-Targ or one of its descendants in the concept hierarchy. Figure 3.1 shows the set of concepts in the hierarchy which the role-filler constraints for Destroy will allow as the OBJECT. The shaded concepts cannot represent the meaning of "flarge".

The second type of restriction that context may suggest is a filler for a thematic role of the unknown word, as in, "Frooble the building." In this case, LINK's unification grammar suggests that Building is the semantic OBJECT of "frooble." Information about role-fillers of an unknown concept places a lower bound on the specificity of the concept: given that Building is the OBJECT, "frooble" can refer to concepts like Destroy and Robbery, but not to concepts like Detonate, Die, or STrans (or any of their descendants) because Building violates the restrictions that these concepts place on their OBJECTs. Figure 3.2 shows which concepts in the action hierarchy are allowed to take Buildings as OBJECTs .

Thus, two types of information are supplied by example sentences: information which provides a lower bound on the level in the hierarchy of the meaning of an unknown word, and information which provides an upper bound. One might think that a least-commitment approach to learning, like Mitchell's *candidate-elimination* algorithm [Mitchell, 1977], would be the best way to approach this task. Mitchell's algorithm used *version spaces* to represent the set of candidate hypotheses, and slowly narrowed the version space depending on the additional constraints provided by new examples. Negative examples lowered the upper bound, and positive examples raised the lower bound.

---

[3] The completeness assumption will be addressed in section 4.4.

[4] As described in the previous chapter, the structure of the action concept hierarchy, which organizes the search space, is constructed in an ad hoc fashion from an analysis of the texts. The structure and the semantic constraints which partially determine the organization are *not*, however, constructed for the lexical acquisition task, they are constructed for semantic disambiguation in parsing for the information extraction task.

[5] Throughout this thesis, the term "hypothesis" will be used to refer to a disjunctive set of concepts that represents Camille's guess at the meaning of a word.

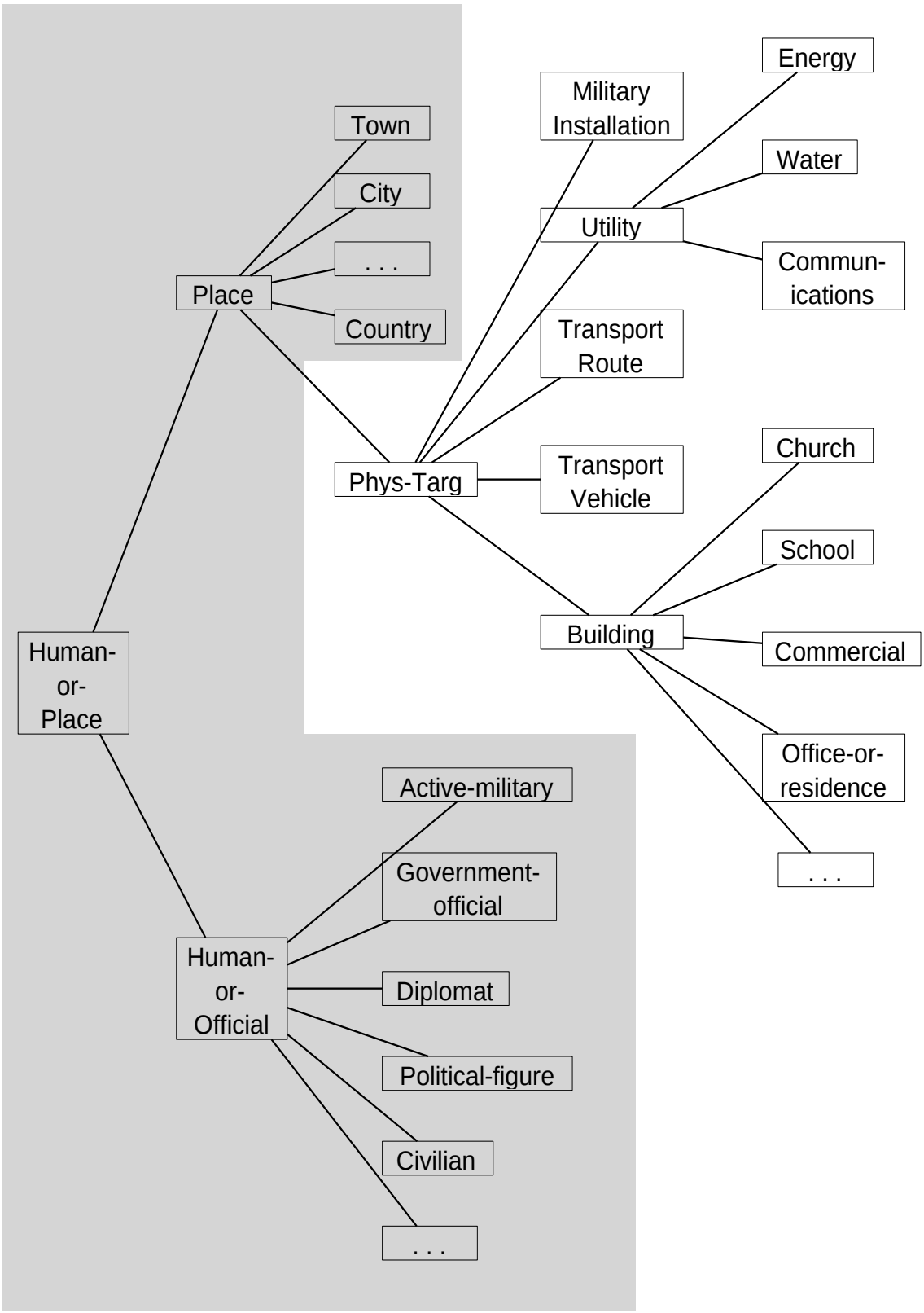


Figure 3.1: The pruned object tree

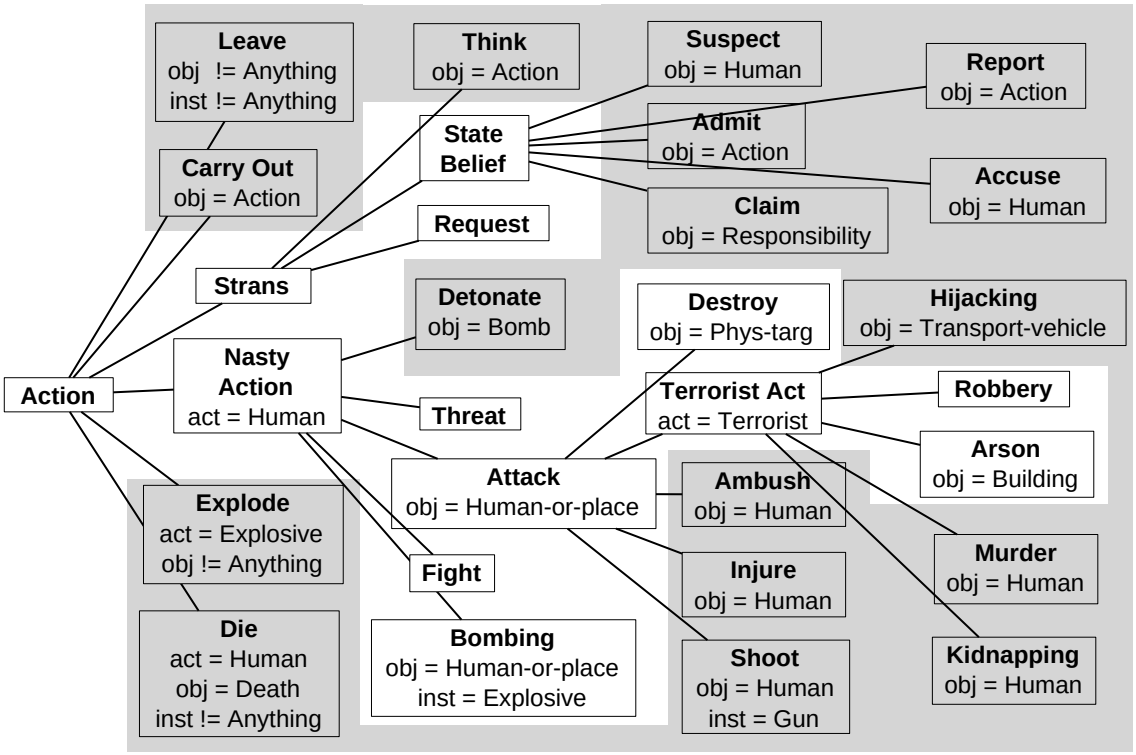


Figure 3.2: The pruned action tree

Because of the structure of the constraints, however, this approach would fail. In the word-learning task, it is usually the case that particular kinds of words only appear in examples that provide one of the two types of restrictions. Nouns, which usually refer to objects, almost always appear as role-fillers of actions or states. Thus examples of the noun in context serve only to limit the upper bound of the candidate hypotheses. Verbs, on the other hand, usually appear with role-fillers attached to them, and not as role-fillers themselves, because they refer to actions or states. For new verbs then, examples place a lower bound on their candidate hypotheses. Thus, because examples only provide one of the two kinds of restrictions for significant word classes, a least-commitment algorithm would not converge on a single hypothesis for the meaning of most unknown words.

Besides the graph-search implications of this dichotomy, there are practical ones as well, which can be seen from a pair of examples. From a sentence like, "Mary will detonate the flarge," an agent could deductively infer that "flarge" must refer to a type of Bomb. That is, the concept Bomb serves as the upper bound of the space of possible interpretations. Although there may be many different types of bombs within this space, this hypothesis has significant information content. All descendants of Bomb share certain features that an agent might like to know.

On the other hand, from a sentence like, "John froobled the pedestrian," one could deduce that "frooble" *does not* mean Hijack or Detonate, for example. The OBJECT constraints on these concepts are contradictory with the type of the OBJECT of the sentence. Thus, they form part of the lower bound of the possible interpretation for the unknown verb. If an agent only has this knowledge about a verb, however, it does not know much. The members of the space that is delineated by this lower bound might not share any attributes other than that

they are all Actions.[6] Thus, this upper bound / lower bound distinction makes a big difference in the inferable information content.

The distinction also makes it more important to learn verb meanings. The semantic constraints that serve as the leverage for learning word meaning are attached to the action concepts. If the system does not know a verb, it cannot know which constraints apply. In order to offset this lack of evidence, Camille does not use a least-commitment approach like Mitchell's. Instead it uses a simple, but critically important, rule to limit the hypothesis space:

```
For nouns, choose the most general consistent hypothesis.
For verbs, choose the most specific hypothesis.
```

Thus, Camille actually makes guesses about what the word means — guesses that, because they are extreme, are easily falsifiable. Thus additional examples of the word's use are quite likely to contradict the initial hypothesis, unless, of course, it was correct.

As previously mentioned, learning nouns is much easier than learning verbs. Some early work in learning nouns with LINK is described in [Lytinen and Roberts, 1989a]. Additions to the noun-learning mechanism are described in section 4.3.1. Because semantic constraints are attached to the verbs, it is more difficult to learn unknown verbs, therefore verb acquisition is the primary focus of this thesis. The next section describes the implementation of the initial lexical acquisition mechanism.

3.2 Camille 1.0

As described in the previous section, the meaning-inference procedure in LINK consists of a search through the concept hierarchy for the appropriate concept to represent the meaning for an undefined word. This section describes the details of this search process and how Camille limits the hypothesis set by choosing the most specific applicable concepts.

3.2.1 Enter Generic Definition for Unknown Word

As LINK parses a sentence, if it finds a word that is not in the dictionary, it looks up a special "default" definition instead. Figure 3.3 shows this special word definition which is entered into the chart like that for any other word. Several points should be noted regarding this definition. The name of the word in the definition, "undefined" (from line 1), is replaced by the actual word as it appears in the sentence. This allows Camille to posit a new meaning for the word after it processes the sentence. Lines 2–6 contain the default definition for the active transitive verb sense of an unknown word. Lines 7–10 have the corresponding intransitive description, and lines 11–13 contain the definition for the stative sense (e.g. "The bomb exploded"). Lines 15–18 contain default definitions for adjectives and nouns.

The TYPE specifications in lines 6, 10, and 14 restrict the sort of syntactic constructions that these definitions can occur in. The two latter senses require intransitive sentences. The grammar rules that create verb phrases ensure that when a noun phrase follows the verb phrase, the verb is marked as transitive. Otherwise it is marked as intransitive. Because the markers TRANS and INTRANS cannot unify, the parser will never attach a noun phrase as the

[6] The only type of input that could help solve this problem would be negative input like, "You can't say 'Mary tossed the flarge.'" Unfortunately, this does not occur in normal text. This "no-negative-evidence problem" limits what a lexical acquisition mechanism could deduce from context. It is also a problem for human language learners as discussed in section 6.2.

```
(define-word undefined                          ; 1
   (Verb (Head Rep) = (Action)                  ; 2
         (Head Undef) = (Head Rep)              ; 3
         (Head Subj Rep) = (Head Rep Actor)     ; 4
         (Head Obj Rep) = (Head Rep Object)     ; 5
         (Head Type) = Trans)                   ; 6
   (Verb (Head Rep) = (Action)                  ; 7
         (Head Undef) = (Head Rep)              ; 8
         (Head Subj Rep) = (Head Rep Actor)     ; 9
         (Head Type) = Intrans)                 ; 10
   (Verb (Head Rep) = (State)                   ; 11
         (Head Undef) = (Head Rep)              ; 12
         (Head Subj Rep) = (Head Rep Object)    ; 13
         (Head Type) = Intrans)                 ; 14
   (Adj  (Head Rep Mod) = (Modifier)            ; 15
         (Head Undef) = (Head Rep Mod))         ; 16
   (Noun (Head Rep) = Phys-Obj                  ; 17
         (Head Undef) = (Head Rep))             ; 18
```

Figure 3.3: Default definition for unknown words

object of an intransitive verb. In this way, the parser eliminates the definitions that are not consistent with the syntax of the input sentence.

Line 3 serves as a hook which allows Camille to easily access the inferred meaning hypothesis for the undefined word. Lines 4 and 5 specify that the syntactic subject of the sentence will become the semantic actor in the final meaning representation for the sentence and that the syntactic object will be the semantic object as well. These connections are crucial in inferring the meaning for an unknown word. When a phrase is attached to the verb as the syntactic object, for example, this rule also causes the representation of the syntactic object to serve as the semantic OBJECT of the verb. The filling of any such slot of an unknown verb triggers the learning mechanism. The actual operation of this mechanism is described below. Line 2 specifies the initial set of hypotheses for the meaning node for an unknown verb, namely the single concept Action. This is the only difference between the representation in the chart for known words and the representation for unknown words. Entries for defined words refer to a single concept in the hierarchy. Undefined word entries have a set of labels which correspond to the active hypotheses for the meaning of that word.

3.2.2 The Refinement Tree

When the set of concepts that comprise the domain knowledge (part of which was displayed in figures 2.1 and 2.2) is loaded into the parser, the resulting network is processed into an additional structure called the Refinement Tree. This structure facilitates the first stage of the inference mechanism. It is basically a discrimination net that has pointers from general nodes in the semantic hierarchy to more specific nodes based on the slot filler that is

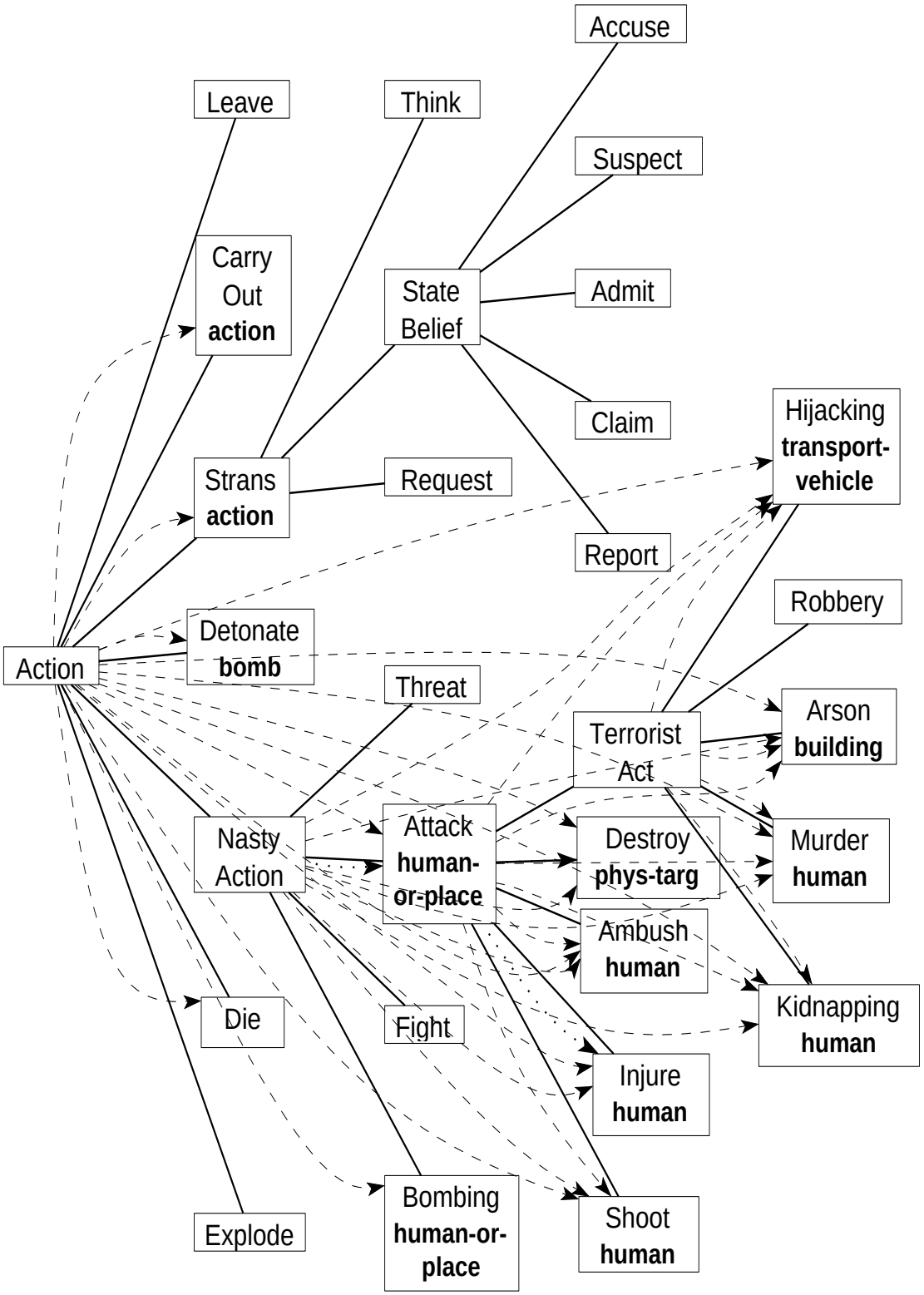


Figure 3.4: Partial discrimination net

attached to the verb.[7]

Figure 3.4 gives an overall idea of the structure of the Refinement Tree (for clarity, a portion of this tree has been extracted and displayed in more detail in figure 3.5). The portion of the RT that applies to OBJECT slot fillers is superimposed over the action hierarchy (which also displays OBJECT constraints). The RT is used to suggest inductive leaps. It says, for example, if the current verb meaning hypothesis is Action, and the OBJECT is a Transport-Vehicle, then replace Action with Hijacking as the new hypothesis.

An example will help illustrate the use of the RT. Assume the parser is processing the sentence, "Terrorists froobled a bus yesterday in Geneva." As soon as the parser attaches the noun phrase "a bus" as the OBJECT of "froobled", Camille is triggered. It takes the current hypothesis for "frooble" (which will be Action if it has not been encountered before), and checks the RT to see if there is a pointer for the OBJECT slot for a filler of type Bus. Because Bus `is-a` Transport-Vehicle, the link to Hijacking is found. This becomes the starting point for the search for the meaning of "frooble". The search will be further described in the next section.

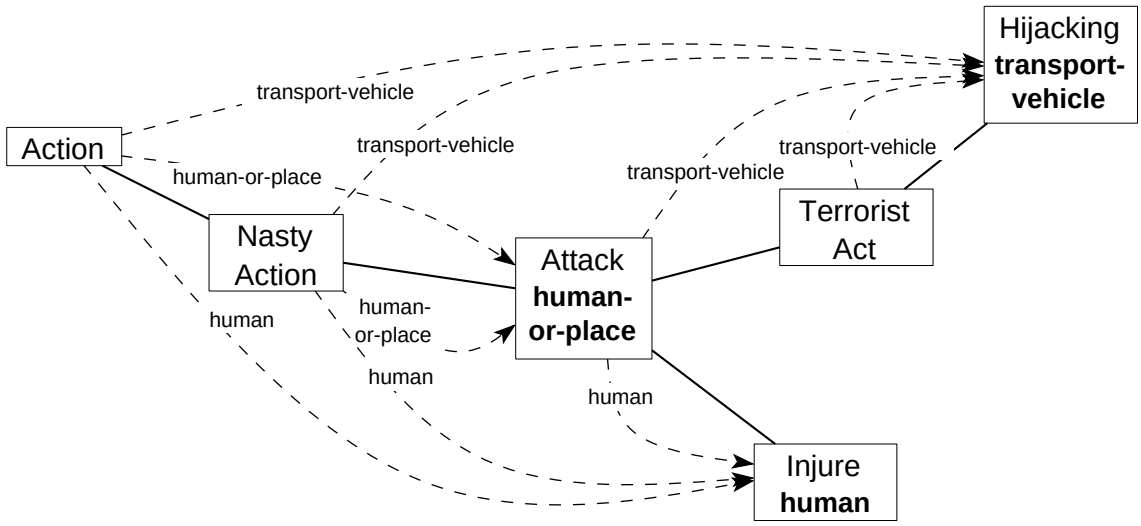


Figure 3.5: A smaller portion of the refinement tree

Camille uses the Refinement Tree for two reasons. First, it reduces search through the space of semantic nodes. The second reason relates to the nature of the information provided by the semantic constraints as discussed in section 3.1. Because there is no upper bound on the search space for a verb's meaning given one of its slot fillers, the RT allows Camille to inductively set a working upper bound. The location in the semantic hierarchy that an RT link points to is the starting place for the search for a meaning hypothesis. If Camille does not find a descendant of this node whose constraints match the slot fillers, then the upper bound will be lifted and the search expanded.

Although the Refinement Tree helps Camille reduce the hypothesis set, the use of the structure is not crucial, only the *function* it performs is. That same function could be served by search. An alternative mechanism would search through the Action hierarchy and select nodes whose constraints matched the type of the slot filler. Such a search would be expensive,

[7]This produces a reinterpretation of the domain knowledge, focusing not on the `is-a` relationships, but on the relationships defined by the semantic slot-filler constraints.

however, to perform every time a slot filler is attached. The Refinement Tree allows Camille to compile out that search process.

3.2.3 The Algorithm

In this section, the basic learning system is described. The presentation of Camille's learning algorithm is followed by examples of its use.

Figure 3.6 shows a flow chart that describes the operation of the basic word-learning mechanism for verbs. (As previously mentioned, learning nouns comes as a natural by-product of applying semantic constraints. Therefore, no additional mechanism is required.) An important point about this procedure is that instead of using a single concept to represent the meaning of a word, Camille maintains a set of hypotheses that are consistent with the slot fillers. This set does *not* comprise the entire space of consistent hypotheses. Initially, it is the subset indicated by the Refinement Tree. If there are consistent descendants of these concepts, they replace their ancestors. If not, the `Generalize` procedure is invoked to find a set of hypotheses that is consistent with the slot fillers.

As previously mentioned, the learning method is triggered by the attachment of a new slot filler to an unknown word. This is done by checking the current set of slot fillers for the new word to see if they have changed. If there is a change, then for each of the current hypotheses, the refinement tree is checked using the new slot filler to see if any more specific concepts are suggested. If so, the set of new, more specific hypotheses replaces the original set.

Consider the sentence, "Terrorists froobled the building." The parser starts with a chart very much like the one shown in figure 2.5. Instead of just one entry in the second position, however, the chart will contain a different entry for each portion of the undefined word definition in figure 3.3. The non-verb definitions will be eliminated eventually because a sentence must have a verb to be grammatical. The intransitive definition will be eliminated because it won't be able to combine with the following noun phrase. This elimination process is part of the standard parsing mechanism and happens with all ambiguous words. Figure 3.7 shows the chart with the portion of the undefined word definition that will eventually be successful.

The parser realizes that it has an unknown word in the chart and every time it fires a grammar rule checks to see if the undefined word's slot fillers have changed. When the verb phrase rule attaches the succeeding noun phrase to the dag representing "froobled" (figure 3.8), the Building is attached as the OBJECT of the undefined action, and Camille is triggered.

After applying the refinement procedure to the initial Action hypothesis with the attachment of the OBJECT Building, the set of remaining hypotheses is: (Attack Bombing Destroy Arson). Next, from this set, all ancestors are eliminated. That is, if the set contains both an ancestor and one or more of its descendants, the more general concept is removed. This is another way of pushing the hypotheses down the tree. In this example, Attack is eliminated because Bombing and Arson are its descendants.[8]

From the reduced hypothesis set, Camille attempts to force the hypotheses even further down the tree by selecting the leaf nodes under the current hypotheses for which no other constraints are violated. Each current hypothesis is checked to see if it has any leaf-node descendants whose constraints are not violated by the accumulated slot-fillers. If so, it

[8] Note that this is a very important step because the most general action node, Action, can accept any Object as its semantic object. If Camille did not eliminate ancestors, it would end up with a search that was virtually unconstrained in the upper bound.

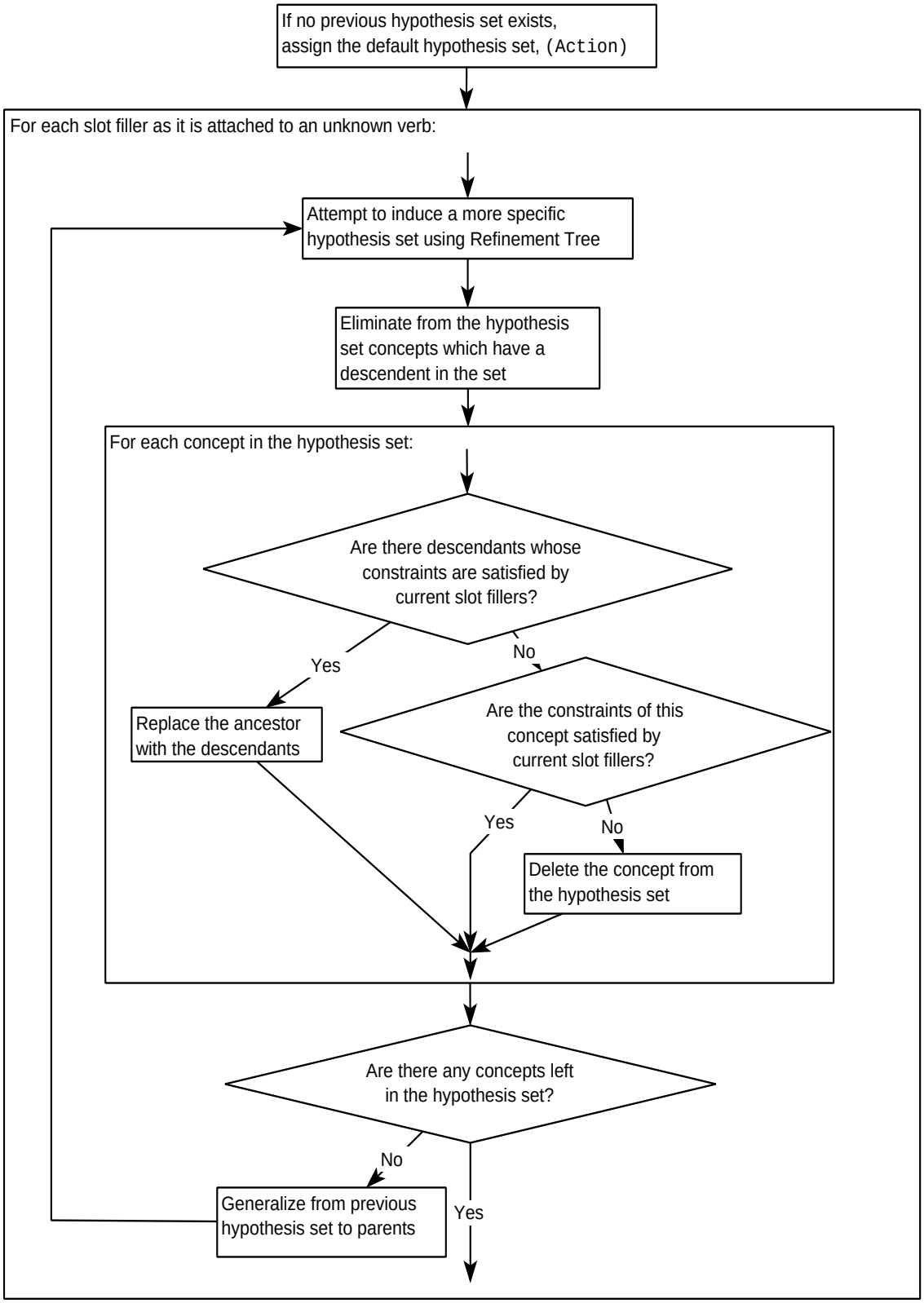


Figure 3.6: The basic word learning algorithm

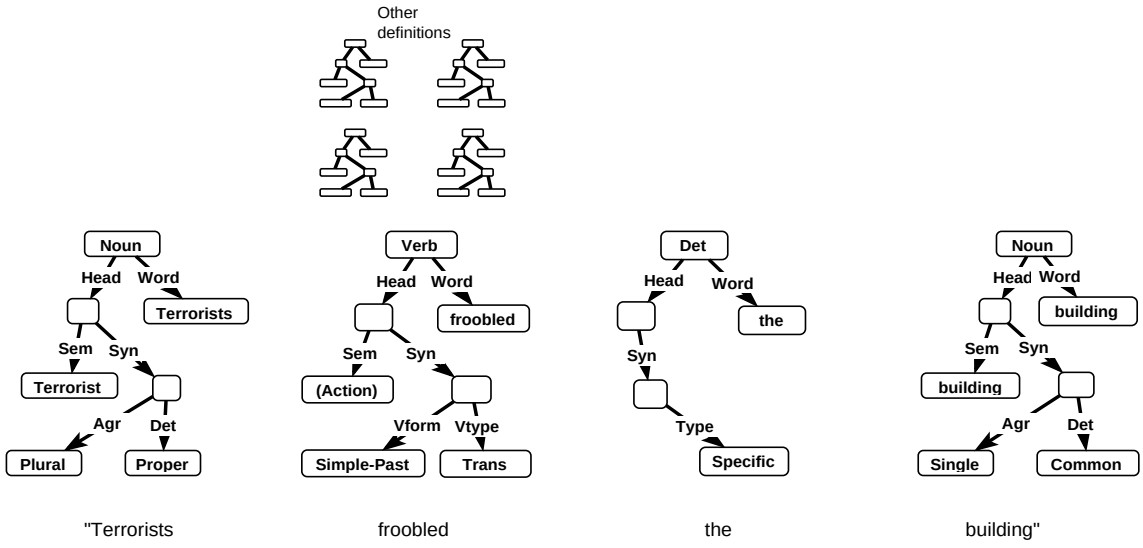


Figure 3.7: An initial chart with an unknown word

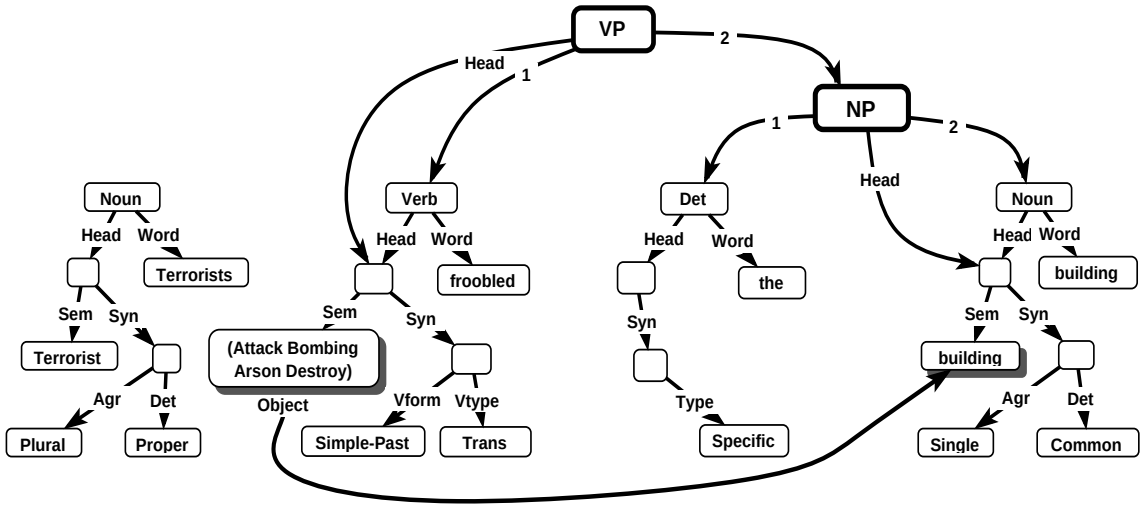


Figure 3.8: The chart after attaching "the building" to "froobled"

is replaced by those leaves. If not, it is checked to see if it has additional constraints that are violated and in that case, it is eliminated from the hypothesis set.

The final parse is shown in figure 3.9. Note that the semantic node for the head of the sentence has the label (Arson Attack Bombing Destroy). This set is the final set of hypotheses for the meaning of "frooble" inferred by Camille from the sentence.

After parsing a sentence with an unknown word in it, the system stores the word's inferred meaning hypotheses in the lexicon. The definition is similar to that for any other word, with the exception that, again, instead of specifying a single concept for the meaning of the word, the definition specifies a list of concepts. This signals Camille that the definition is tentative and can be further refined.

If additional sentences using the word are parsed, the refinement process continues,

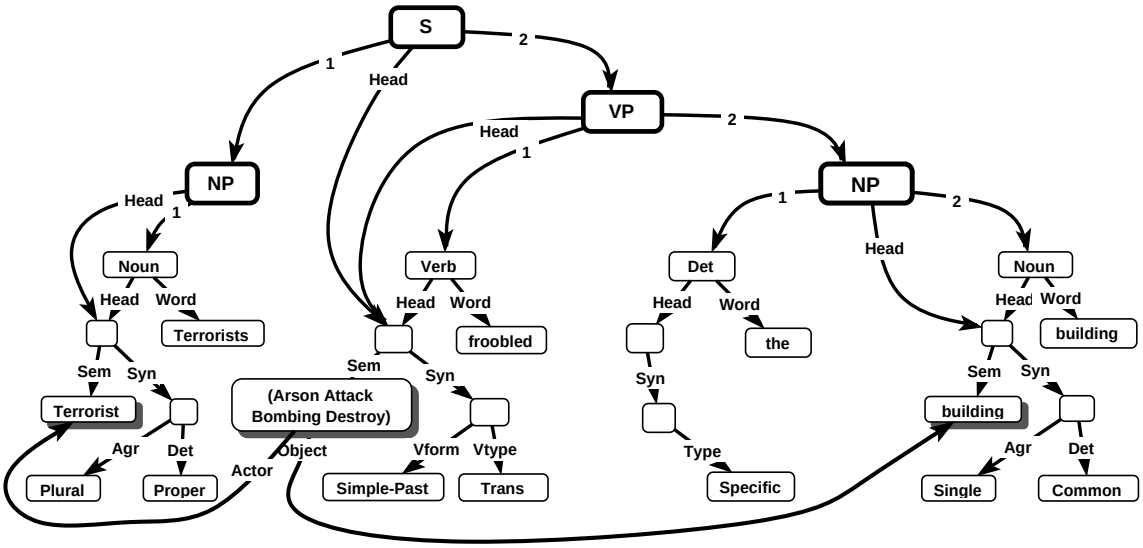


Figure 3.9: The chart after attaching "Terrorists" to "froobled the building"

starting with the stored hypotheses. In some cases, all of the original hypotheses may turn out to be too specific; that is, the set of constraints on the hypothesized concepts are violated by the current slot fillers. The appropriate meaning for the unknown word must be somewhere else in the concept hierarchy. When this happens, the `Generalize` procedure (`Generalize` is schematically depicted in figure 3.10) is called to travel up the tree, gradually expanding the search space starting from the original hypotheses. The set of former slot-fillers is combined with the new ones, so that the constraints of new hypotheses will be met by all of the available evidence.

For example, assume the system has already processed the sentence described above, "Terrorists froobled the building." It has reached the hypothesis that "frooble" means Arson. Now it gets another sentence, "Terrorists froobled the pedestrians." When "pedestrians" (with semantic representation Civilian) gets attached as the OBJECT of "frooble", the inference process begins anew. There are no refinements from Arson, so we go to the next step. Camille checks for more specific concepts. There are no consistent descendants of Arson either. So Camille checks the constraints of Arson itself. Because Civilian is not a Building, this node is not legal either, so `Generalize` is called. `Generalize` checks the parents of the current hypotheses and collects constraints as it ascends. The parent of Arson is Terrorist-Act, so `Generalize` checks to see if there are any concepts under Terrorist-Act whose constraints accept the current slot filler, (OBJECT = Civilian), as well as the previous constraint, (OBJECT = Building). In this case, the node Robbery, because it has no additional constraints, inherits the OBJECT constraint from Attack, Human-or-Place. Both Civilian and Building satisfy this constraint, so Robbery becomes the new hypothesis for the meaning of frooble.

If no legal node were found at this point, `Generalize` would be recursively called, expanding the search to larger and larger portions of the tree. There are two important consequences of this approach. First, the learning mechanism is incremental. Instead of starting over from the default every time new information is found, Camille takes up the search from the point of its previous hypotheses. As far as computational efficiency is concerned, this approach is better than the alternative because the search is more limited and the system need not store all the previous information.

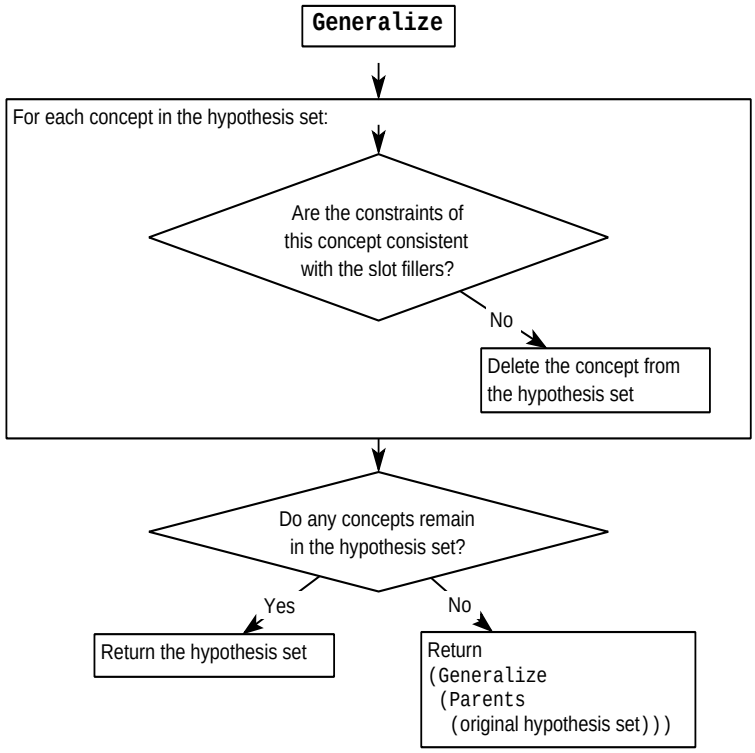


Figure 3.10: The `Generalize` function

The second consequence is a comment on the efficacy of this approach. In general, this procedure is effective because of the nature of verbs and the frequency with which they occur. The more general verbs, that is those which can take many different types of slot-fillers, tend to occur more frequently than specific verbs. Thus with a "specific" unknown verb, Camille wins even if it is encountered only once or twice because it chooses the most specific hypothesis. When Camille encounters a verb with many different types of slot-fillers, the generalization procedure infers a more general hypothesis.

In summary, the learning procedure attempts to counteract the one-way-constraint problem described in section 3.1 by constantly trying to find the most specific hypotheses possible. This has two advantages. First, a specific inferred meaning is inherently more useful than a general one because it contains more information. Second, the more specific the hypothesis is, the more falsifiable it is. The incremental nature of the learning process is exploited to ensure that even if Camille initially infers a hypothesis that is too specific, later evidence can disprove that hypothesis and allow the system to make another, better-informed guess.

3.2.4 Empirical Evaluation

In order to evaluate Camille's learning abilities, a series of tests was performed. The first domain in which Camille was tested was the Assembly Line domain. A description of the task and knowledge representation for this domain is in Appendix B. In order to improve the flow of the thesis, the specific details of the test results have also been placed in the appendix. Summaries of the evaluation will be included in this chapter and the next to aid in the analysis

of Camille's strengths and weaknesses.

In the test of the initial Camille implementation, a set of 100 sentences from the Assembly Line corpus and 50 sentences from the Terrorism domain were chosen at random.[9] The definitions of all of the verbs that appeared in the test sentences were removed from LINK's lexicon. The sentences were then processed in turn by Camille, and the resulting word definitions written to a file. Camille 1.0 hypothesized 22 verb definitions in the Assembly Line domain and 17 in the Terrorism domain.

For 18 of the verbs (82% of the set of 22) in the Assembly Line corpus, the appropriate concept was included in the hypothesis set that Camille inferred. In the Terrorism domain, 8 of the 17 verb meanings, or 47%, were correctly inferred. These results were encouraging[10] and led to several conclusions about the nature of the learning mechanism.

The first conclusion was that the system seemed to be learning quite well. This was emphasized by an analysis of the corpora that were tested. Because Camille learns incrementally, the number of instances of a word in example sentences is an important factor in the system's performance. As detailed in section B.2.4, the mean number of occurrences of the unknown verbs in the test sets was 3.7 and 2.7 for the different domains. In both domains, however, the median was 2 repetitions, and a large percentage of the verbs occurred only once. This meant that Camille had little evidence on which to base its conclusions. Despite this paucity of evidence, Camille still performed quite well.

Another attribute of the corpora, especially in the Terrorism domain, was the difficulty of the test sentences. LINK produced a complete parse in only 1 of the 50 test sentences in this domain. Although Camille uses a post-parse analysis of the chart to extract useful fragments, it is frequently unable to extract all of the verb's arguments and thus Camille has limited evidence from which to make its inferences.

The second conclusion from the initial test was that the results were good but could be better. There were only a small number cases in which Camille chose a single hypothesis for the word's meaning. In the other instances, a set of hypotheses was chosen. For the Assembly Line domain, there were on average 6.2 concepts per word, and 3.2 for the Terrorism domain. So in a sense, this initial version of Camille caught a lot of fish, but did it by throwing a very big net.

Part of the reason for this behavior came from the fact that the domain knowledge did not contain enough information to distinguish between many of the concepts. For example, in figure 3.2, the two concepts Murder and Kidnapping both take Human OBJECTs and Terrorist ACTORs. There is no additional discriminating information in the semantic representation because it is not needed to parse the sentences in the domain. Unfortunately, this leaves Camille unable to distinguish between these concepts. The ensuing enhancements to Camille (which will be described in the rest of this chapter and the next) were primarily aimed at exploiting more efficiently the available knowledge or increasing the knowledge available.

---

[9] The test sets were kept relatively small to simulate a sparse-input learning task. The assumption is that most of the lexical definitions have been entered as part of the knowledge engineering of the system. Words that were overlooked in this process are not likely to be encountered frequently by the system, so the testing set contains a small number of examples of each word. The evaluation of Camille's performance on larger test sets is mentioned in section 3.4 and described completely in section B.2.5.

[10] Unfortunately, the only other lexical acquisition systems which were tested on real-world input, MayTag and AutoSlog, did not learn verb meanings to the same extent that Camille did. Thus, the testing results of these systems are not directly comparable. Chapter 5 contains a qualitative comparison of these systems with Camille.

The tradeoff between inferring correct hypotheses and limiting the number of hypotheses generated also resulted in the use of a more discriminating scoring mechanism, adapted from the MUC conferences [Chinchor, 1992]. These measures, Recall and Precision (originally taken from the field of Information Retrieval), are defined below. Two other calculations, Accuracy and Production, describe respectively the system's performance on the hypotheses it made (as reported above), and the percentage of possible verbs for which it produced hypotheses. A final measure, Parsimony, shows the percentage of definitions Camille made that were exactly right, i.e. where the correct concept was only concept generated.

- CONCEPT: a single concept from Link's domain representation
- HYPOTHESIS: a set of CONCEPTS that constitutes Camille's definition for an unknown word
- CORRECT: the number of words for which a correct CONCEPT was included in the HYPOTHESIS
- ONE-CONCEPT-CORRECT: the number of words for which the correct CONCEPT was the only member of the HYPOTHESIS
- CONCEPT-SUM: the sum of the concepts generated for all of the words
- GUESSES: the number of HYPOTHESES generated
- POSSIBLE: the number of undefined words which could have been assigned HYPOTHESES
- RECALL: CORRECT / POSSIBLE
- PRECISION: CORRECT / CONCEPT-SUM
- ACCURACY: CORRECT / GUESSES
- PRODUCTION: GUESSES / POSSIBLE
- PARSIMONY: ONE-CONCEPT-CORRECT / POSSIBLE

Recall is similar to the score described above, Accuracy, except that the denominator is the total number of words that could have been hypothesized from the test set instead of the number that were actually hypothesized. Precision scores increase as the number of superfluous hypotheses goes down. Production rates how well the system does at producing *some* hypothesis from the test set.

In the Assembly Line test, Camille scored a Recall of 51%, a Precision of 13%, a Production of 63%, (as reported above) an Accuracy of 82%, and a Parsimony of 9%. The Terrorism test produced scores of 33% Recall, 15% Precision, 67% Production, and 6% Parsimony. The low Precision scores reflect the fact that Camille was producing a large number of hypotheses per word.

3.2.5 Assumptions and limitations: Camille 1.0

In the initial test, several of the words could not be correctly inferred because they were ambiguous. Camille 1.0 had difficulty inferring the correct meanings of ambiguous words, because it had no easy way of forming disjunctive hypotheses. Methods of addressing this difficulty will be described in section 4.3.

Another obvious limitation of this initial system is that it assumed that every aspect of meaning about the domain was *a priori* represented in the concept hierarchy. This has two consequences: First, a new word could only be assigned to an existing node in the hierarchy. There is no mechanism for adding new elements of meaning. Second, there was no way to refine the meaning of a given node with additional information. These two factors limited the type and value of the inferred meanings. Thus, although the system could learn the meanings of words it didn't know, it couldn't learn new concepts. Within the context of the information extraction task, where the type of knowledge about the domain is quite limited, this is not critical. In order to apply the system to other tasks, concept learning would be useful, and it is discussed in section 4.4.

Another limitation is the extent of the knowledge which is used to influence the learning process. The initial system used only information from within a sentence to make its inferences. Chapter 4 describes the extensions made to utilize additional domain knowledge.

3.3 Camille 1.1: More specific concepts

As mentioned above, the initial system worked fairly well in that it found the appropriate concepts for the words to map to, but it found many others as well. Thus, the hypotheses that were produced were of little value. In order to reduce the number of hypotheses, Camille was extended to make it rank the remaining set due to the tightness of fit between the constraint of the hypothesized concept and the slot filler in the example sentence. For example, the concept Arson takes an object that is a Building, and the concept Bomb takes an object that is a more general concept, Human-or-Place. Therefore, given an example sentence with an instance of a building as the object, the system regards Arson as a "tighter fit" than Bomb.

After measuring all the distances between the fillers and the constraints, Camille removes from consideration all but the tightest matches. Note that this is another inductive step. Either hypothesis in the example above is consistent given the evidence; consistent, but not very useful. By eliminating lower-ranked hypotheses, the system induces meanings that are more useful since they are more specific and therefore have a higher information content. In general, more specific hypotheses are more *falsifiable*. In other words, because Building IS-A Human-or-Place, there must be at least as many instances of Human-or-Place as of Building. Therefore, future example sentences are more likely to violate Arson's constraints than Bomb's — unless, of course, the hypothesis for the word is correct.

Because the testing protocol for Camille was developed after several of the versions were completed, it was impossible to test this version independently of the next version of the system, Camille 1.2. Thus the results shown in the next section reflect improvements based both on the elimination of less specific concepts and the improvement of the system's instance memory.

Unfortunately, given the granularity of the semantic representation for text analysis in these domains, many sets of concepts remain that are indistinguishable by their constraints. Therefore, even this more particular version of the system infers many hypotheses for a word's

meaning. Further efforts to address this difficulty will be discussed in section 4.2.

The initial inferences that the Camille 1.1 made were good, but it was possible for the inference process to become confused by multiple instances of a word's use. When Camille infers a word meaning that is later proved to be incorrect, the initial inferred concept is taken as the starting point of the search for a more suitable hypothesis. As described above, the `Generalize` procedure ascends the tree[11] to begin the search for a new concept to map the word to. On its way up the tree, the initial version of this procedure collected the constraints of the parent nodes. For example, if Camille 1.1 started with the Arson concept, and got a sentence with an object that was a Person, the original hypothesis was no longer valid. The system checked the parent of Arson, Terrorist-Act, to see if it had legal descendants. It took the constraint from Arson, i.e. that the OBJECT was a Building, to apply to future hypotheses.

At this point, Camille 1.1 examined the constraints of the parent node to see if its constraints were violated by the accumulated evidence. If so, the `Generalize` procedure was recursively called. If not, as it does normally for unknown words, the system tried to find any more specific nodes whose constraints were not violated by the accumulated evidence. With the current example, the system would find the Robbery node that can be applied to both people and buildings.

The problem with this approach is that given another instance where constraints are violated and `Generalize` must be called, the system lost information. The only constraint on the OBJECT of a Robbery is that it is a Human-or-Place, but this information is stored at the Attack node. `Generalize` could select any of the siblings of Robbery since it no longer remembered the previous examples of the word's OBJECTs. Note that one alternative is to collect the most specific constraints of all the parents using them to check new hypotheses, but this would result in hypotheses that were too general — no concept more specific than the one with the constraint for a slot filler could be chosen. The other alternative is to change what the system views as a hypothesis about a word's meaning. This approach is discussed in the next section.

3.4 Camille 1.2: Remembering slot fillers

Camille 1.2 extended the hypothesis structure in the following way: Instead of just saving the concept that the word refers to (as well as the associated syntactic information), the structure of the hypotheses that the system keeps was extended to contain a record of the specific slot fillers that it had encountered in example sentences. This allowed the `Generalize` procedure to use its "memory" of the prior examples of the word's usage when searching for a new hypothesis. Thus Camille 1.2 was assured of inferring a meaning that was consistent with the current *and* prior evidence.[12]

Another potential approach to this problem would be to save more of the final structure of the parse rather than just saving the slot fillers. If the DAG:

---

[11] Because all descendants of a node inherit the parent's constraints, any example sentence that violates the parent's constraints must also violate the descendant's.

[12] This reduces one of the advantages of having an incremental system: the saving of space. The algorithm remains incremental, however, and saves search time. Furthermore, it saves space over an implementation that stores every entire parse or example sentence for each word.

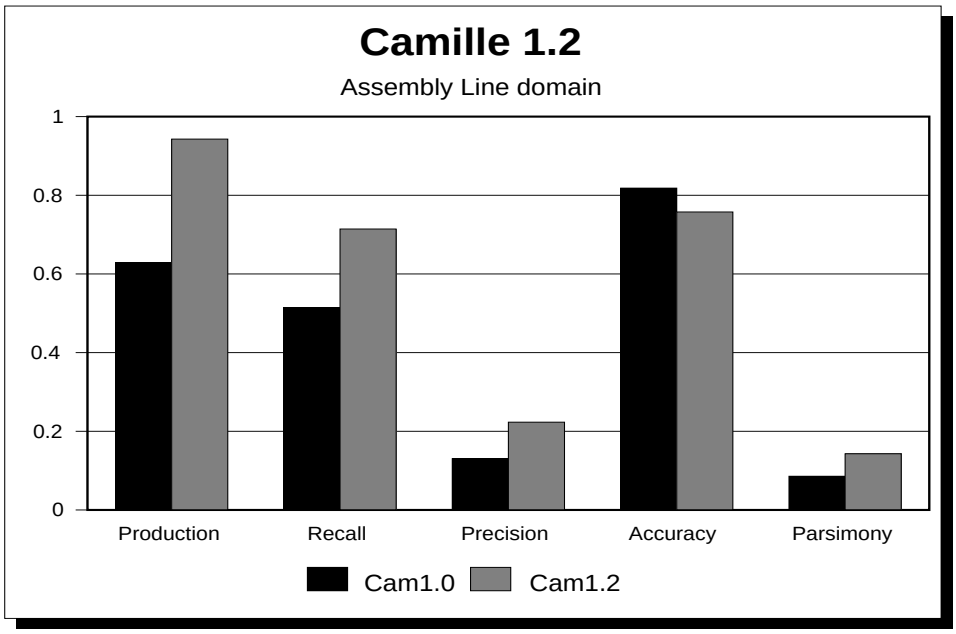
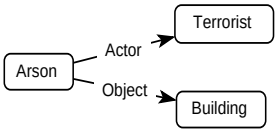


Figure 3.11: Camille performance, Assembly Line domain



were saved as the meaning of frooble for the sentence, "Terrorists froobled the building," then, hypothetically, unification could ensure that all the constraints were fulfilled. Although this would be an elegant solution, it would not work. Extending the previous example with another instance of an unknown word with a Civilian as its OBJECT, the system would have to try to unify Building and Civilian. Because neither is subsumed by the other, unification would fail, and the inference procedure could not continue. In effect, this would be applying the constraint to the conjunction of the two slot fillers. Instead Camille will apply the constraints to each slot filler separately. If each one satisfies the constraint, then the current hypothesis is consistent with the evidence that has been encountered.

This version of Camille achieved a Recall of 71%, a Precision of 22%, a Production of 94%, an Accuracy of 76%, and a Parsimony of 14% in the Assembly Line domain. As displayed in figure 3.11, this represents a significant increase in Camille's performance. Production, Recall, Precision, and Parsimony all increased (by roughly 50%, 40%, 70%, and 56% respectively).

In the Terrorism domain, the test results told a more complicated story as shown in figure 3.12.[13] Camille's scores (41% Recall, 19% Precision, 99% Production, 47% Accuracy, and 18%) showed an increase in Precision, but a slight decrease in the other measures. This is a typical case of what the Information Retrieval field calls the Recall / Precision tradeoff. If a system generated every possible concept as the meaning for each unknown word, it would be

[13]Section 4.6 contains an analysis of the differences between the domains and the effects that they had on the hypotheses inferred by Camille.

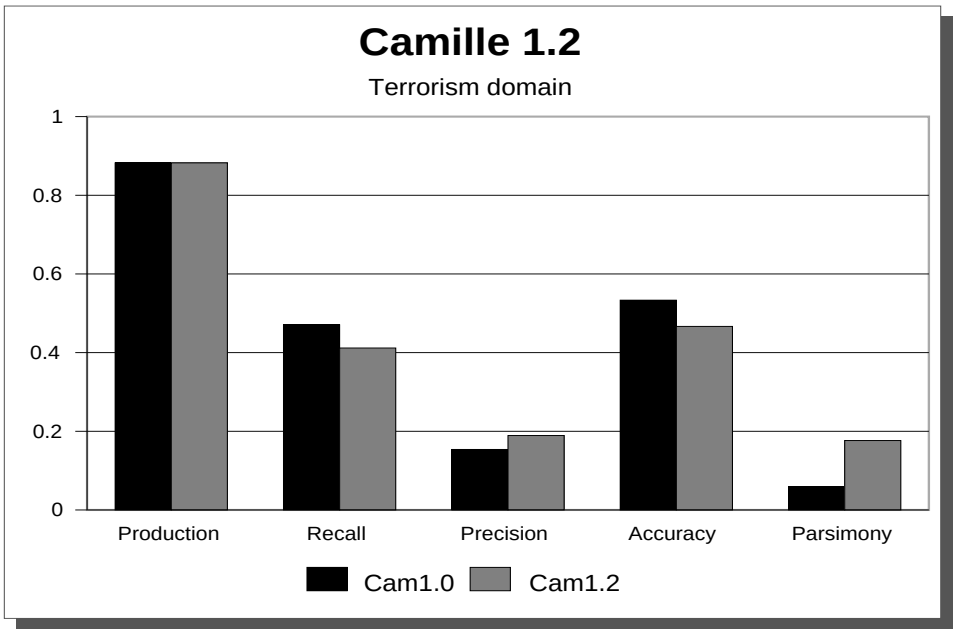


Figure 3.12: Cumulative Camille performance, Terrorism domain

bound to get 100% Recall — but very low Precision. By reducing the size of the hypothesis sets, Precision (and Parsimony) increases. Sometimes, however, a correct concept is removed from a hypothesis along with the incorrect ones. This results in reduced Recall (and Accuracy) rates. The most straightforward method for Camille to increase its Recall (without decreasing Precision), would be to obtain more instances of the example sentences. Because Camille is a non-interactive system, however, it cannot control its input.

This version of Camille was also evaluated on larger tests with the expectation that increasing the number of repetitions of each word would increase the likelihood that Camille would correctly infer meanings. The results said more about the complexity of the domain than the performance of the system however. After processing 100 sentences, Camille's scores *dropped* significantly, to 29% Recall and 16% Precision. After 150 sentences, the score was down to 24% Recall and 13% Precision. At this point the score seemed to bottom out, with 50 more sentences producing no further decline.

The reason for this decline was the difficulty that the parser had in creating a successful parse. The noise generated by incorrect parses not only hampered Camille's ability to infer meanings, it seriously degraded it. For example, from a sentence like, "Authorities have officially reported that several bank offices were adversely affected tonight in the ...," the parser attached "bank offices" as the OBJECT of "reported". This caused Camille to infer that "reported" meant Arson.

In order to isolate the effects of the parser on the learning mechanism, a set of test sentences was "hand-parsed" to extract the correct slot fillers.[14] The correct argument structure for the previous example would be: ("reported" (OBJECT . IGNORE-ACTION) (ACTOR . GOVERNMENT-OFFICIAL)). Testing on these structures showed the expected increase in performance, from 59% Recall and 32% Precision for 50 sentences to 71% Recall

[14]The resulting structure was similar to those used by Salveter, Selfridge, and Siskind as described in the chapter 6.7.

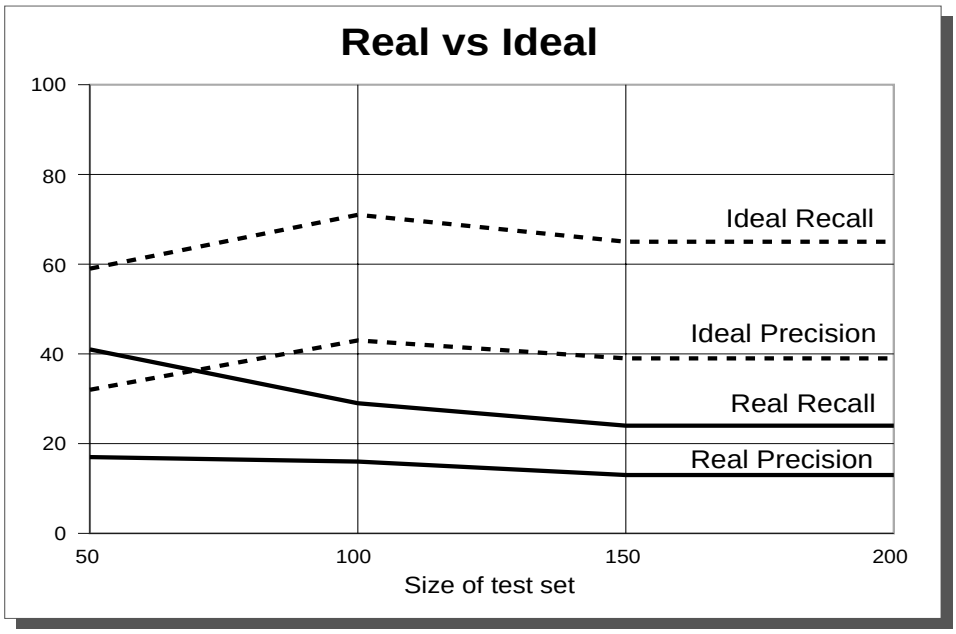


Figure 3.13: Comparing parsed input to provided input

and 43% Precision. Figure 3.13 compares the Recall and Precision for the "real" results (using parses produced by LINK) and the ideal results (using the hand-parsed structures). The reduction seen after 100 sentences in the ideal version is largely due to mismatches between the expected argument structures and some actual usage. For example, the constraint on the ACTOR of State-Belief is that it is a Human (including Organizations). A sentence like, "The constitution stated that . . . " produces conflicting evidence for Camille's hypothesis that "stated" means State-Belief.

3.5 Analysis of Camille's basic implementation

The task of lexical acquisition for Camille reduces to searching for an appropriate node in the domain representation. This abstraction of the task reveals an important distinction between learning nouns and learning verbs. The constraints on actions provide a natural upper bound on the interpretation of unknown object labels. For action labels, no such upper bound exists. Thus, in order for Camille to make useful inferences about verb meanings, it must inductively limit its search space. Camille does this by choosing the most readily falsifiable hypotheses. This gives Camille the best chance for correcting its mistakes. Thus the system can quickly converge on an appropriate hypothesis for many unknown words.

This approach to lexical acquisition is incremental so its processing and storage requirements are minimized. The system learns automatically from example sentences so it does not require guidance from a human trainer. Camille doesn't need additional knowledge sources. It uses only the knowledge that is present for standard parsing.

Camille's implementation was an evolutionary process based on analyses of its performance in empirical testing. The major steps in the evolution were:

- The initial system: The Refinement Tree was used to inductively set an initial upper bound on the interpretation of verb meanings. From the resulting subspace, the most

specific concepts were chosen which had constraints that were satisfied by the slot fillers. If the initial guess was wrong, the `Generalize` procedure started the search anew, maintaining the constraints for the previous hypothesis.

- Weak hypotheses removed: After the most specific consistent concepts were found, they were ranked based on tightness of fit between the slot filler and the constraint. Only the tightest were kept.

- Improved memory: The hypotheses were extended to maintain the semantic types of the slot fillers that were attached to them. This allowed the `Generalize` procedure to avoid rechoosing previously rejected hypotheses and select ones that were consistent with all of the examples of the word's use. The record of slot fillers was also crucial for the approach to recognizing ambiguity that is described in the next chapter.

The basic Camille approach does have some weaknesses. The production of large sets of concepts in hypotheses was not completely mitigated by the elimination of less-specific concepts that was described in section 3.3. Many sets of concepts remain that are indistinguishable based only on the use of slot fillers. Section 4.2 describes one mechanism for further refining hypotheses.

The learning procedure is sensitive to noisy input. Because it uses an inductive procedure, Camille assumes that if one of its hypotheses conflicts with subsequent evidence, then the original guess was incorrect and the hypothesis should be altered. Noise can be produced by a number of sources, most commonly incomplete parses and ungrammatical input. The test domains in this thesis contained mostly grammatical text. The Terrorism corpus was so complex, however, that it caused great difficulty for the parser, and incorrect or incomplete parses were common. Noisy input can cause Camille to infer that a word takes a larger range of slot-fillers. As a result, the system will make an overly general hypothesis for a word's meaning. One approach to handling noise is suggested by the Camille variation for dealing with ambiguous words which will be described in the next chapter. The implementation of this addition is left to future research.

As just mentioned, Camille has difficulty learning ambiguous words. If a verb occurs with two distinct types of slot fillers, the system will search for a concept that can accept the least upper bound of the fillers for that slot. Section 4.3 describes a mechanism for recognizing situations where fillers separate into two distinct classes, and then hypothesizing multiple definitions. The section further describes a method of inferring meanings for ambiguous nouns.

Because Camille was implemented with the goal of using only the knowledge that LINK requires for parsing, it is unable to make certain inferences about word meaning. The representation for action concepts describes only their names, their IS-A relationships to each other, and their constraints on slot fillers. Although the script mechanism described in section 4.2 allows Camille to make inferences based on sequences of actions, the system has no knowledge of the results of actions, their causes, or what goals they might achieve. The addition of such knowledge would enhance Camille's learning abilities, but it would also impose an additional resource requirement.

CHAPTER 4

CAMILLE 2: VARIATIONS

In the development of Camille, the initial emphasis was on exploiting the linguistic constraints within a sentence and perfecting the search mechanism that relies on those constraints. The previous chapter described the initial implementation and the enhancements that were made to Camille to fully exploit all of the available intra-sentence information.

This chapter describes variations on the basic graph search mechanism. The first section describes the implementation of a technique taken from current Psycholinguistic theories, Mutual Exclusivity. This mechanism is proposed as a way that children can decrease the complexity of their word-learning task, especially for learning object labels. Implementing it within Camille allows examination of the technique as applied to labels for actions. Furthermore, because it requires a strict formalization of the theory, it brings up additional issues that have not been addressed in the Psycholinguistic literature.

The next section describes enhancements which exploit the connections between sentences in a dialogue. The use of scripts, which describe canonical sequences of actions, allows NLP systems in general to infer missing information from default values. Scripts allow Camille to make inferences based on the co-occurrence of actions in texts.

The third section describes the enhancements that were made to Camille to enable it to deal with a problem described in the previous chapter: ambiguous words. These enhancements allow Camille to learn ambiguous words by splitting its hypotheses given sufficient evidence that there are distinct senses of those words.

The fourth section describes a similar approach to expanding the concept hierarchy. If learning a new word suggests to Camille that its domain representation is incomplete, an additional concept can be added to the hierarchy.

The final section of this chapter contains an analysis of the advantages and disadvantages of these enhancements to Camille. It also describes the features of the different test domains that made them amenable to particular word-learning techniques.

4.1 Camille 2.0: Mutual Exclusivity

The Camille 1.2 implementation could be misled into inferring the same meaning for many different words. For example, if it processed a news story that chronicled a series of attacks on a certain building, Camille, because its closest object constraint for Building is Arson, would infer that all the actions were Arsons, even if there were bombings, machine-gun attacks, etc. A similar problem occurs if there are few examples of a word's usage. For example, if Camille processed only one example of the use of the word "rob", and that sentence had a Building for an OBJECT, then it would infer that "rob" meant Arson. In order to infer the proper concept, Camille needs the additional evidence that "rob" can also be applied to people.

When children learn language, they are faced with a similar problem. They are presented with a large number of words and a large number of possible referents of the words. Psycholinguistic researchers have suggested one mechanism that children might use to overcome this computational difficulty: the Mutual Exclusivity constraint [Markman, 1991]. The theory behind this mechanism will be further addressed in section 6.6. The hypothesis is that when young children are just starting to learn words, they assume that the meanings of words are mutually exclusive; that is, that each word has a completely distinct meaning.[1] Although most of the Psycholinguistic work in this area has been in the application of this constraint to noun/object learning, Camille was extended to incorporate Mutual Exclusivity to examine its efficacy for noun *and* verb learning.

The basic implementation of this constraint was fairly simple. Camille was extended to keep track of the words that refer to each concept. During the acquisition process, Camille 2.0 would not consider concepts that were already the referent of another word. Mutual Exclusivity was enabled by a switch to facilitate evaluation of the system with and without this additional feature.[2]

One implementation decision had to be made that was not addressed in the Psycholinguistic literature. The question was what to do with hypotheses that contained multiple possible referents (as Camille's hypotheses often do). The work in children's language acquisition assumes that a child maintains only one referent for each sense of a word. Thus when another word is encountered that appears to refer to the same concept, Mutual Exclusivity can be applied, resulting in the rejection of that hypothesis and a continued search for the referent of the unknown word.

The approach taken in the Camille implementation was to store the mapping from a concept to the word *only* if the system was "sure of itself", that is, if Camille inferred only one possible concept as a referent for the unknown word. Thus, if Camille had produced multiple concepts for the meaning of some word, this would not be used as evidence against assigning another word to one of those concepts.

The results of testing Camille with the Mutual Exclusivity constraint active were as follows: for the Assembly Line, Recall was 69%, Precision was 23%, Production was 94%, Accuracy was 73%, and Parsimony was 14%. For the Terrorism domain, Camille achieved 24% Recall, 21% Precision, 88% Production, 27% Accuracy, and 12% Parsimony. For the Assembly Line test, the results (both overall and word-by-word) were quite close to those produced by Camille 1.2. The Mutual Exclusivity system changed only two hypotheses. In one case, the concept Uncoil was rejected as the meaning of "uncoil" because the word "route", which can also take a Wiring-Harness as an object, was attached to the concept first. ("Route" later

[1] This hypothesis is clearly not true in general. It would imply, for example, that "dog" and "collie" could not refer to the same object. Current Psycholinguistic theory suggests two resolutions: First, there are certain conditions under which children will suspend the Mutual Exclusivity constraint. Second, this mechanism is used only for a brief period during early linguistic development, and then discarded. Note that this is a clear contrast to Camille's situation, especially in the Terrorism domain, in which a large number of lexical items have already been defined.

[2] This is a simplification of the Mutual Exclusivity constraint described in the Psycholinguistic literature. As previously mentioned, the Psycholinguistic work has concentrated on object labels. The basic version of the constraint has as many implications for concept creation as for object labeling (related Camille work is discussed in section 4.4). For example, in the "dog" / "collie" example given above, Mutual Exclusivity would suggest the creation of two (exclusive) concepts at the same level of the object hierarchy. Another formulation of the Mutual Exclusivity constraint, however, specifies that it applies only within a level of the knowledge representation hiararchy. Thus it would distinguish "collie" and "poodle", but not "collie" and "dog".

appeared in an example which caused the system to change its hypothesis for the meaning of this word.)

In the Terrorism domain, the results were lower than for Camille 1.2. Because of the range of expression in the texts and the coarse granularity of the required output, there were many words that referred to the same concepts. There were 10 different words that mapped to the concept Attack. In general usage, these words ("club" and "torture", for example) mean different things, but for the purposes of the information extraction task, they were synonymous. For the words meaning Attack then, there was only a one in ten chance that the appropriate concept had not already been "claimed" (assuming Camille inferred only one concept per word). Thus, a high number of correct hypotheses were rejected because a synonym had already been defined.

This led to a general conclusion about this mechanism. Mutual Exclusivity seems to work well for the early stages of learning, when the agent is learning a lot of new words. Later, more of the words tend to overlap in meaning, and therefore, Mutual Exclusivity may steer the learner away from a reasonable hypothesis. This is consonant with accounts of the use of Mutual Exclusivity in children [Markman, 1991; Markman, 1990], which suggest that children use this constraint for only a brief period in their development. Clark [1987] maintains that language users always follow her more general Principle of Contrast, which states that no two words are exact synonyms. This does not conflict with the conclusion about Mutual Exclusivity described above because LINK only represents gross features of word meaning. Thus a maturing language learner can allow two words to be synonyms on the level of representation that Camille uses, but allow that they are distinct at some more subtle level (for example, Formality, as in "cop" versus "policeman"). Because Camille does not use such a fine-grained knowledge representation, its lower scores on the Terrorism domain tests were not surprising.

The implementation of this constraint was interesting in that it seemed to raise more questions than it answered (leading to speculation about the use of cognitive models which will be taken up again in Chapter 6). The first problem was that of the multiple-concept hypotheses as mentioned at the beginning of the section. It is unclear what humans do. Do they *never* have multiple possibilities for the meanings of words? Do they have a threshold of activation over which they assume that they have found the proper meaning?

The second difficulty could be called the Truth Maintenance problem. What should the learner do if some mapping of a word to a concept rejects a later hypothesis, and then the original mapping changes? Consider the case of the word "rob" as described at the beginning of this section. Under the mutual exclusivity constraint, if the learner heard the sentence "Harry torched the building", Arson would be ruled out as a possible meaning because the word "rob" is already attached to it. But what if the system later corrected its hypothesis of the meaning of "rob"? Is some sort of truth maintenance system necessary to track these mappings and their dependencies?

The last question is, under what circumstances is the Mutual Exclusivity constraint overridden? It is clear that people eventually learn that some words are very close synonyms, and the meanings of some words overlap with each other. What type of evidence must the learner receive to override this constraint? The discussion of the Psycholinguistic theory behind Mutual Exclusivity and the analysis of its use in a cognitive model are carried on in Chapter 6.

4.2 Camille 2.1: Scripts

The enhancements to Camille that were described in the previous chapter resulted in a fairly powerful meaning-inference mechanism. Despite the promising results, however, Camille was still incapable of making certain distinctions. Too many different concepts were indistinguishable on the basis of the intra-sentential knowledge that the system utilized to make its hypotheses. As a result, many of the hypotheses contained a large number of concepts. For example, Camille 1.2 inferred the set of hypotheses (Ambush Injure Kidnapping Murder Shoot) for the word "kidnapped". Despite the fact that the system encountered several examples of the word's use, in each, the ACTOR was a Terrorist and the OBJECT was a Civilian. Using only knowledge about role-filler constraints on actions, Camille was unable to distinguish among the members of the set of hypotheses. Additional discourse information, for example that there was a ransom demanded, can steer Camille to the correct single hypothesis.

In order to extend the system's knowledge — and thereby extend the inferences that it could make about unknown words — knowledge about sequences of actions was added to the semantics in the form of scripts [Schank and Abelson, 1977; Cullingford, 1977]. Scripts specify common sequences of events or scenes. The classic example of a script describes what happens in a restaurant: the patron enters, is seated, gets a menu, orders, eats, pays, and leaves.

Cullingford's Script Applier Mechanism, SAM [1977], processed stories using scripts. Each script had certain trigger words defined for it, many of which were verbs. For example, trigger words for the restaurant script might be "went out", "dined", "restaurant", or "ordered". After a script was selected, SAM invoked a Conceptual Dependency [Schank, 1973] analyzer on each sentence and matched the constituents of the sentence with the expectations of one of the scenes in the script. The integration process filled in the slots of the script and created pointers between the sentences in a form of anaphora resolution. After all the sentences of a story were processed, the script contained a cohesive account of the story. Because scripts have expectations about the scenes and specific slot fillers in a scene (for example, the restaurant customer pays the bill at the register), the script could use these expectations as defaults for information which was left out of the story.

Scripts can be put to a different use in the word-learning process: they constrain which actions (and thus which word referents) are likely to co-occur. This provides a sort of discourse-level knowledge that was missing in previous versions of the system. Thus the context of surrounding sentences can be leveraged in the lexical acquisition task. This section describes the implementation of Camille's script applier, its relationship to lexical acquisition, and results of empirical testing.

4.2.1 The script structure

Camille's scripts are defined in the semantic knowledge base in a similar manner to the basic domain knowledge. Figure 4.1 shows an example of a script from the terrorism domain. This script defines a sequence of actions likely to be associated with a bombing.[3] Line 1 gives the name of the script and its parent in the semantic hierarchy. The numbered arcs in lines 2–4 specify the subevents that are expected, similar to the specifications in grammar

[3] Note that this type of script differs somewhat from the standard type mentioned above. Instead of describing a sequence of actions that occur in a story, Camille's terrorism scripts describe actions that are likely to be reported in newswire accounts. Thus, there is less of a connection between the scenes of the scripts.

```
(define-sem bomb-script is-a (terrorist-action-script)      ; 1
   formulae (((1) = purchase                                ; 2
              (2) = bombing                                 ; 3
              (3) = destroy                                 ; 4
              (1 actor) = (2 actor)                         ; 5
              (1 object) = explosive                        ; 6
              (1 object) = (2 instrument)                   ; 7
              (2 object) = (3 object))))                    ; 8
```

Figure 4.1: A simple terrorist script

rules of which constituents make up the particular phrase.[4] Lines 5–8 specify constraints on the scripts. 5, 7, and 8 provide links between slot fillers of the different events. Line 6 specifies the type of a slot filler for a particular subevent. As with regular concept definitions, the script definitions are translated into DAG form so that they can be unified with parse results.

Scripts can also represent more complicated situations. Figure 4.2 shows a set of interrelated scripts. These scripts illustrate two additional mechanisms for representing sequences. The first is the use of subscripts. Line 4 specifies that the third action of the assassination-script is a script itself, investigation-script. This script has one child in the is-a hierarchy, namely investigation-script-1. This representation allows a subscript to be included in various other scripts.

The second advantage of this representation is that it allows for alternatives. The definitions starting at lines 20 and 24 are both children of the trial-script node which is part of investigation-script-1. These subscripts allow for two different sequences of subevents to be included in a script. Their application to lexical acquisition will be described below.

4.2.2 Camille's script applier

The most basic action involved in applying a script to a discourse is the same as the basic action for parsing: unification. Because the scripts are represented in the same format as all the other information that LINK uses, it is possible to simply unify the semantic representation of the parse with a scene of the script. However, since the "constituents" of the scripts are not words as they are for the parser, a different mechanism is needed to invoke the unification. Note that this mechanism is independent of the representation of the scripts, and so there are many possible ways of applying the scripts to the discourse. Traditionally, key words are used to "trigger" particular scripts. Because verbs provide the linguistic head for the meaning of the sentence, the verbs usually serve as the script triggers. But it is precisely these words that Camille is trying to learn, so the system can't use this type of script application. The mechanism described here is appropriate to the lexical acquisition task.

The basic script applier started with a list of possible scripts that could be encountered (or were defined as "interesting" by the MUC-type tasks). The script applier looped through

[4] Although the ordering of the subevents in the scripts is the standard ordering that occurs in the accomplishment of the action, the events in newswire corpora such as that used in the Terrorism domain are often reported in a different order. For this reason, different versions of the application mechanism were tested, one which required strict sequentiality and one which accepted the events in any order. The results of the evaluation are described in sectionřefscript-eval.

```
(define-sem assassination-script is-a (terrorist-action-script)    ; 1
   formulae (((1) = plan                                           ; 2
              (2) = murder                                         ; 3
              (3) = investigation-script                           ; 4
              (4) = identification                                 ; 5
              (2 object) = human-or-official                       ; 6
              (1 actor) = (2 actor)                                ; 7
              (3 3 object) = (2 actor)                             ; 9
              (4 object) = (2 actor))))                            ; 10

(define-sem investigation-script-1 is-a (investigation-script)     ; 11
   formulae (((1) = investigation                                  ; 12
              (2) = questioning                                    ; 13
              (3) = charging                                       ; 14
              (4) = trial-script                                   ; 15
              (1 actor) = law-enforcement                          ; 16
              (2 actor) = (1 actor)                                ; 17
              (3 actor) = government-official                      ; 18
              (4 object) = (3 object))))                           ; 19

(define-sem acquittal-script is-a (trial-script)                   ; 20
   formulae (((1) = acquittal                                      ; 21
              (2) = release                                        ; 22
              (1 object) = (2 object))))                           ; 23

(define-sem conviction-script is-a (trial-script)                  ; 24
   formulae (((1) = conviction                                     ; 25
              (2) = sentencing                                     ; 26
              (1 object) = (2 object))))                           ; 27
```

Figure 4.2: A complex terrorist script

the sentences in a text and ran the parser, including Camille's lexical inference procedure, on each one. After each parse, the script applier attempted to unify the semantic representation of the parse with one of the numbered scenes of any of the scripts. As a side effect of successful unification, the semantics of that parse were included into the script. If the unification was unsuccessful, nothing was added to the script. This process continued for each sentence, attempting to unify its semantic representation with some scene of each script.

If the scene was filled by the parent of some subscript, additional processing was necessary. Since there could be multiple incompatible subscripts, a copy of the entire script DAG was made for each subscript. Then the unification process was attempted recursively on these subscripts.

When all of the sentences were parsed and their parses merged into the scripts, the script applier filtered the set of scripts, keeping only the script with the highest percentage of its scenes filled. This remaining script was then the representation for the entire discourse.

An example will help illustrate the operation of the mechanism. Consider the short text:

```
Terrorists bombed the Parliament building today with high explosives.
The attack destroyed the east wing of the building.
```

After parsing the first sentence, the script applier attempted to unify one of the scenes of the various terrorist action scripts with the semantic representation of the parse. Because the head of the parse was Bombing, the only script with an acceptable scene was the bomb-script. Unification fit the action into the second scene of the script and created links to the first and third scenes. After parsing the second sentence, the Destroy action was fit into the third scene of the script. Note that the finished script supplied the default information that the explosives were purchased somehow by the offending terrorists.

4.2.3 Scripts and lexical acquisition

The introduction of scripts into the parsing process is potentially quite beneficial for the lexical acquisition task. Given a sentence in isolation like, "The terrorists froobled the senator," Camille could not distinguish between the various possible interpretations of "froobled". If, however, the same message contained information about the senator being held hostage, or the terrorists demanding random, then Camille would be able to instantiate the kidnapping-script and uniquely infer the meaning of "frooble".

The method for doing this is basically the same as was described above. The key though, is that if the main verb of the sentence is unknown, then Camille's lexical inference mechanism will be invoked and propose a set of hypotheses for the word's meaning. Unification handles this event by eliminating those concepts from the set which do not unify. For example, unifying Murder with (Murder Kidnapping Injure) returns (Murder). Because the scripts tend to mention very specific concepts like Murder, the application of the scripts to the text can quickly reduce the number of extraneous hypotheses.

A simple example will explain how this could work. Consider this slight alteration of the previous example message (assuming the system does not know either verb):

```
Terrorists nuked the Parliament building today with high explosives.
The attack obliterated the east wing of the building.
```

While processing the first sentence, Camille initially infers that "nuked" could mean one of several different concepts, (Arson Bombing Destroy). Because only Arson has a constraint that the OBJECT is a Building, the other concepts are eliminated from consideration. When the script applier attempts to unify the parse result with the scripts, several are eliminated because they deal with objects that are human. When attempting to unify the parse result with the bomb-script, the script applier finds that the slot fillers of the sentence match those in the script, but the concept doesn't match. Because the meaning is just hypothesized, the script applier starts searching for a concept that is legal for the particular slot fillers and fits in the script. Eventually it finds the Bombing concept and enters it into the script. After processing the next sentence (again Camille produces Arson as the possible meaning of "obliterate"), the script applier finds that by generalizing, it can find the concept Destroy which fits into the script. After processing the entire message, the resulting script is searched for word hypotheses, and they are written to the lexicon.

In order to add the use of scripts to the Assembly Line domain (described in Appendix B), a different approach to implementing the script applier was needed. Because the "messages" in this domain describe sets of actions that an assembly line operator must perform, the sentences are inherently sequential. Furthermore, adjacent sentences often refer to the same slot fillers, for example:

```
Get door handle from bench.
Position handle.
Secure handle with two nuts.
```

In order to fit this type of sequence into the Assembly Line scripts, another version of the script applier was created that required sequentiality of the actions. In order to allow for botched parses and unexpected operations, this script applier allowed some of the scenes to be skipped. The script applier was also extended to allow iterations of actions, for example allowing a Get followed by any number of Manipulations followed by an Assemble.

4.2.4 Empirical testing

The use of scripts promised to be a powerful mechanism for adding discourse information to LINK and to Camille. The actual test results were somewhat equivocal, however. In the Terrorism domain, Camille 2.1 scored Recall 30%, Precision 43%, Production 60%, Accuracy 50%, and Parsimony 30%.[5]

The biggest success of the script mechanism was in increasing Precision. Camille produced a total of only 7 concepts in its 6 hypotheses that it generated for the Terrorism verbs. As hoped, the scripts succeeded in eliminating many of the extra concepts which could not be ruled out based on intra-sentential information.

In some ways, however, the script mechanism did not meet expectations. In the Terrorism domain, the nature of the texts was quite different from the intuitive notion of script structure. For example, in 100 messages, 9 mentioned kidnappings. Although the use of a script to represent kidnapping seems appropriate (the victim is taken hostage, a ransom is demanded and paid, and the victim is released), the articles as a rule did not refer to related

[5]These results are not directly comparable to the tests run on the basic system. The texts were complete newswire messages instead of selected sentences. The communication verbs ("reported" for example) that were learned by the basic Camille system did not appear in the scripts, so they were not counted in the calculations. Nevertheless, a graph that combines all the test results is included in section 4.5.

events. None of the 9 articles mentioned any ransom demands. One mentioned an escape from an attempted kidnapping. Only one used the word "hostage" and that was as a noun referring to the victims.

What did the messages include? All of the messages that mentioned the kidnapping of a single person went into detail about who the person was and sometimes where he or she was going when kidnapped.[6] This is exactly the same type of information provided about the victims of *any* terrorist attack. Thus, none of the concepts that one normally considers as being part of a "standard" kidnapping sequence helped Camille learn meanings of unknown words. In the end, a more general script form was used that allowed greater variation, for example: Nasty-Action, Wound, Murder. This type of script outperformed the more specific scripts defined above.

In the Assembly Line domain, the "stories" described repetitive sequences that the operators would perform at their station. These also appeared, at first glance, to be well-suited for representation with scripts. The sequences often could be described as:

> Get a part. Maybe get a tool. Prepare the part in some way. Attach the part to the car. Start over.

Unfortunately, the uniformity of this sequence was not maintained. For example, a particular operator could Position a door handle that another operator several stations down the line would Secure. Several different scripts were applied to the example texts. As in the Terrorism domain, the scripts for which lexical acquisition worked best were the most general: "Do any number of Factory-Actions. Do any number of Finishing-Actions." As described in Appendix B, the scores for the test were: 34% Recall, 18% Precision, 40% Accuracy, 86% Production, and 6% Parsimony.

Although the use of scripts did help the lexical inference process, the mechanism was not as useful as had been reported by other authors. As will be described in Chapter 5, scripts were the primary knowledge source for lexical acquisition for two of the more prominent earlier systems, Foul-Up and Rina [Granger, 1977; Zernik, 1987b]. Unfortunately, neither Granger's work nor Zernik's was systematically applied to real-world texts. It is no surprise that these systems performed well when the authors wrote the scripts *and* the texts that their systems processed. Considering the results found here, there is little reason to believe that their approaches would be as effective in more realistic circumstances.

Although scripts do serve to bring in more discourse level knowledge, they are limited in the type of knowledge that they provide. Scripts only describe likely sequences of actions and possible links between the slot fillers. They do not allow inferences based on goals or plans of the agents involved. A more robust concept representation mechanism could encode, for example, that the result of Getting an item is to have that item, and perhaps that the goal is to use that item. Then the system would not have to rely on sequences like: Get Tool, Use Tool, Discard Tool. It could use the fact that is has some Tool to predict that it will be used for some later (but not necessarily immediate) action.

As suggested by the analysis of the kidnapping texts, there was one other action that was associated with a kidnapping: Escape. This was quite rare, however, and was often separated in time from the original event. In the other message where the word "hostage" was

[6] This is not really surprising for several reasons: many of the other actions would not be known at the time of the report; the other actions might not be considered important enough to report; and unless the victim is world-famous, the victims "importance" must be described.

used, it was the ACTOR of the sentence. Thus a script mechanism which links series of actions would not be able to make the connection. Hypothetically speaking, if the system's goal were just to learn the meaning of the word "kidnap", a faster (although still rarely applicable) mechanism would be to perform a keyword search through the message looking for related words (similar to some Information Retrieval stuff). If certain trigger words were found, the various concepts in the hypothesis could be disambiguated. Although such a mechanism could work for particular cases, it would not be generally applicable. Very few of the other Terrorist actions have such "partner" actions.

4.3 Camille 2.2: Learning ambiguous words

Lexical ambiguity has been a thorn in the side of NLP for a long time (for an overview of the difficulties caused by ambiguous words, see [Lytinen, 1988]). Much research has been devoted exclusively to this problem. The goal of most of this work, however, was to devise a mechanism for choosing between word senses. In other words, if a system knew that it had encountered a word for which it had multiple definitions, how could it determine which sense of the word was appropriate?

Fortunately, LINK handles this problem quite elegantly. An appropriate example was cited in Lebowitz' thesis [1980, p. 229] which presented a system which deliberately made generalizations based on natural language input:

`Terrorists sprayed a car ... with automatic weapons fire ...`

Lebowitz' system used several different heuristics to disambiguate between the different possible meanings of spray. In this case, the system had a heuristic that if the domain was terrorism, then the appropriate sense must be Shoot.

LINK's word definitions, with their constraints on the slot fillers that can be attached, handle this situation in a different way. LINK's lexicon has multiple definitions for the various senses of spray. They differ in their semantic interpretation *and* in the slot fillers that they take. For example, spray could be defined (in a simplified way) like this:

```
(Define-Word Spray
   (Verb (Head Rep) = Squirt-Liquid
         (Head Rep Instrument) = Hose)
   (Verb (Head Rep) = Shoot
         (Head Rep Instrument) = Gunfire))
```

When LINK encounters the word "spray", both definitions are entered into the chart. During the parsing process when the phrase "with automatic weapons fire" is attached, "weapons fire" which is subsumed by Gunfire is attached as the instrument of the main verb, "spray". This action rules out the interpretation of spraying with a hose, and that branch of the parse is not continued.

While learning new words, however, a different problem arises. If Camille does not know a word and that word is ambiguous, Camille must recognize that fact and somehow differentiate between the meanings. Failure to recognize ambiguity can result in overly general hypotheses. For example the word "apply", in the Assembly Line domain, is used in two slightly different ways. In "Apply tape to manifest," the operator is being told to attach a Fastener to a Record. "Apply manifest to door" means to attach a Record to an Auto-Part, and implies the use of a Fastener. If Camille does not recognize this ambiguity, it will search for a meaning for "apply" that can take OBJECTs that are both Fasteners and Records. Because

the least upper bound of these nodes in the concept hierarchy is Factory-Object, there are few verbs which have such a general constraint. Moreover, once the system infers one of these verbs as a potential meaning, it is difficult to disconfirm. Only a sentence with an object that is *not* a Factory-Object could provide the necessary negative evidence, and these are rare because Factory-Object subsumes most of the object hierarchy. This section describes the variation of the learning mechanism that allows Camille to recognize ambiguity and generate multiple definitions.

4.3.1 Noun ambiguity

Because of the differences that were previously discussed between verbs and nouns, the mechanisms to deal with ambiguity in these words differed significantly. This section describes Camille's method for processing ambiguous nouns. The problem of handling ambiguous verbs is discussed in the following sections.

When Camille encounters a noun that is undefined, it creates a generic definition with a general object concept. When the noun is attached to a verb, that verb's constraint, as a by-product of unification, limits the interpretation of the meaning of the noun. At the end of the parse, Camille need only record that interpretation as the hypothesized meaning of the word.

Using this method, Camille is not making inductive hypotheses about the word's definition.[7] It is only applying a constraint that it knows must be true. For example, the object of Detonate must be some descendant of Bomb (barring non-literal usage which will be left for someone else's dissertation). Thus, if Camille did not know the meaning of "charge", it could infer from the sentence, "Terrorists exploded a charge under the bridge," that "charge" was a type of Bomb. If the system later processed the sentence, "Terrorists assassinated the charge d'affaires of the embassy," the system could (somewhat ignorantly) realize that it had a different meaning for the same word and hypothesize two senses of "charge".

The actual implementation of this mechanism is along the lines described above. The generic noun definition which is put into the chart for an unknown noun contains the marker UNDEF-NOUN. At the end of the parse, Camille searches for all words that are marked as undefined. Any tentative definitions are saved to the lexicon. Further examples of the word's use can either confirm the hypothesis, or provide additional evidence. If the unknown noun is used in a more restrictive setting, unification will again narrow the interpretation of the word. If the constraint on that noun is outside the subtree described by previous usage, the system finds itself trying to unify a marked definition with an incompatibly-labelled DAG. The unification procedure was extended to notice this event and to produce a disjunctive meaning for the word. For "charge", this definition would be (Bomb $\vee$ Human).

In order to test the ambiguous noun definition mechanism, the definitions of the following words were removed from the lexicon: branch, charge, lines, others, plant, post, quarter, state, and system. Because the lexicon only included definitions for the words that were likely to be found in this domain, and it only included senses which were relevant to the domain, these were the only ambiguous words in the lexicon.[8] Furthermore, many of these words were not "targets" for the domain. That is, their contribution to the sentence was not

[7] A more liberal approach will be discussed in section 4.4.

[8] Like the word "others", some additional words in the lexicon were vague. (For a discussion of dealing with vague versus ambiguous words, see [Lytinen, 1988].) "Others" was the only vague word tested because it occurred prominently in such examples as, "11 others were wounded."

important to the information extraction task. Nevertheless, after processing 100 sentences which contained this set of words, Camille 2.2 was able to create ambiguous definitions for five of the nine words: lines, others, post, state, and system.[9] The system hypothesized 5 out of 9 ambiguous definitions, for a Production score of 56%. Recall, counting the correct definitions, was 8 of 18 possible definitions, or 44%. Precision and Accuracy were 8 out of 12, or 67%. Because the system created only one concept for each sense of the ambiguous definitions, Parsimony was the same as Recall, 44%.

This mechanism also provided valuable leverage for the process of inferring additional concepts for the system. This variation of the system will be discussed in section 4.4.

4.3.2 Verb ambiguity

The ambiguous verb problem was most evident while processing the Assembly Line domain.[10] Due to the telegraphic nature of the text in this corpus, several words referred to different actions if used in different contexts. If the word "check" had some kind of Record as its OBJECT, the intended action was to examine the record, which contains the specifications of the car, to see if a certain option was included. If "check" had some other Factory-Object as its OBJECT, then the intention was to examine the object for defects.

This situation could be interpreted in two ways. It could be seen as an indictment of the concept representation. One could claim that in this example, there are not really two different senses of "check" being used, there is just one: to examine something. The only difference is that in the first case, there is something specific to look for. Because the second example of "check" does not specify something to look for, the default, Defects, is implied.[11]

The other interpretation is that there is a true ambiguity: two distinct senses of "check". If the system does not know the meaning of "check", then it will have a difficult time inferring its meaning. Because "check" will be encountered with a variety of slot fillers, it will look like a general verb. The key to recognizing that there should really be two specific definitions is for the system to realize that the slot fillers fall into two neat groups. The rest of this section describes Camille's implementation of this capability.

Because Camille only has its linguistic input and no external context to leverage against the ambiguity problem, it was necessary to implement a heuristic that splits hypotheses based on multiple examples. Two different approaches are possible. One approach would be something like this:

> **The Liberal Approach:** If a new word is encountered with a different slot filler than previously seen, split the definition into two senses. In order to reduce the number of hypotheses, join new words that have been encountered enough times and whose slot fillers do not fit into two obvious groups.

The other possibility, loosely stated, would be:

---

[9] It also created single definitions for many other words that had been overlooked in the system development. For example the word "impunity" was inferred to be an Instrument-Object.

[10] Appendix B contains a description of this domain.

[11] Note that this is a relatively rare situation. As suggested in Lebowitz' "spray" example, words with multiple senses can often be disambiguated by putting them in a larger context. The information extraction task is inherently single-context and thus most (but not all) ambiguous words have only a single meaning within the domain.

The Conservative Approach: If a new word is encountered enough times and its slot fillers break into two obvious groups, then try splitting it into two different senses.

Of course, this principle must be operationalized, and it was done (somewhat arbitrarily) as follows: "Enough times" is defined as four instances. This seemed like the least number of occurrences that would be likely to limit the probability of the instances coincidentally breaking into two groups. "Obvious groups" was defined along the lines of *basic levels* (see, for example, [Waxman *et al.*, 1991; Corter and Gluck, 1992]). The psychological literature suggests that these are sets of culture-specific concepts which people are likely to use to name things, for example, Chair as opposed to Furniture and Lounge-Chair. For this task, the basic levels are operationalized as the concepts which are most likely to be used in slot-filler constraints, for example, Tape and Record.

Although both approaches might yield the same result with a large number of examples of unknown words, the relative infrequency of ambiguous words within these domains and the likelihood that an unknown word would be encountered rarely suggest that the conservative approach is more suited to the information extraction task. Camille relies heavily on differing slot-fillers to inform it that it has chosen an overly specific hypothesis for a word's meaning. This allows Camille to settle on an appropriate hypothesis in a small number (usually two or three) of examples of a word's use. The liberal approach would conflict with this strategy and result in a highly fractured set of hypotheses for unknown words.[12]

In order to implement the conservative approach, Camille 2.2 checked a new word's definition before it was stored in the lexicon. If the word had been encountered four or more times, the Camille performed the following steps on its record of slot fillers (as described in section 3.4):

- Order the principal slot fillers by the importance of the slot (defined, from greatest to least as: OBJECT, ACTOR, INSTRUMENT, DESTINATION, PURPOSE.
- Check each slot to see if its fillers fall into exactly two basic categories.
  - If so, break the slot fillers into two separate groups and attempt to find more specific concepts that accept the different slot-filler groups.
    - If that works, store the two different definitions.
    - If not, store the original definition.
  - If not, store the original definition.

This mechanism was run on the same set of 100 example sentences as was the original test of the Assembly Line domain. Unfortunately, because of the small number of repetitions of the words in the test set (median value 2), none of the ambiguous words in the corpus were encountered enough times for the mechanism to be triggered. Of the three words that were classified in the initial results as being ambiguous, two of them occurred only twice in the test set and the third occurred three times. In this limited test, none of the ambiguous words occurred often enough to allow Camille to make an appropriate decision on whether or not it was ambiguous.

[12] There might be some domain, however, in which ambiguous words are prevalent. In this case the liberal approach might be indicated.

Camille did split one definition, however, for the word "open". The test set contains 5 uses of "open", and in 4 of them, the object is a Box and in the other, a Door. Because Box is a child of Container and Door is a child of Chassis-Component, the objects break into two obvious groups, so Camille splits the definition. However, because Door and Container have a common ancestor, Door-or-Container, and because this node is the OBJECT constraint for Open, both split concepts end up inferring that Open is still the most reasonable hypothesis for the two senses of "open".

This unexpected state of affairs led to the discovery of a different method for handling ambiguity. Instead of splitting the word's definition into two different senses, an additional concept could be created (as discussed in section 4.4) to make a disjunctive constraint on the action concept. In the "apply" example, a concept, Tape-or-Record could be made a parent of the two objects, and then could serve as the constraint for the concept Apply. Intuitively, it seems that each solution would be appropriate in different circumstances. Thus, in the "apply" and "open" examples, the different senses seem similar enough that it would be more parsimonious to represent their meanings with only one node. In the "spray" example from Lebowitz' thesis, however, the two senses are quite different, and should be split.

4.3.3 Limitations

Camille cannot tease apart ambiguity if it already has some other definition for a word. The system assumes that the definitions it was given are undisputable facts. The only words it will learn are those that do not exist in any form in its lexicon. So if, for example, the system already knew the Squirt-Liquid sense of "spray", and found it in the example sentence above, it would simply fail to parse the sentence because gunfire would be an unacceptable INSTRUMENT. The system could only learn the different senses of "spray" if it knew *neither* of them beforehand.

The most straightforward way of dealing with this limitation would require that LINK treat all of its word definitions as tentative and to have a mechanism for overriding them. An explanation-based method for accomplishing this would be for the system to realize that it *could have* parsed the sentence, if the semantic component of the definition for "spray" had been different.

Another method would be to have some sort of preference system to "score" parsing decisions instead of making them as all-or-nothing decisions. Thus, the system could allow the combination of Gunfire as the INSTRUMENT of "spray", but give it a very low score. When no other parse was available, it could use the fact that its only parse had a sub-threshold score to signal the learning mechanism. Intuitively, this coincides with psychological theories that suggest that learning is more likely to occur in humans when their expectations are not met.[Kaplan *et al.*, 1990]

4.4 Camille 2.3: Expanding the domain knowledge

The basic assumption of complete *a priori* concept knowledge is too restrictive. A lexical acquisition system should be able to learn concepts as well as word meanings. In humans, the concept acquisition task must rely to a large part on the various modes of perceptual input, visual, aural, even tactile. Perception becomes codified as concept knowledge. Psycholinguistic evidence suggests that there is a linguistic role as well. A study by Carey and Bartlett [1978] which will be further described in section 6.1 showed how children might use the presence of

an unknown word as a signal that they should learn a concept.[13]

Unfortunately for Camille, it has no input mode other than its example sentences. Learning concepts without these other modes is difficult, but the Mutual Exclusivity constraint provides one technique. Section 4.1 described the use of this constraint for influencing the inference of verb meaning. If a concept already had a word referring to to it, that concept was prohibited from becoming the referent of any other word. The constraint could be turned around, however, to signal a gap in concept knowledge. If the system is fairly sure that it has made a correct meaning hypothesis for a word and that hypothesis is rejected by Mutual Exclusivity, Camille can infer that a new concept should be formed. This technique will only work for object labels for two reasons. First, Camille can only be "fairly sure" of a hypothesis for a noun's meaning. As discussed in section 3.1, the constraints on action concepts place an upper bound on the meaning of an unknown slot filler. The referent concept for the noun must be a descendant of the target node named by the constraint. Second, a new concept created to be the referent of a noun can be located within the subtree underneath the constraint's target node. For a verb, it would have to be above the target node, but Camille would not know where exactly to locate it.

As in some of the other situations that Camille was confronted with, a conservative approach and a liberal approach were both available. They can be described as follows:

> **The Conservative Approach:** When creating a new node, place it immediately under the node specified by the constraint.
> **The Liberal Approach:** When creating a new node, search under the node specified by the constraint. If a node without a label is found, attach the word to that concept. If not, create a new node as described above.

When learning verb meanings, Camille must take a liberal approach, favoring the most specific hypotheses, in order to get usable, falsifiable hypotheses. For recognizing ambiguous verbs, Camille used the conservative approach. In order to maintain its ability to learn ambiguous nouns, Camille uses the conservative approach here too. If the system used the liberal approach, and later encountered a conflicting use of the noun, Camille would not know if it had found an ambiguous word, or if it had made a wrong initial guess about the referent of the word. As discussed in section 3.1, taking this conservative approach to learning noun meanings does not significantly reduce the usability of the hypotheses.

The implementation of this technique required only a simple extension to the noun-learning mechanism described in the previous section. As every word was defined (either from the lexicon or from learning by Camille), Camille 2.3 kept track of the word or words that referred to each concept (in the usual Mutual Exclusivity way). When the system inferred a meaning for a new word, it checked to see if the concept to which it referred had already been the referent of another word. If there was a conflict *and* the new word was inferred to be a noun (based on its morphology and position in the parse), then a new concept was created under the original node. The unknown word was then mapped to this node.[14]

---

[13] Carey and Bartlett actually implied that the concept already existed and that the children mapped the new word to it. The example that they used, however, was an olive color. Because there is a continuous range of colors, it would be impossible to have a concept for each one, so it is likely that the children created the appropriate concept "on the fly".

[14] Ideally, the system would later attach additional information to this node to differentiate it from others. Possible methods for augmenting a concept will be discussed below.

Camille processed the sentence, "As a result of these attacks, several persons were wounded and others died," without a definition for "others". Because the constraint on the ACTOR of Die is Human, the word "others" was inferred to refer to the concept Human. Of course, there were many other referents to this concept that had already been defined. Camille 2.3 then created a new node (labelled Other30078 for lack of a better name), made it a descendant of Human, and made it the referent of "others".

Camille 2.3 was evaluated on the same test as the Camille 2.2 version for recognizing ambiguous nouns. 100 sentences were processed with 9 target words. The Production score was the same as for the ambiguous nouns test: 5 out of 9, or 56%. Because there were also 8 correct definitions, the Recall, Precision, Accuracy, and Parsimony scores were also the same as for the ambiguity test: 44%, 67%, 67%, and 44%. Of the 12 definitions that Camille created, only two did not result in the creation of a new concept. These two concepts, Place and Human-or-Place were so general that there were no other words in the lexicon that referred to them. As previously mentioned, the Terrorism domain contains a large number of words that are defined as synonyms for the information extraction task. Because of this, all of the words except these two general ones resulted in the creation of new nodes in the concept hierarchy.

Although this mechanism allows Camille to posit the existence of new concepts, the system has little distinguishing information about the new nodes. The system only has a name for the node (generated from the referring word), its approximate position in the hierarchy, and a word that refers to it.[15] Chapter 7 includes one suggestion for addressing this limitation by integrating Camille with a vision processing system.

4.5 Analysis of the evolution

This section contains an analysis of the variations on Camille's basic implementation. The major extensions to the initial Camille system were:

- Mutual exclusivity: This technique was taken from the psycholinguistic literature and applied to Camille's lexical acquisition task. When the system inferred a meaning that already was assigned to another word, it rejected that hypothesis and continued its search.
- Scripts: Often, there was not enough information to distinguish concepts based on the evidence provided by one sentence. By encoding sequences of actions in the same format as general domain knowledge, Camille 2.1 was able to increase its inference power beyond sentence boundaries.
- Ambiguous words: Camille 2.2 analyzed its record of slot fillers for a word to infer when the definition should be split into two separate definitions. This enabled it to avoid overly general hypotheses for a word's meaning. It also included a mechanism for inferring ambiguous noun meanings.
- Adding concepts: By applying the Mutual Exclusivity constraint, the system determined when its concept knowledge was incomplete. Additional nodes were added to the hierarchy based on the occurrence of unknown words.

A compilation of the test results is shown in figure 4.3. The graph shows the initial and improved basic systems, and the Mutual Exclusivity, Script, Noun Ambiguity, and Node

[15] In fact, this is the only explicit information that Camille has for any object node. It is assumed, however, that the underlying task has additional information about the domain-specific objects.

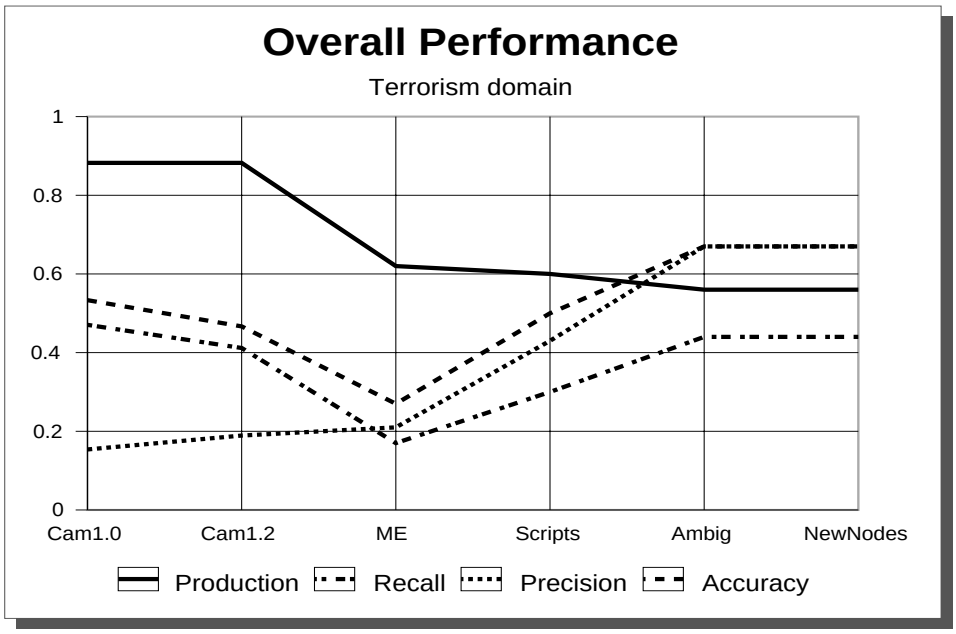


Figure 4.3: Overall Camille performance

Creation scores. As previously mentioned, the results are not directly comparable, but this graph does show some of the basic strengths and weaknesses of the system.

The major benefit of the implementation of the Mutual Exclusivity mechanism was to provide a formal setting in which aspects of the theory can be tested. One such aspect was the application of the constraint to verb learning in addition to noun learning. Although the results of empirical testing on Camille 2.0 were somewhat equivocal in regards to its broad application to verb learning, some interesting issues were raised. In accord with psycholinguistic theories, the application of Mutual Exclusivity appeared best suited for the early stages of word learning. This constraint can be applied to define a mapping quickly between a set of unknown words and their referent concepts. In future research, this ability could be better tested by simulating Mutual Exclusivity experiments that have been applied to children.

The implementation of Mutual Exclusivity also brought up fundamental questions that are not addressed in the psycholinguistic literature: how positive of a word's definition must an agent be in order to rule out the connection of the referent concept to other words? What happens to rejected mappings when the basis for their rejection is changed (i.e. the system revises a previous definition)?

The use of Mutual Exclusivity also allows Camille to infer when its concept knowledge is incomplete. As demonstrated in section 4.4, Camille 2.3 was able to use this constraint to expand its concept representation for objects.

The most basic benefit of the script mechanism was as a well-integrated representation for discourse information. Because Camille used LINK's DAG structure to represent scripts as well, information from different sentences could be combined together via unification.

The use of scripts for verb acquisition was a tricky matter. The usual method of using verbs to trigger script application was clearly unsuitable for this task. Testing showed that the best approach was to leave some of the responsibility for learning a verb's meaning to Camille's graph search mechanism. The script mechanism proved quite valuable for further restricting the hypothesis set.

This also suggested the need for tighter integration of the script processing with the actual parse process. LINK already integrates the application of syntactic, semantic, and pragmatic constraints during parsing. This allows it to rule out spurious parses as soon as possible. If it could also apply discourse-level constraints during parsing, lexical acquisition using script information could be seamlessly integrated. This would require some extension of the LINK parser, but could be very valuable. It would allow the system to make inferences based on the connections between sentences as well as the connections within the sentences.

The implementation of the script mechanism also led to an observation about the type of text that was tested. Although it seemed like the news report format should fit easily within the structure of scripts, that was often not the case. Most of the texts were more concerned with describing the entities involved in an incident than with the actions that occurred.

Camille 2.2 allowed the acquisition of the meanings of ambiguous nouns and verbs. For nouns, the system took a conservative approach to learning, inferring meanings at the general level of the constraints of the verb to which the nouns attach.[16] When an additional example of a word's use conflicted with the previous definition, an ambiguous definition was inferred.

For learning verbs, Camille 2.2 relied on the word's pattern of usage to indicate if it was ambiguous. If the slot fillers could be separated into two basic-level groups, then the definition was split. This approach is limited because it requires a certain minimal number of instances of the word's use before it can reach any conclusions about the word. It would be interesting to compare this behavior to psychological findings about children learning ambiguous words.

Camille 2.3 just scratched the surface of the concept acquisition problem. It did demonstrate one method for recognizing a deficit in the concept representation by applying the Mutual Exclusivity constraint to an unknown word. If a word does not have an appropriate referent within the current structure, then a node should be added.

There is an alternate approach possible to concept acquisition. Instead of taking the conservative approach from Camille 2.2's mechanism for recognizing ambiguity in nouns, the system could take a more liberal approach. If a verb constrains one of its slot fillers to a node that already has a referent, the system could search that node's descendants for an unlabelled concept. Thus instead of adding a node to the semantic hierarchy, the system could search for an unlabelled node. This approach would be consonant with psycholinguistic research that shows that children use unknown words to fill gaps in their lexicon [Carey and Bartlett, 1978].[17] It would conflict, however, with the conservative approach required by Camille's ambiguity recognition mechanism. If the system encountered an example of a word's use that was inconsistent with a previous example, it could not know if it was encountering an ambiguous word or if it had been driven by Mutual Exclusivity to erroneously search for an overly-specific hypothesis.

The weakness of the concept acquisition mechanism stems from the limitations of its knowledge about the world. The only knowledge source that Camille has is its linguistic input. Some of the systems that will be described in the next chapter utilize a coded form of "visual

[16] This approach is conservative in the sense that it does not try to find the most specific (and therefore most information-full) referents. The approach is not conservative in the sense that, like Camille's approach to verb learning, it chooses the most falsifiable hypotheses.

[17] There is still uncertainty, however, whether the gap is due to a pre-existing concept in the child's knowledge representation which has no lexical connection, or whether the appearance of a novel label signifies that a concept should be formed.

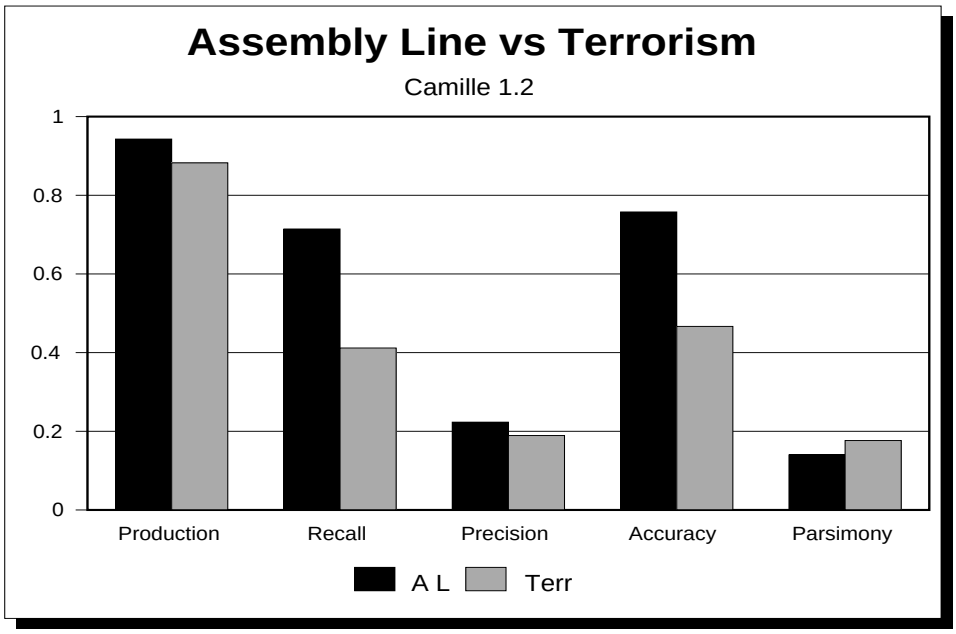


Figure 4.4: Comparing Camille performance, Assembly Line vs Terrorism

input", that is, a propositional description of a scene. By receiving "spoon-fed" external input, however, these systems might just be learning what their programmers want them to learn. Future research will examine the viability of combining Camille's acquisition mechanism with a real-world visual input system.

4.6 Cross-domain analysis

This section describes the differences that were observed as a result of the empirical testing of Camille's basic system its variations.

As shown in figure 4.4, there was an obvious inter-domain difference in the overall efficiency of the basic word-learning mechanism. This was due to two major factors: First, the complexity of the sentences was much higher in the Terrorism domain. The average sentence length is sufficient to give a rough measure of the sentence complexity. The average length of the test sentences in the Assembly Line domain was 4.3. In the Terrorism domain, it was over 23. The higher complexity of the Terrorism sentences meant that the parser seldom came up with a complete parse for the entire sentence. Although parse fragments were extracted after the parser gave up, there was no guarantee that the constituents in the fragments were consistent with the unparsed portions of the sentence. Furthermore, an incomplete parse could not be checked for required constituents.

Second, the nature of the text was fundamentally different. In the Assembly Line domain, a small set of engineers was describing (for themselves) a finite (although large) set of operations. In order to do this, they naturally developed a technical jargon. Thus a word like "aside", normally an adverb, was used as a verb, as in, "Aside packaging to trash" (get rid of the packaging by putting it in the trash). The Terrorism texts were written by a large number of journalists. Far from constraining themselves to a precise sublanguage, these writers use quite varied and complex forms of expression (more on this below).

In the Assembly Line Mutual Exclusivity test, Camille performed at a similar level

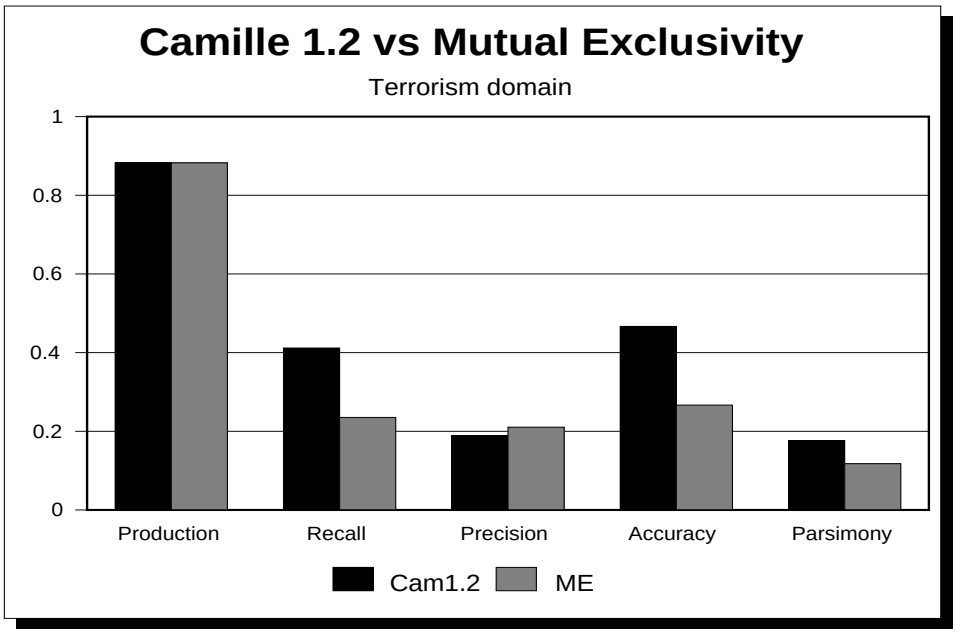


Figure 4.5: Comparing Camille performance, Basic system vs Mutual Exclusivity

as in the basic test. As shown in figure 4.5 however, all of the terrorism scores for Mutual Exclusivity were significantly lower than for the basic test with the exception of Precision which increased slightly.

The reason for this large disparity was mentioned above. The texts were written, for the most part, by professional writers, who are charged with entertaining (in some sense) their readers. In order to do so they vary the language that they use. Viewed from the vantage point of the coarse-grained domain representation for the information extraction task, the language appears to contain myriad ways for expressing the same concepts.[18] This is precisely the opposite of the language phenomenon that Mutual Exclusivity was intended to capture. With so many words defined as synonyms for the purposes of the information extraction task (9 synonyms for "attack"), it is no surprise that Mutual Exclusivity pushes Camille into making many incorrect hypotheses. It is notable, however, that Camille 2.0 also dramatically decreased the number of concepts produced in its hypotheses (from 37 to 19 for the same number of hypotheses).

The implementation of the script mechanism gave some interesting insights about the nature of the domains. In the Assembly Line domain, it appeared that the repetitive nature of the process would fit naturally into a script representation. Unfortunately, the repetitions of the actions did not fit so well into the intuitive script structure. One possible method for extending the script structure to better handle this sort of text will be presented in the future work section of Chapter 7.

It was, perhaps, even more surprising that the Terrorism texts were difficult to represent using scripts. Many of the early implementations of script mechanisms were applied to texts that were quite similar in style to the ones used here. Analysis of the corpus showed, however, that these messages do not typically describe sequences of actions. They describe

[18] It is only by inspecting the language more closely, for example at the level of discourse implications, that the Principle of Contrast would be evident.

one event and the background for that event. Camille was still able to perform quite well in the script test, but it was forced to use rather loosely-connected scripts.

CHAPTER 5

RELATED COMPUTATIONAL APPROACHES

During the past decade and a half, there have been many significant implementations of lexical acquisition mechanisms. The evolution of these systems has tended to follow a more general trend in AI. Early AI systems were aimed at demonstrating that a certain intelligent behavior was possible. Early lexical acquisition systems had the same type of existential goals. They demonstrated that within a certain micro-world and with certain input, an NLP system could make a particular kind of inference about word meaning. As AI techniques became more widely accepted and more successful on small-scale problems, the time came to demonstrate that they could be applied to larger tasks. The recent ARPA-sponsored Message Understanding Conferences (MUCs) [Sundheim, 1992; Lytinen *et al.*, 1992a; Lytinen *et al.*, 1992b; Lytinen *et al.*, in press] are a good example of current efforts to show that NLP techniques can be applied to real-world tasks. These conferences have also been a very clear demonstration of a major difficulty in scaling systems up: the knowledge acquisition bottleneck. As one group said after facing a second round of the massive development effort that they had undertaken the year before, "We either had to get some new grad students or automate our system."[Lehnert, 1992] There has been an obvious trend in the evolution of NLP systems to incorporate some type of acquisition mechanism in order to reduce this most difficult aspect of porting an information extraction system to a new domain.

At the same time, other developments in lexical acquisition systems suggest a cyclic trend. As many of the basic aspects of lexical acquisition are tackled, some researchers are focusing on demonstrating the tractability of aspects of linguistic development that were previously assumed to rely on special-purpose mechanisms. Typically, these demonstrations need to mature before they can be applied to real-world tasks. An important aspect of any AI system is its appropriateness for large tasks.

The purpose of this chapter is to locate Camille within the space of lexical inference research. The chapter begins with a tabular delineation of systems along several axes. These significant implementations of lexical acquisition mechanisms are then grouped by their overall approach and described in detail. The chapter ends with an analysis of the strengths and weaknesses of the various approaches.

5.1 Cross-system comparison

Table 5.1 summarizes the positions of the relevant systems along several axes. The axes represented in this table are:

- The name of the system
- The general approach to learning

- The purpose of the system as a whole
- The knowledge representation used
- The type(s) of information learned

Table 5.1: Various attributes of Lexical Acquisition Systems

System	Overall Approach	Purpose	Knowledge Rep	What's learned?
Moran	cognitive model	lexical acquisition	propositions	verb meaning
Child	cognitive model	cognitive model	CD, pos. grammar	verb semantics, syntax
Davra	cognitive model	cognitive model	$\overline{\text{X}}$	lexical semantics, syntax
Foul-Up	script-based	story understanding	CD	lexical semantics
Rina	script-based	story understanding	CD, sem. net	verb / particle semantics
Autoslog	tool	information extraction	Concept Nodes	lexical semantics
MayTag	tool	information extraction	Concept Nodes	lexical semantics, syntax
Loom	graph search	knowledge representation	inheritance hierarchy	lexical semantics
Camille	graph search	information extraction	inheritance hierarchy	lexical semantics, syntax

Table 5.2 shows the breakdown of the system along several other axes:

- The type(s) of input to the system
- Whether or not psychological validity is claimed
- Whether or not the system is trained
- Whether or not the system learns incrementally
- Whether or not the system was applied to a real-world task

These tables emphasize the large number of possible ways of examining lexical acquisition systems. In order to simplify the comparison, the systems are separated into four groups based on their overall approach to the acquisition task: cognitive models, script-based-systems, acquisition aids, and graph-search systems. The systems are individually described in the rest of the chapter.

5.2 Cognitive models

Salveter made one of the first notable contributions to cognitive modeling of lexical acquisition in 1979 with her system Moran. She did not explicitly claim psychological validity

Table 5.2: More attributes of Lexical Acquisition Systems

System	Input	Psych plaus?	Trained?	Incr?	Real World?
Moran	snapshots, sentences	maybe	yes	yes	no
Child	scenes, sentences	yes	yes	no	no
Davra	scenes, sentences	yes	yes	no	no
Foul-Up	stories	maybe	no	no	no
Rina	stories	yes	yes	yes	no
Autoslog	sentences, trainer	no	yes	no	yes
MayTag	sentences, cases	no	yes	no	yes
Loom	semantic structure	no	no	no	yes
Camille	sentences	maybe	no	yes	yes

for her model, but it was clear from the task that it was intended to parallel children's learning. CHILD has been under development by Selfridge for many years. It is explicitly intended to give a psychologically plausible computational account of the process that children go through in learning language, and does so by following a qualitative progression of stages that children follow. Recently, Siskind developed two systems, Maimra and Davra which were both aimed at demonstrating that computational "shortcuts" suggested by psycholinguists are not required for learning. This section describes each of these systems.[1]

5.2.1 Salveter's Moran

Although it was one of the earliest systems that modeled children's lexical acquisition, Salveter's work [1979; 1980] remains unique in its approach to incremental word learning. Like the other systems presented in this section, her system, Moran, used the help of a human trainer. The trainer supplied pre-parsed sentence case frames, like:

AGENT: Mary
ACTION: move
OBJECT: book
PREP: to
INDOBJ: table

for "Mary moved the book to the table." In addition, the trainer provided "snapshots" (basically before and after descriptions) of a scene, for example:

[1] The organization of this section goes against the general organization of the chapter. In the other sections, the systems are grouped based on the underlying approach that they use to learn language. Here the systems are organized by their basic goal — to model human behavior. Although these systems use different means to achieve that goal, they can be compared best in this way.

Before	After
MARY AT LOCA	MARY AT TABLE
MARY PHYSCONT BOOK	BOOK ON TABLE
CLOCK ON WALL	CLOCK ON WALL

Moran computed a graphical description of the action which had two parts. One node or set of nodes described the Arguments to the verb which are quite similar to Camille's slot-fillers. Another set of nodes encoded the Effects of the action which were computed from the differences between the before and after snapshots. For example, with the sentence and picture descriptions given above, the system would infer the meaning of "move" depicted in figure 5.1. This specifies that "move" takes an Agent Mary, an Object which is a book, and another argument which is a table. Initially the Arguments are not generalized at all. They are assumed to be at the exact level of specification in the example sentence. The Effects node says that the action takes the Agent from some location LocA to the table, moves the book to the table, deletes the attribute that the agent has physical contact with the book, and maintains the fact that the clock is on the wall.

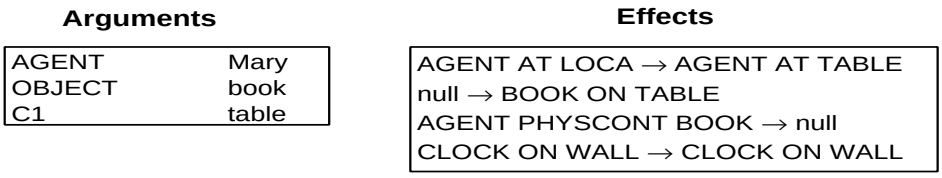


Figure 5.1: The meaning of "move" from "Mary moved the book to the table."

Deriving this single definition doesn't require much processing finesse. The power of Salveter's system came in combining different examples of the same word into separate but related definitions. For a word with multiple senses, the common attributes were kept in the original nodes, but the different attributes were split out into other nodes. Then the meaning of each sense of the word could be described as a subgraph of this larger graph. In addition, the system incorporated a generalization mechanism that could broaden the definition of words given multiple senses. So, if Moran were subsequently given representation for the sentence, "Joe moved the chair", along with the snapshot:

Before	After
JOE AT LOCB	JOE AT LOCD
JOE PHYSCONT CHAIR	CHAIR AT LOCC
CHAIR AT LOCB	CLOCK ON WALL
CLOCK ON WALL	

it would generalize some of the slot fillers and split others as shown in figure 5.2. The subgraphs defined by the arcs labeled "1" describe the meaning of the "move object to table" sense of move, and the subgraphs labeled "2" define the "move object" sense. The nodes with both labels specify the features that are shared by the different senses.

This method of representing meanings (similar to that described by Katz and Fodor [1963]) is very powerful. It allows for very compact storage of different verb sense definitions, and allows for the ability to periodically reorganize the knowledge structure, grouping arguments or effects that appear to be related.

A shortcoming of Salveter's system was that although it knew a little about the classifications of objects, it had no other knowledge about the world. It could not, for example,

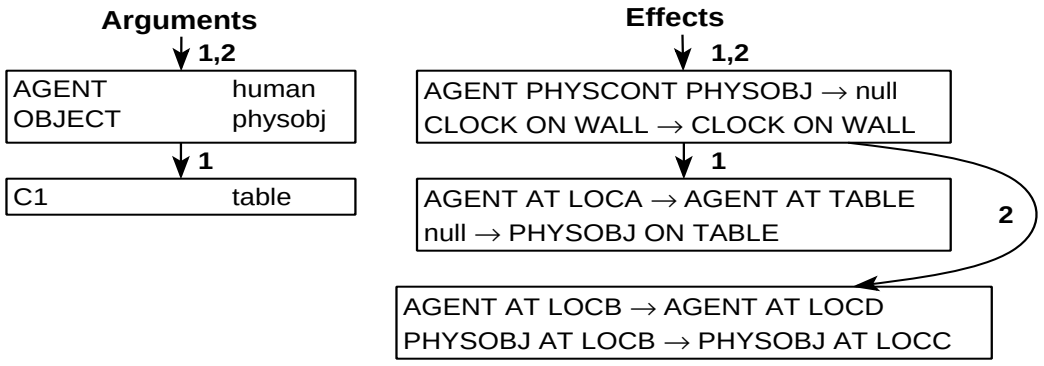


Figure 5.2: The meanings of "move" from adding "Joe moved the chair."

reason that the fact, "CLOCK ON WALL" in the above example was irrelevant, because it had no idea about the connection between the action and other things in the scene. Another limitation was its reliance on a human trainer. The system was able to derive very complete meanings for verbs (within the limits of Moran's knowledge representation), but it depended on the proper sequencing of examples from the trainer, as well as the proper descriptions of the visual scene and sentence deconstruction.

As previously mentioned, this system was not claimed to be psychologically valid, but it was clear that the task was set up to mimic, in some ways, the child's learning task. The system was never applied to a real-world task, but, by demonstrating what could be learned about different senses of a word, it laid an excellent foundation for future lexical acquisition work.

5.2.2 Selfridge's Child

Selfridge's Child theory and program [1986; 1991] are intended to model children's language acquisition. In order to evaluate this goal, Selfridge chose six "facts" about the way that children develop language:

- Comprehension precedes generation
- Vocabulary growth rate first increases then decreases
- Utterance length increases
- Irregular words are regularized
- Unlikely actives are initially misunderstood
- Reversible passives are initially misunderstood

On the basis of these attributes, he suggests an eight-stage developmental progression that children follow during the course of their language learning. These stages take the children from the age of ten months when they know no language at all, to the age of 5 years when they are able to understand reversible passive sentence and all actives, have a large vocabulary, and can produce arbitrarily long sentences. In general, Child follows this progression with the help of a "parent" (trainer), and a supplied representation of the visual input.

Based on this ability to account for these stages, Selfridge uses his model to suggest answers to several unresolved questions about children's language acquisition:

- How do children learn to recognize ungrammatical sentences?
- How do children learn an infinite language from finite data?
- How do children learn syntactic word classes?

Selfridge proposes answers to these questions based on the mechanisms that Child uses to process language.

As the basis of Child's knowledge representation, he uses a simple sort of Conceptual Dependency [Schank, 1973] formalism. This determines not only how domain knowledge is represented, but also how syntax is — or is not — used. Instead of a traditional context-free grammar-like mechanism, Child uses a purely positional account of grammar, specifying, for example:

> The actor of "give" precedes the word "give", the object and the recipient. The object follows the actor and "give", and precedes the recipient. The recipient follows the actor, "give", and the object.

As will be seen in the next chapter, this bare-bones grammatical mechanism seems to be countered by psycholinguistic evidence that by a young age, children have acquired quite sophisticated syntactic knowledge.

To learn words, Child starts with knowledge of function words (i.e. prepositions and determiners) and an assumed ability to perceive "visual input" (actually, a description of an action provided by the trainer, similar to the Moran's). The visual input consists of an action and a set of features, for example: `(PTRANS Actor (Father) Object (Ball) To (Top (Table)) Time (Past))` [Selfridge, 1986, p. 196]. Child assumes this to be the meaning of the input sentence. The input sentence is also provided by the trainer and is either an imperative or a declarative description of an action. The trainer can simulate intonational emphasis by capitalizing one of the words in the sentence. If a word is emphasized, Child assumes that the meaning of that word is the implied meaning for the sentence with meanings of known words subtracted from the set.

An example will help clarify this process. If Child is just starting to learn and has the visual input given above along with the sentence, "papa put the Ball on the table", it will take the entire representation to be the meaning of the word "ball". Given other examples of the emphasized use of "ball", Child restricts the meaning to be the intersection of the prior and current visual inputs. This has two major implications: First, Child meaning development relies heavily on the trainer's input. Second, polysemy (multiple word senses) is not handled.

It is interesting to note that Selfridge draws from Child the conclusion that "children use knowledge of known words to limit the hypotheses about an unknown word." [Selfridge, 1986, p. 210] Camille does the same thing, but in a totally different manner. Child has for each word a set of attributes that add up to the meaning of the word. For Camille, words have constraints about the contexts and combinations that they can be a part of. Instead of whittling down a provided meaning representation, Camille applies syntactic and semantic constraints to locate word meanings in the concept space.

Selfridge took on an ambitious project, attempting to explain much of the general phenomena involved in language acquisition. Because it was specifically developed to emulate children's behavior, however, its psychological claims must be scrutinized.[2] Certain assumptions are suspect, for example that the lexicon is implemented as a stack, with the most recently

[2] An infinite number of programs can be written to fit a particular curve.

learned definition for a word taken as *the* definition. Because Camille was not intended to emulate a certain behavior, any similarities with human cognition that it displays are more likely to be a result of general properties of the task.

5.2.3 Siskind's Davra

In developing Davra,[3] Siskind's [1991] goal was to demonstrate that two popular and competing psycholinguistic theories were computationally unnecessary. These theories both described strategies that children might use to apply what they already know to the task of learning the rest of their language. The first theory, *semantic bootstrapping* [Grimshaw, 1979; Grimshaw, 1981; Pinker, 1984], suggests that children use their knowledge of what a sentence means in order to determine the syntax of that sentence, and by repetition, their language. *Syntactic bootstrapping* [Gleitman, 1990] (see also section 6.3) states that children figure out grammar at an early age and use that knowledge to guide their acquisition of word meanings. Siskind's goal in developing Davra was to demonstrate that neither of these mechanisms is necessary for language acquisition. In order to do this, Davra deduced semantic *and* syntactic information from examples in a task setting similar to Child's and Moran's.

The input to the system consisted of simple sentences that were provided by a trainer, for example, "Bill ran from John." The system also received a set of *possible* meaning interpretations of an associated scene in the form of Jackendovian conceptual structures [Jackendoff, 1983], for example:

```
((BE(person3,AT(person1))) ∨ (BE(person3,AT(person2))) ∨
(GO(person3,[Path])) ∨ (GO(person3,FROM(person1))) ∨ ...)
```

Unlike the other similar systems, this meaning representation was not hand-coded. It was provided by an independent program that analyzed a stick-figure animation.

The underlying linguistic structure was based on Universal Grammar [Chomsky, 1981; Chomsky, 1985; Chomsky, 1986]. Specifically, Siskind encoded 12 principles into his system. Some of the principles defined Davra's basic abilities, for example that it had the ability to segment the input sentences and to comprehend the semantic representation. Other principles dealt with the specifics of the $\overline{\text{X}}$ grammar formalism and its parameter setting. Another set of principles provided simplifying assumptions, for example that the input would contain no polysemy, a single language, and only grammatical sentences.

The task for the system, was to process a set of example sentences along with associated scene interpretations, and compute the syntax and semantics for that sub-language. Fortunately for Davra, the simplifying assumptions really did their job. The only grammatical information that the system had to infer was if the sub-language was SPEC initial (i.e. articles precede nouns) or SPEC final and if it was head initial or head final (following the conventions of $\overline{\text{X}}$ parameters). For each word it had to infer the lexical category (noun, verb, or preposition) and a mapping for that word to a segment of one of the supplied semantic representations. In a training session, Davra received a small set of sentences (around 10) and 3 to 6 possible interpretations for each. Davra was able to compute all of the parameter settings and meanings for the 11 words. The only difficulty was that it couldn't decide if the prepositions it encountered were verbs, nouns or prepositions.

Siskind suggested that the key to success for the system (in the absence of one or both of the simplifying psycholinguistic strategies, presumably) was what he called "cross-situational learning."[1991, p. 159] The system required several different examples of a word's

[3] The predecessor to Davra, Maimra, is described in [Siskind, 1990].

use and several different syntactic structures to be able to make its inferences. This underscores a weakness of the system. Davra has the luxury of collecting all the appropriate (trained) evidence and then applying its substantial computational capabilities to calculate a consistent interpretation of the syntactic and semantic information. As will be further discussed in the next chapter, this type of unconstrained hypothesis testing approach is highly improbable for children.

Nevertheless, Siskind's approach is very interesting. He reportedly plans to devote further research to discovering if his method will scale up to larger domains and more complex input. If future research does find a way to extend this approach, it could be helpful in defining the possibilities of human language acquisition.

5.3 Script-based systems

Other systems rely on script-based information as their main source of lexical acquisition power. One of the earliest lexical acquisition mechanisms was developed by Granger in 1977. More recently, Zernik did an updated version of this work that was tailored toward dealing with certain complex verb constructions.

5.3.1 Granger

Granger [1977] developed one of the first systems that inferred the meanings of words from context. His system was called "Foul-Up" because when the NLP system encountered an unknown word, the parser could not continue without a special mechanism to doctor the parse structure. His program was implemented as an extension of a system called SAM, which was based on Schank's Conceptual Dependency framework [Schank, 1973] and analyzed news reports using scripts [Schank, 1981] so that they could be paraphrased. This section describes SAM's parsing process, the extensions that Granger made to acquire word meanings, and an analysis of the merits of his approach.

As previously mentioned, SAM was based on Schank's Conceptual Dependency (CD) framework. At the time it came out, this was a radical approach to processing natural language. The conventional wisdom was that understanding the syntax of a sentence was the key to understanding the sentence. In fact many systems had no semantic analysis components, on the premise that producing a parse tree was tantamount to understanding the sentence. Schank said that this approach was misguided — that the key to understanding natural language was in the semantics. In fact, he went so far as to say that performing the traditional *syntactic* analysis was unnecessary, that the structure of a sentence could be derived from expectations based on the meanings of the words. For example, the word "throw" expects to be combined with an actor that is a human, an object, and an animate recipient.[4] The actor would come before the verb, the object immediately afterwards, and the recipient in a prepositional phrase.

The ELI (English Language Interpreter) segment of SAM processed a sentence using the expectations described above, and transformed it into a set of primitive relations and objects. In the case where all the words of the input sentence were defined, ELI used the definition for the verb to set up the expectations (as described above) for the other elements of the sentence. The result was a frame which listed the type of action and the various slots and slot fillers for that action (e.g. Actor, Object, Destination).

[4]CHILD's rules are similar because they were derived from this theory.

In order to understand the connections between sequences of actions, scripts were added to represent common combinations of sentences. The power of scripts was described in section 4.2 — they allow the system to infer details that are not included in the story. They also allow a lexical acquisition system to make inferences about parts of the story that are left out or that the analyzer can't understand. The mechanism for this inference is described below.

To derive his treatment of unknown words, Granger used his intuitions about how humans perform the same task. The new word triggered a reaction by the ELI system that it should enter a place holder in its representation to record ELI's expectations, so that when a script was applied, those expectations could be combined with the missing script elements to make a hypothesis about the meaning of the unknown word. As an example, assume the system encountered the following sentences:

> Friday, a car swerved off Route 69. The car struck an elm. [Granger, 1977, p. 173]

If ELI didn't know the word "elm", it would complete processing by putting a place-holder into the OBJECT slot of a PROPEL frame. The place-holder recorded what the word was and that it had an indefinite article. Then the APPLY mechanism went to work, fitting the representation into an appropriate script. In this case, the script Vehicle-Accident had an unfilled Obstruction slot which takes a PHYSOBJ. ELI's partial representation for "elm" was consistent with it being a PHYSOBJ, so Foul-Up inferred that "elm" was a PHYSOBJ and put it into the script.

Granger attempted to learn the meaning of verbs as well and noted the difficulty in dealing with them because they provided the expectations and script triggers for events. In order to handle unknown verbs, Foul-Up took a four-step approach. First, ELI entered a place-holder representation for an action that set up expectations for all possible types of attachments. Prepositional phrases were attached using heuristics. For example "to", "towards", "into", and "at" fill the TO slot of a frame. Granger hand-crafted a table to give preferences for the type of action based on the prepositions and their objects found in the sentence. A sentence with the preposition "to" followed by a Locale preferred a PTRANS (physical transfer) action. Finally, a flexible match was done between the slot fillers for the preferred concept and the slot fillers that were generated by ELI. If the slots matched, the preferred action was taken as the definition of the unknown verb. If not, the system searched for a closer match.

Granger's work laid an firm foundation for future research. He showed the importance of using world knowledge for inferring verb meanings and pointed out the difficulty in learning the meanings of verbs which provide the constraints that apply to the other components of the sentence. His work was limited primarily by the weakness of the concept organization that was used to represent objects in the world. SAM used only five object categories, PHYSOBJ, LOCALE, HUMAN, BODYPART, and CONCEPT. This representation lacks two important sources of information. First, the coarse grain of the categorization scheme obscures much discriminating information that could help the system learn words. Bananas and bulldogs, although both PHYSOBJs, are rarely used in the same way in sentences. Second, because there were no connections between the categories, the system's ability to generalize or refine concepts were limited.

In order to make up for this lack of discrimination at the intra-sentence level, Granger relied heavily on prepositions that occurred in the sentences to suggest hypotheses. Because natural languages use prepositions in many different ways, this approach is bound to yield

limited success. Granger himself recognized this problem but didn't make suggestions for its solution. By using a much more powerful syntactic grammar, LINK can make fine distinctions in the syntactic structure of a sentence, and Camille can use these to influence word acquisition.

Granger's learning mechanism was based on his "intuitions about how the analogous tasks are performed by people."[Granger, 1977, p. 172] Unfortunately, he did little psychological analysis to back up his intuitions. His system is weak in terms of its psychological validity in several ways. First and foremost is the lack of discriminating information described above. We know that humans have different categories for different types of objects because all natural languages show varying treatment for different categories (e.g. basic level effects). Second, as I will discuss in the next chapter, psycholinguistic evidence shows that humans use syntactic information to a much greater extent than Granger's system does. Bowerman, for example, cites evidence that 17 month old children can distinguish subtle variations in syntactic forms and apply them to their lexical acquisition task.[Bowerman, 1983] Granger made a good start at defining the problem of automatic acquisition of lexical acquisition, and the research described in this thesis is in some ways an extension of his. By using more intra-sentential constraints, a richer semantic representation, and more syntactic power, Camille can accomplish more robust lexical inference.

5.3.2 Zernik's RINA

Zernik's [1987b] thesis work was one of the most significant prior implementations of a lexical acquisition system due to the breadth of knowledge sources that it used. His program, RINA, combined a hierarchically-structured lexicon with script-like information and planning information to demonstrate the ability to learn verb-particle combinations. The goal of the system was to parallel the task of second-language learners, acquiring difficult phrases from examples with the help of a trainer.

RINA did not have a traditional grammar. Instead all of its syntactic information and lexical information was combined in a phrasal lexicon. Each element in the lexicon contained three items, a pattern, a concept, and a set of presuppositions or constraints. In this respect, RINA's knowledge representation was quite similar to Camille's. Although LINK maintains a distinction between lexicon and grammar, both forms of information are represented by the same structures, and so can be easily combined. Furthermore, the grammar rules can contain lexical information, allowing them to serve as the types of patterns that Zernik uses. Because LINK's constraints are on the semantic constituents (for example the constraints are on the Actor of an action, not the Subject of the sentence), this aspect of Camille's representation has greater flexibility than RINA's. RINA makes up for this lack of flexibility by allowing its lexical items to inherit from more general items. Thus a phrasal lexicon entry like,

```
<Person1> take on <Person2>
```

can inherit information from a more general pattern for "take" and a more general pattern for "on". In this sense, RINA closely parallels the representation of Salveter's Moran.

Learning occurs during a training session when there is a mismatch between the presuppositions and the input. In an example (starting on p. 227), RINA was presented with the following text:

```
Jenny wanted to buy a car.
She took it up with her dad.
```

In the second sentence there is an ambiguity concerning the referent of "it". The word could refer to the car, or (as inferred from the information that Jenny wants something) an unresolved goal of Jenny's. RINA's presupposition processor inferred that in order for Jenny to take the car "up" somewhere, she would have to have it, and in order for her to want to buy it, she must not have it. Thus there is a contradiction, and RINA signaled its difficulty to the trainer: "Jennifer drove a motor-vehicle upwards with her father?" At this point the trainer signified that it was not an appropriate interpretation, and RINA asked for another example. After the trainer supplied, "She took up the problem with her dad", RINA inferred a new sense for the word "took". With additional interaction, RINA determined the importance of the particle "up" in this construction. This resulted in the construction of the pattern:

```
<Person1> take <Problem> up with <Person2>
```

with the associated concept MTRANS (or mental transfer).

Zernik's work is important in that it includes the use of planning information in the acquisition of new phrases. It was limited, however, by several factors. First, it was purely a demonstration system. The system was made to work on a few examples, but was not applied to a real-world test. Second, although the system was claimed to be a cognitive model of second language acquisition, no corroborating psycholinguistic evidence was supplied. Third, like most of the other systems described here, RINA suffered from a lack of discriminating syntactic knowledge. The patterns that it used were too coarse-grained to represent many grammatical structures. In a sense, Camille is an alternative approach to the same problem. Instead of seeding the system with a lot of special-purpose knowledge that will help with specific examples, Camille uses a simple, general approach and a strong grammatical representation to infer meanings without the help of a trainer.

5.4 Acquisition Aids

The systems described in this section are on the borders of the space of lexical inference mechanisms. The are intended as systems which aid humans in generating lexical knowledge. They are of interest because they delineate some of the boundaries of lexical acquisition. Furthermore, if the part of these systems that requires human intervention could be replaced to some extent by a machine learning system, then they could be considered full-fledged lexical acquisition systems.

5.4.1 Autoslog

Autoslog [Riloff, 1993] is not only geared toward the information extraction task, it exploits the knowledge base that the task provides. The system creates possible definitions of templates that might be useful for extracting text. It does this by examining the development keys which are basically examples of the filled-in database forms that the system should produce. For each field in the keys which is filled with a string value (as opposed to a number or a member of some set), AutoSlog searches for the target string within the associated message, and pulls out the sentence that it first appears in. The words in the sentence are tagged for part-of-speech by another system. Then, using a set of thirteen simple linguistic patterns, the system searches for a good "conceptual anchor point" for the desired text. For example, one pattern is:

```
<subject> passive-verb
```

The system uses the simple syntactic categories SUBJECT, DOBJ, and NP to specify the constituents before and after the verb, and after a preposition. In the pattern, they denote the relative position of the target text. If the sentence contains such a pattern, Autoslog proposes the verb as the conceptual anchor point for a mapping from the SUBJECT to the slot from which the original text came.

For example, assume that AutoSlog had a Terrorism key with "John" filling the VICTIM slot of a kidnapping. AutoSlog extracts the string "John" and searches for it in the text. With the sentence, "John was kidnapped," the system recognizes the pattern above and proposes the following definition:

```
If a SUBJECT  is followed by a passive form of ''kidnap'', put the
SUBJECT into the VICTIM slot of a KIDNAPPING template.
```

AutoSlog processes all of the available templates and associated messages in this manner. Then a human knowledgeable with the domain examines the proposed definitions and filters out erroneous ones. From 1500 answer keys and texts in the MUC Terrorism domain, AutoSlog produced 1237 definitions. In five hours, the human user winnowed this set down to 450 good definitions. The CIRCUS parser [Lehnert, 1990], using the AutoSlog definitions, produced Recall and Precision scores that were very close to those produced by the official UMass system which used a (painstakingly) hand-crafted pattern dictionary.

Can this be considered lexical acquisition? In one sense it can. Given that the ultimate goal of the system that uses these definitions is to extract information from text, these definitions satisfy exactly the functional constraints that were set out for Camille in Chapter 1. Because these simple pattern definitions allow the system to perform its task, why bother with additional information about the words?

One answer to this question is clearly that AutoSlog is so knowledge-poor, that it needs a human to check its work and throw out the bad parts. Camille exploits its domain knowledge to generate usable hypotheses.

The more fundamental answer can be found by comparing the different definitions that the systems create. As previously mentioned, the Terrorism domain contains 10 different words that refer to the concept Attack. Because AutoSlog creates definitions that are triggered by particular words, it would have to create a full pattern definition for all 10 words — and that is just for the passive constructions. It would need quite a few more definitions to handle the actives, infinitives, and gerunds. Camille defines these words as mapping to its Attack concept. This single node interacts with the general grammar rules to interpret each of these forms. Thus by inferring a deeper definition for a word, Camille can actually reduce its memory load.

5.4.2 MayTag

Recently Cardie developed another novel approach to lexical acquisition using a human trainer, not for presenting appropriate examples, but for developing a basis for bootstrapping [Cardie, 1993]. Her system, MayTag, works within the environment of an information extraction task and uses the CIRCUS parser [Lehnert, 1990]. CIRCUS, like Granger's and Zernik's systems, is a descendant of the Conceptual Dependency approach, and relies only minimally on the syntactic properties of the sentence. It works by designating certain concepts as triggers. When a word is found that refers to that concept node, the trigger is activated, and expectations are set up for arguments to attach to it. Syntax is only used to help distinguish major noun phrases. The system assumes that the NPs are either arguments to the verb or extraneous to the information extraction task. Thus, the parser must know

a word's part-of-speech, its semantic interpretation, and any task-specific concept nodes that it triggers. Unlike Granger's and Zernik's systems, which were intended for understanding stories, CIRCUS was specifically designed for the information extraction task. So CIRCUS does not use scripts. Its concept nodes generate portions of the template that serves as the output of the task.

MayTag infers lexical information by setting up a set of 39 feature-value pairs (including the word features mentioned above) that describe the state of the parser when an unknown word is encountered. Word features are the word itself, its morphology, and its global semantic interpretation.[5] Local parse features describe the immediate context of the parse — the word features of the two words before and the two words after the unknown word. Global parse features describe the global parser state — the semantic information for the major constituents (subject, object, verb group, etc) that have been recognized so far. This last set also contains information about the immediately preceding low-level constituent (noun phrase, verb, or prepositional phrase). The search space for MayTag is the range of possible values for these features.

To initialize the mechanism, a case base is constructed with the help of a trainer. To start, the parser processes a sentence until it finds an unknown word. (Lexical information for closed-class (function) words is assumed.) The parser fills in all the feature-value pairs for the known elements of the parser context. The trainer fills in the part-of-speech, semantic, and concept activation information for the unknown word.[6] In the experiments cited, 108 sentences were processed in this manner creating around 2000 cases.

The actual lexical inference process starts after the cases are built. When additional sentences are processed and new unknown words are encountered, the system matches the known parser features against the cases in the case base. A decision tree mechanism tunes the selection of features. The closest matches "vote" on the missing features — part-of-speech, semantics, and concept activation — for the unknown word. In experiments done on a narrow, full-text domain typical of the information extraction task, MayTag achieved a success rate of about 93% correct on part-of-speech, 80% correct on semantic interpretation, and 95% correct on concept activation.

These results look quite good. MayTag is geared toward the information extraction task and shares the same basic goals as Camille. Therefore, I will include an in-depth analysis of the system's performance.

At one level, MayTag and Camille are quite similar. MayTag's general and specific semantic features form a two-level knowledge representation hierarchy which is shallower than but still similar to Camille's.[7] MayTag's task is to infer part-of-speech and semantic information based on context.[8] It does this using a decision tree which ranks the predictiveness of features of the surrounding text. Based on examples of the decision trees which MayTag's C4.5 algorithm [Quinlan, 1992] made (provided by Cardie), it was clear that the previous and fol-

[5] The system distinguishes between local and global semantics.

[6] Note that there is a bootstrapping problem here. Because the parser state includes information about adjacent constituents, the system could end up defining unknown words based on other unknown words. To get around this difficulty, the system uses some basic heuristics to infer a default meaning for adjacent unknown constituents.

[7] The general features seem to roughly correspond to the higher-level or basic level concepts, for example, Human. The specific features correspond to lower-level concepts that are required for the task, for example, Officer.

[8] It also infers concept nodes, which should correspond quite closely to the semantic features.

lowing semantic nodes were the most predictive of the word's semantics and its part-of-speech. This is, in a sense, a roundabout way of doing what Camille does, predicting meaning based on neighboring constituents.

An interesting distinction between the two systems is that Camille assumes that words which are assigned the VERB part-of-speech label will refer to semantic nodes in the action subtree of the concept hierarchy. MayTag decouples these, inferring part-of-speech separately from semantic features. It would be interesting to check if Camille's assumption is a good one. Unfortunately, this is not possible based on the broad statistical analysis of MayTag's performance.

MayTag's inference is somewhat limited because of its lack of syntactic knowledge. Because it focuses on local features, it may not be able to utilize the constraints of constituents which are distant in the sentence but proximal in the syntactic structure. Its global context features provide a certain amount of this type of information, but they do not rival the rich syntactic structure provided by LINK.

Three other important features separate Camille and MayTag. Camille learns as a side-effect of understanding text. MayTag is a totally distinct machine learning procedure geared toward lexical acquisition. MayTag is not incremental. Based on its decision trees, it infers the most likely values for a word's features and sticks with that decision.

Camille is primarily geared toward learning verb meanings because of their importance for the information extraction task and for providing the structure for the interpretation of the sentence. MayTag makes no distinction between words, with one exception. It does not assign any semantic information to verbs. There are two possible reasons for this. One is that the task[9] doesn't rely on verb meanings. The other is that the verb meanings were simply too difficult to learn. In any case, this is a fundamental difference between the systems. Camille learns verb meanings. MayTag will only infer their part-of-speech.

5.5 Graph search mechanisms

There are only a couple of mechanisms that use graph search as their primary source of inference, and one of them is not intended for this purpose at all. LOOM is a general knowledge representation system that is oriented toward use with NLP systems. It shares with Camille its ability to search through a hierarchy for a node that meets certain constraints.

The systems described in the previous sections all use a minimal semantic hierarchy, with only broad categories like Human, and Physical-Object. MayTag used a two-level hierarchy that only distinguished general and specific semantic concepts. This lack of semantic discrimination limits the types of inferences that those systems can make. Bananas and bulldogs are physically different and do different things. They should be treated differently.

5.5.1 Classification Systems

Why is a classification system like LOOM [MacGregor, 1990] included with these lexical acquisition systems? The purpose of this and other such systems is to represent concepts and the relationships between concepts, and to support inference on those concepts. As part of this support, when a new concept is entered into the knowledge base, the system determines, based on its features, where in the hierarchy that concept should go.

[9] MayTag was evaluated on the MUC Joint Ventures domain, which shows many differences from the Terrorism domain.

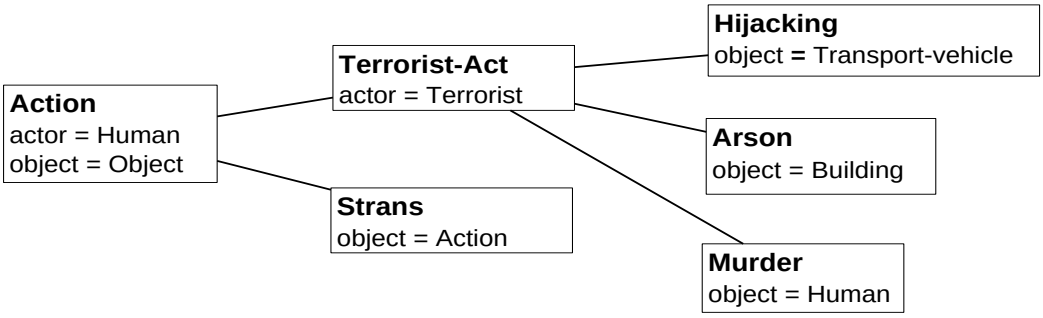


Figure 5.3: A simple terrorism subsumption hierarchy

An example will clarify this process. Assume that LOOM has already developed the simple hierarchy depicted in figure 5.3. A new concept, Frooble, is added to the system with the following attributes:

```
Frooble is-a Action
  actor = Terrorist
  object = Building
```

The classifier algorithm inserts this concept into the hierarchy where it is logically subsumed, under the Arson concept. In fact, because it has no reason to separate Frooble and Arson, it merges them into a single concept.

If the knowledge base is later told that this instance can also take a Human as an object, then the previous inference is incorrect and has to be retracted. The classification system now infers that Frooble should be located further up the tree. The concept is merged with the Terrorist-Act concept, because it takes a Terrorist as its ACTOR and an Object (assuming Object subsumes Person and Building) as its OBJECT.

Thus classifier systems provide a very similar inference mechanism to Camille's. Because they are not designed for the lexical inference task, however, they stop short of inferring the best hypotheses. If a classifier system received an example with a general slot-filler, it would make a general hypothesis. Camille makes the most specific hypothesis possible. Specific hypotheses have a higher information content and are more falsifiable.

5.5.2 Camille

This section contains a re-examination of Camille in terms of the relevant axes for lexical acquisition systems. This will provide a direct comparison and contrast with other systems which have similar goals.

Camille was designed for use in the information extraction task. As described at the start of Chapter 3, this has implications about the level of its domain representation and the type of processing used. Camille's more robust model of grammar, lexical, and semantic knowledge allows it to make more powerful inferences than the other information extraction systems.

Although several other researchers have observed the difficulty caused by inferring verb meanings, none have developed a general weak method to attack the problem. Others have glossed over the problem entirely. Camille reduces the verb acquisition task to a graph search problem. It uses the semantic constraints on its concepts to guide the search through its domain representation, always preferring the most specific concepts.

Because it is incremental, the system can recover from overly specific hypotheses. Its incremental nature also reduces Camille's processing and storage requirements.

Camille learns part-of-speech and semantic information about words. This is more complex than the simple patterns inferred by AutoSlog. It does not infer grammatical parameter settings like Davra or patterns like AutoSlog and Child, but these patterns seem to be too simple for general language understanding.

The system does not use any representation of the external context like Child, Davra, and Moran do. The addition of such a mechanism is not required for the information extraction task, but may be addressed in future research.

Camille is totally automatic. It does not require the help of a human trainer. Furthermore, it uses only the knowledge that is present for normal parsing.

Camille was not designed to work as a cognitive model, yet it displays many similarities with human linguistic behavior. The fact that this behavior was brought about by the requirements of the task rather than as a design goal of the system makes it more likely that the system will be predictive of other cognitive phenomena.

One of the biggest advantages of Camille over most of the other systems described in this chapter is that it has been systematically tested with real-world data. This has resulted in some unforeseen conclusions, for example, the difficulty of representing many texts with a script mechanism. Systems which are not tested on real world texts may demonstrate that a task is theoretically possible, but they do not necessarily prove that they have a good method for performing that task.

CHAPTER 6

RELATION TO PSYCHOLINGUISTICS

Camille was not intended as a cognitive model. It was developed to glean the meanings of words from context in order to create a more robust NLP system. The goal was purely computational. As Camille was developed, however, it became clear that some of its behavior was quite similar to that of children when they are learning language.

Why are these similarities important? Dennett gives an answer from the philosophical standpoint:

> A good psychology of Martians, however unlike us they might be, would certainly yield general principles of psychology or epistemology applicable to human beings. [Dennett, 1978, p. 113]

Any agent which successfully processes the same sources of knowledge that humans do must have *something* to say about the properties of that knowledge that make it amenable to processing in general, and about how other agents must process it.

Computationally speaking, it appears that the lexical acquisition task is so inherently difficult (it has resisted computer solution for a long time) that the task itself forces *any* solutions to conform to some sort of overall qualitative structure. If this is true, then any solution to the problem will also have some extent of predictive power as to how other solutions (human or computational) will work. The extent of this predictivity will depend on the proximity of the solutions.[1] In order to assess the similarity of Camille's processing to human language learning, this chapter describes related psycholinguistic findings and an analysis of how they relate to Camille as a cognitive model.

6.1 Fast mapping

Young children are remarkably adept at using a small amount of information to drive their learning process. In order to learn lexical items, they normally receive a small number of positive examples of how the word is used in context (maybe repetitions of the same phrase), and very few negative examples that would tell the child what uses of the word do not lie

[1] There must be a continuum of cognitive modelness. No silicon model is equivalent in every way to a neural model, so at the lowest level (Marr's mechanism level or Newell's physical level or Pylyshyn's "low road" [Marr, 1982; Newell, 1990; Pylyshyn, 1989]), there can be no absolute equivalence. Cognitive models must all, therefore, abstract to some level at which they assert that their process is the same or similar to human processing. At the other end of the spectrum, all computer programs that perform some task that humans do can claim at least some level of similarity to human processing. The task that remains then is to locate a particular model on this continuum.

within the language (the "no-negative-evidence problem", section 6.2). Despite this lack of evidence, children very quickly figure out what words mean. The process that children go through in assigning meaning to new words has been described as "fast mapping" by Carey [1978]. She suggested that children are forced to make poorly informed, quick guesses at how new words map to the internalized concepts.[2] Carey and Bartlett [1978] tested children to see how much they could learn about color terms with no explicit teaching but examples like, "Bring me the chromium one, not the blue one, the chromium one." The children were tested a week later to examine their recall of the word. They found that at least for these color terms, the children's learning abilities were remarkable. More than half of the children demonstrated a week later that they had learned something about the word that they had only heard on that one occasion.

A more recent study by Heibeck and Markman [1987] expanded on that earlier work. They took three groups of children, aged two, three, and four, and tested them using a procedure similar to that described above. They expanded the test to include texture and shape terms as well as color terms to see if the results might have been specific to just one particular domain. In addition, they changed the testing procedure to explore just how much information the children gained from the examples. After a brief delay from the time that they were given the original example, the children were tested to see if they could produce the word that they had just learned. Next they were tested to see if they could tell what domain the word came from. For example, the children were asked a question like, "See this book? It's not chromium because it's _____." If the child responded with another color term, he or she was credited with knowing the domain of the target word. Finally, the children were given a comprehension test to see if they could remember the meaning of the new word (even if they could not produce it) by being asked to show the tester an object of the particular color, shape, or texture.

The results of their experiments demonstrated that with very little difference between age groups, the children performed quite well at deriving word meaning information from the brief examples given to them. Together, these findings argue against a competing theory, the "unconstrained hypothesis-testing" view of lexical acquisition, which suggests that learners simultaneously consider many hypotheses for a word's meaning, collecting evidence for each and carefully evaluating them against each other. In the context of the experiment mentioned above then, the child would have to maintain the possibility that the word chromium could refer to the shape of the item, its texture, or any of a number of other attributes of the item. Instead, we find the children applying whatever information is available to them to make a quick "guess" as to what the unknown word could mean. This inferential task is made even more difficult by limitations in the available evidence, as described below. The task required of Camille requires it to act in a similar manner. Because the system is incremental, it must make guesses about words as it goes along. This allows it to quickly hypothesize meanings for words although the initial hypotheses may not be correct.

6.2 The No-Negative-Evidence problem

When children learn language, they must induce the structure of the language relying almost entirely on examples of utterances which are *within* the language [Bowerman, 1983]. They do not have the benefit of negative evidence that would tell them which possibilities

[2] The question of which comes first, the concept or the lexeme, is an interesting one that will come up again in section 6.5. It is not clear whether a novel word signals the child that a new concept should be created, or if the concepts already exist in the mind of the learner, waiting to have words attached to them.

to rule out. Although this absence of discriminating information makes the learning process computationally very complex, children *do* learn language. The Subset Principle was described by Berwick [1985] as one way that children could reduce the complexity of the grammar-learning task. This principle suggests that children have an innate hierarchical mental representation of language structure which is ordered on the specificity of the grammars. When learning syntax, children initially hypothesize the most specific grammar that accounts for the "data" that they have encountered in order to avoid over-generalization. Then, if they hear sentences that aren't covered by the initial strict grammar, they choose the next most specific grammar that includes the new utterance.

Despite the fact that Berwick's model has been criticized for the amount of innate knowledge that it requires on the part of the child language learner, it illustrates an important general principle. The child can overcome the difficulties of the no-negative-evidence problem by tending towards making hypotheses about language that are more strict than the evidence warrants, but that have the advantage of being more falsifiable and therefore being stronger hypotheses. For these hypotheses then, corroborating evidence can be taken as *confirming* the hypotheses, because evidence to the contrary is relatively more likely to be encountered.

For learning word meanings, there is evidence that children (and parents) actively try to make up for the paucity of negative examples in everyday speech. Shatz and Ebeling [1991] enumerated four different types of *language learning-related behaviors* that children engage in to interactively learn language. One of these four, *language lessons*, was most often used in order to provide word-meaning information, for example:

```
Parent:  What color is that?
Child:  Blue.
Parent:  Green!
Child:  Green.
```

These lessons, which accounted for 12% of the language learning-related behaviors, are one way for children to receive negative evidence about word meaning.

Bowerman [1983] described another method that children might use to help them learn word meanings in the face of the no-negative-evidence problem and compares it to innatist theories like Berwick's. On the "nurture" side, she suggested how, when learning words, children could use what they've learned about syntax (more about this in the next section) to make predictions about how particular verbs can be used. As an example, she described (p. 33) how a child might have learned the general principle that verbs that involve direct causative action can be used in the lexical causative form, as in, "I broke the stick." Other verbs which involve less direct causation, can only be used in the periphrastic causative form, as in, "I made the stick disappear." But if the child sees a magician directly manipulating a rabbit to make it disappear, she might make the prediction that "disappear" can be used in the lexical causative form. This sets the stage for a mismatch between the child's expectations and what she hears. Upon encountering the periphrastic use of "disappear", the child will realize that her prediction was wrong. This could provide the type of indirect negative evidence needed to counteract the no-negative-evidence problem.

The important point here is that learning language is an underconstrained problem. Children don't have sufficient evidence to deductively derive the meaning of a word or an appropriate grammar. Thus, they must make many "guesses" during the course of language acquisition. These guesses in a sense add information to the data that they hear. By making highly falsifiable hypotheses, children can assume that without evidence to the contrary, their hypotheses are correct. The next section describes how children can leverage their early knowledge of syntax to guide the semantic guesses that they make.

As previously mentioned, children do receive some negative evidence with regard to word meaning. Because Camille is not trained, it never has the benefit of direct negative evidence. So in learning the meanings of words, it is in a similar situation to children learning syntax — positive examples only.

6.3 Syntactic bootstrapping

Gleitman [1990] described a mechanism called *syntactic bootstrapping* that children might use to guide their search for meanings of verbs through the space of possible meanings that could be inferred from the immediate context. She gave striking evidence that children who can barely produce two-word utterances are capable of using syntactic information embedded in the sentences they hear to constrain interpretations of new words. In an experiment which used the *preferential looking procedure* originally designed by Spelke [1982] and adapted for linguistic use by Golinkoff *et al.* [1987], 17-month-old children who had no prior knowledge of the word "flex" were shown two videos, one which showed Big Bird and the Cookie Monster crossing and uncrossing their own arms, and another with one of them crossing the arms of the other. When one of the sentences "Big Bird is flexing with Cookie Monster" or "Big Bird is flexing Cookie Monster" were broadcast through a speaker, the children showed a definite preference for the "syntactically congruent screen," i.e. the video that was showing the action that was consistent with the linguistic input, even though they had no semantic knowledge of the meaning of "flex." Gleitman argued that without such a constraining mechanism to limit the appropriate hypotheses, the task of word learning would be computationally infeasible.

Bowerman's work [1983] also relates to the idea of syntactic bootstrapping. In order to make the distinctions in meaning that she describes, children must be able to distinguish the lexical and periphrastic causative syntactic forms. The differences in syntax are the keys to learning some of the subtleties involved in the different meanings.

Naigles' [1990] offers experimental results that validate Gleitman's hypothesis that very young children are capable of using syntactic information to choose between different verb senses. In addition, she commented on the likelihood that learning does not occur solely on the basis of a single input, but is "gleaned from the presentation over time of the verb in its particular set of syntactic frames." [Naigles, 1990, p. 371] Although at first glance, this might seem to conflict with the "fast-mapping" hypothesis, there is a reconciling explanation. As Carey and Bartlett suggest, children make a quick guess at what a word means. This guess not only provides a concept that the word maps to but also a set of predictions about how it will be used in other constructions. If over time, these predictions or the concept mapping conflict with new evidence, the learner can incrementally refine the hypothesis. Thus, learners have the advantage of an early, usable idea for what the word could mean along with a mechanism for improving that idea over time.

Hirsh-Pasek and Golinkoff [1993] include this work as part of what they call "skeletal supports" for language acquisition. They suggest that this sensitivity to the arrangements of words, along with an acoustic sensitivity that allows children to separate spoken sentences into words in the first place, form the basic capabilities that children must have to learn language.

Camille takes advantage of its syntactic knowledge in a similar way. By inferring the syntactic structure of the sentence, it can determine the case fillers of the verb and use that information to infer possible meanings for the verb if it is unknown.

6.4 Objects, actions, nouns, and verbs

When psychologists first started studying language acquisition and the types of words that were acquired first, a striking observation was made. Children were learning nouns well before and at a faster rate than they were learning verbs. This prompted several studies of the differences in acquisition between the various types of words and theories about what causes those differences. These studies suggest that differences in the knowledge representation for nouns and verbs force children to use different mechanisms to learn them.

Gentner [1978, pp. 988-989] cites several studies that describe differences in acquisition between nouns and verbs. Some of these studies showed that young children's initial vocabulary consists entirely of nouns with verbs slowly making their way in. Others showed that the first verbs took almost twice as long to appear as the first nouns. A study of comprehension and production by Goldin-Meadow, Seligman, and Gelman [1976] showed two stages of early lexical development. In both stages, many more nouns than verbs were comprehended. Only a portion of the comprehended verbs were produced in the second stage, and none were produced in the first. Finally, Gentner described an additional study that demonstrated that the differences in acquisition are not just attributable to differences in the frequency of verbs versus nouns that the child hears. Even when presented with made-up nouns and verbs, and when balancing the presentation of these new words, children first used verbs an average of 8 months after starting to use nouns.

These results are tempered somewhat by recent suggestions that the findings may be specific to the English language. Gopnik and Choi [1990], in a study of the correlation between linguistic and cognitive development, cite studies that Korean- and Japanese-speaking children show a higher use of verbs during the one-word stage than English-speaking children do. They attribute this difference in behavior to structural differences in the languages. The Korean and Japanese languages both place verbs in the final position of an utterance which makes them more perceptually salient to the child. Furthermore these languages allow liberal deletion of nouns. In a study of maternal speech by Japanese and American mothers, Fernald and Morikawa [1993] found that there were differences in the overall distribution of nouns versus verbs in maternal speech but they attributed this to cultural differences instead of differences in the structure of the languages. As a result, they found that Japanese children between 12 and 19 months increased their use of verbs about the same amount as did American children of the same age, but they increased their use of object labels significantly less. This underscores the importance of the input in lexical acquisition.

Along a slightly different vein, Behrend conducted in-depth studies of different types of verbs to compare children's comprehension and production among these various verbs [Behrend, 1990]. The types of verbs that he studied were those that described actions (e.g. "squeeze", "pound"), results ("flatten", "break"), and instruments ("hammer"). He found that when labelling actions ("What is the person doing?"), children are more likely to use an instrument verb than an action verb. This seems strange for two reasons. First, the learning biases for the "fast mapping" procedure described above suggest that a child should make the best possible guess about the meaning of the word given what she knows. All other things being equal, this should correspond to the type of verb that occurs most frequently in the language. But instrument verbs are far less frequent than action verbs are. Second, instrument verbs carry more information than action verbs and are therefore more specific. Thus the children in the experiments were labeling the events with the *most specific* label possible. This contradicts the results found in acquisition of nouns, which demonstrate that "specific subordinate terms are used much less frequently than basic-level terms as labels for familiar

objects." [Behrend, 1990, p. 694].

What could explain these psycholinguistic results? They suggest a difference between the internal mental structures that nouns and verbs map to. Gentner calls the basis for this difference the "referential / relational" distinction. Nouns normally refer to objects or "thing-like elements." Objects (at least concrete ones) tend to be highly constrained by the physical world. Hence, similar objects share almost all the same attributes. On the other hand, verbs tend to express relationships between objects or changes in those relationships. Relationships are more abstract and less easily perceived by humans. In fact, because there are basically an infinite number of imaginable relationships between any pair of objects or events [Bowerman, 1976], children must rely on linguistic input to inform them what relationships are culturally important. Because very young children have not fully developed this knowledge source, we expect them to focus on the more compelling perceptual aspects of their environment. Under this assumption, it is clear why they learn nouns first: nouns refer to objects that they can see. Verbs refer to relations which tend to be less constrained by the physical world, so their meaning components "cut across all semantic fields." [Behrend, 1990, p. 694].

What kind of mental representation can account for the differences that these data suggest? The representation that Gentner espouses is a semantic net, in which meanings are built up compositionally by referring to more basic elements of meaning. She suggests that research shows that relational meaning is more componential than object meaning, and that children acquire this meaning piece-by-piece. She admits, however, that this model of componential meaning "accretion" is not sufficient to account for all language acquisition.

Gentner's representation focuses on the representation of single nodes of meaning. Behrend supports Huttenlocher and Lui's proposal [1979] that these differences in behavior are caused by the overall structure of the representation. They suggest that objects are organized in a structured hierarchy so that nearby elements share many of the same features. The relational elements that are expressed by verbs are represented in a matrix structure with nodes connecting across the various object hierarchies. Graesser, Hopkinson, and Schmid [1987] have recently done experimental testing to support this hypothesis. The subjects were asked to sort sets of words by similarity. The findings suggested that people tend to sort nouns hierarchically while verbs were less structured and more "cross-classified." The cross-classification of Camille's concepts is discussed in section 6.7.3.

6.5 Formation, alteration of concepts

Recent psychological evidence suggests that infants have some conceptual capabilities. Mandler [1988; 1992] cites evidence that children as young as 3 and 4 months old display some attributes of conceptual representation. Six and seven month-old children have shown symbolic functioning, and 9, 10, and 11-month-old children have demonstrated recall capabilities. Mandler further suggests that her hypothesis is consistent with findings in regards to the early linguistic development of children. She says that, by the time they start talking, children have solid foundations for such concepts as Containment and Support. This allows them to easily and efficiently acquire meanings of such spatial function words as "in" and "on".

The relevance of this to Camille is that its constraint of an *a priori* concept representation is not totally outrageous. Psycholinguistic theories suggest that a significant amount of concept knowledge is in place when children start learning language.

Section 4.4 described one method by which linguistic input can influence the concept representation. The further development of the crucial interaction of linguistic and concept acquisition will be a topic for future research.

6.6 Biases / constraints for learning

One of the most popularly cited mechanisms for children's use in reducing the complexity of their language acquisition task is for them to employ some sort of heuristic rules to cut down the number of hypotheses that they must consider. Markman [1991] described three of these assumptions that children might use when learning new words: the Whole Object, Taxonomic, and Mutual Exclusivity assumptions. When a child sees an object and hears a spoken word that refers to it, that word could theoretically apply to any of a number of features of that object. It could specify the color, the texture, or the weight of the object. It could refer to one of the pieces that make up a complex object or to the combination of the object and the arm of the person holding it. But children assume (usually successfully) that the referent of the new word is the object and nothing else. This is called the Whole Object Assumption.

As previously mentioned, there are an infinite number of imaginable relations between objects that children could attend to. Interesting results are found in children's use of thematic versus taxonomic relations. Thematic relations are based on co-occurrence in common situations, for example cows and milk. Taxonomic relations are derived from a structuring of objects into classes, for example mammals or farm animals. Studies showed that thematic relations are particularly salient to children. When asked to find an unlabelled object that was like a target object, they chose a thematically related item. If the object was labelled with an unfamiliar word, however, they chose a taxonomically related item as the referent. This taxonomic constraint was proposed as a method that children use to help them learn appropriate names for object categories.

By applying the Mutual Exclusivity Constraint, children can use their existing lexical knowledge to limit what a new label can mean. This assumption tells them that there will not be two different names for the same thing. So if they encounter an object that they already have a label for along with the novel word, children will assume that instead of being a label for that object, the new word applies to some other aspect of the object. Note that there can easily be interactions between the various assumptions. For example, if the word "cup" is known, and a parent points to a cup's lid and names it, the Mutual Exclusivity Constraint can help the child override the Whole Object Assumption and decide on the proper meaning for "lid."[3]

Clark [1989] described a related rule that adults use in language, and showed experimental evidence that children use it too. Similar to mutual exclusivity, the Principle of Contrast holds that no two words are exact synonyms.[4] Clark pointed out that even for words like "cop" and "policeman," where the extension or reference set of both words is surely the same, another aspect of their meaning differs; that is, the conversational context in which each is likely to occur. Thus, instead of allowing children to infer that two words are complete synonyms, this principle forces them to explore other meanings or aspects of meaning. These

[3] This constraint has a weak and a strong version. The weak version is the one presented here, and restricts each object to having exactly one label. The stronger version takes this one step further and has implications for knowledge representation. It says that *categories* are mutually exclusive. If something is a "dachshund," it can not also be a "dog." This implies that there can be no hierarchical structure in the knowledge base, just a set of collections. Proponents of this theory do not suggest that hierarchical structures never exist in the mental representation, however, only that Mutual Exclusivity is used as a heuristic at an early age to speed lexical acquisition. Eventually, it is abandoned. Furthermore, Mutual Exclusivity can be overridden under certain circumstances, as in the case of the Taxonomic Assumption example above.

[4] The Principle of Contrast can be viewed as a weaker version of Mutual Exclusivity, although the latter only stresses the extension of the word.

various assumptions can combine (in poorly-understood ways) to make the complex cognitive task of learning word meanings more tractable.

Most of the psycholinguistic work with the Mutual Exclusivity constraint has been concerned with learning object labels. As mentioned in Chapter 4, Camille provides an excellent testbed for applying Mutual Exclusivity to verb acquisition. The analysis of this mechanism is included in section 6.7.1.

6.7 Implications of cognitive aspects

What is more interesting, an apple that tastes like an apple, or an orange that tastes like an apple? Clearly, the orange is more interesting.

To give a more extended example, consider a pioneering aeronautical engineer who wants to build a flying machine, but doesn't realize that there are animals that fly. She just knows a bit about lift and drag and aerodynamics. Using this knowledge, she makes a machine, an airplane, that flies.

One day someone says, "Hey, birds fly and they don't fly anything like that. There's nothing similar about birds and your flying machine." She says, "Fine, I didn't mean to imitate birds, I just wanted to make a machine that could fly." Our engineer is intrigued, though. There's a natural mechanism that does the same thing as her flying machine, but apparently it does it in a totally different way. So she studies more about how birds fly. She sees that they flap their wings and use that motion to generate lift *and* thrust. Her machine generates thrust in a completely different way. But to generate lift, her airplane uses wings that have a particular shape. Bird wings seem to have that same general shape. Now that is interesting . . .

The point of all this is that although Camille wasn't intentionally developed as a cognitive model, it does perform a task that humans perform and it is thus capable of rendering interesting insights on human language learning. An analysis of some of the similarities to psycholinguistic findings is included here, but because it is not the primary focus of the thesis, testing of many of the cognitive aspects of Camille is left to future research.

This section analyzes three different aspects of Camille in terms of what they predict about learning in humans: the use of constraints, how input relates to learning, and how the conceptual organization relates to learning.

6.7.1 Mutual Exclusivity

As previously mentioned, Camille includes a simple version of the Mutual Exclusivity constraint for the word-learning mechanism. Despite the fact that most psycholinguistic literature deals with the application of Mutual Exclusivity to nouns, Camille can apply it to verb-learning as well. This brings up an issue that is not dealt with particularly well by either the psychological literature or Camille. The difficulty stems from the fact that when learning an unknown verb, the system can entertain multiple hypotheses, all of which are consistent with its experience. So the question becomes, "How certain of a meaning hypothesis must the system be in order for it to rule out that hypothesis for other new words?"

Since there have been no thorough studies of the use of biases in learning labels for actions, this issue has been largely ignored by psychologists. But there is a more general question that subsumes it: "How can children (or computational models, for that matter) choose between several consistent meaning hypotheses?" If the child has a clear preference for one type of meaning over another, the first problem goes away. But there is little evidence

for how children might construct such preferences. Behrend's work suggests one possible mechanism, choosing the most specific hypotheses for verbs and preferring the basic level for objects, but it is clear from Camille's implementation of the Mutual Exclusivity constraint that more work needs to be done to explain these phenomena fully in psychological terms.

Another interesting topic brought up by this addition is the question of when Mutual Exclusivity should be overridden. It is obvious that at some point children realize that both "animal" and "dog" can apply to the same object, but what conditions allow for the constraint to be overridden? This is the topic of current psycholinguistic research, and future research with Camille.

6.7.2 Input and Learning

Although the testing of Camille was done on input sentences picked randomly from a large corpus, it is clear from the results that there are some important factors that are influenced by the input to the system, and that variation in the input can improve or degrade the system's performance.

One prediction is evident from Camille's use of the hierarchical structure which distinguishes general from specific concepts. Because Camille assumes the most specific meaning for an unknown word, general words will be given overly specific hypotheses unless they are encountered in the input with a variety of slot fillers. Thus if a child were to receive only input sentences like, "Get the string" and "Get the twine", Camille predicts that the child would infer an overly specific meaning for "get" like Tie. In other words, reducing the breadth of input that a child receives should result in overly specific hypotheses.

No psychological tests have been performed that address this prediction directly, but there has been work which more broadly addresses the role of input in learning. First, the previously described Fernald and Morikawa [1993] work concluded that differences in distributions of nouns and verbs was due to differences in maternal input. Second, Huttenlocher et al. [1991] showed that despite previous predictions that early lexicon size would be dependent on learning capacity, children's lexica are instead related to how talkative their mothers are. In general, children who receive more input have larger lexica.

6.7.3 Concept Organization and Learning

The organization of the concept representation is, as mentioned, an important aspect of Camille's implementation. The general framework consists of an IS-A inheritance hierarchy, a type of representation that is widely used in Artificial Intelligence. Various psychological studies support the existence of hierarchical structures in the brain ([Kaplan *et al.*, 1990] and [Keil, 1991], for example). At the lowest level, this representation is clearly not "brain-like". It is highly unlikely that the brain uses such a rule-like arrangement for representing constraints. But the hierarchical structure has advantages that make it a powerful representation scheme for computers and humans. This format makes it easy to make generalizations, an ability that is a key to learning and reasoning. It also provides efficient storage of information.

These advantages are most easily seen in the case of representing objects. Here, the hierarchical scheme allows for similar objects to be located proximally in the representation, even across different types of objects. Thus for natural kinds like animals, dogs can be stored close to wolves, somewhat further from cats, and rather far from insects. These distinctions can be made based on physical attributes which humans use to delineate natural kinds. For artifacts, the same distance attributes can be found, but the distinctions can be made on

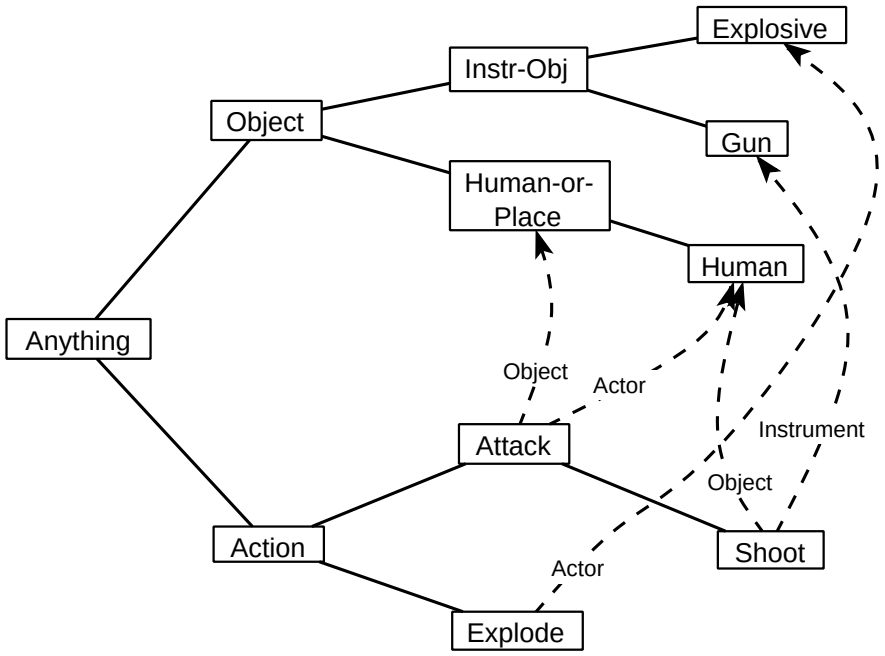


Figure 6.1: Matrix-like organization of action concepts

functional values, grouping, for example, kitchen appliances together. This type of arrangement allows for the efficient storage of constraints like, "mammals are warm-blooded" and "kitchen appliances are used for food-related activities." Finally, this structure makes it easy to tell when Camille needs further discrimination in the concept representation. Given an input sentence like, "I took my Queensland Blue out for a walk," we can infer that Queensland Blue is a type of dog, even if we've never heard of that particular breed. The ability to make this type of extension was discussed in section 4.4.

For representing actions and relations, the situation is somewhat murkier. As previously pointed out, although some psycholinguistic researchers have postulated a hierarchical scheme for their representation, recently the focus has turned to more "matrix-like" schemes. But the latter approach may be seen to conflict with the observation about the nature of constraints provided by the input and whether an upper bound or lower bound is created on the set of possible meanings for an unknown word. It's just not clear what "lower bound" would mean in a matrix-type organization.

On close inspection, it appears that Camille's representation has the best of both worlds. If the slot-filler constraints are displayed graphically (see figure 6.1 where the solid lines represent paths in the IS-A hierarchy and the dashed lines represent constraints on the actions), it is apparent that the connections do (as Huttenlocher and Lui put it) cut across the various parts of the hierarchy. This leads to the question, "Does it make sense to have the additional structure imposed by enforcing a hierarchical structure on *actions*?" The answer appears to be yes, for the same reasons given for object representation above. The hierarchy has representational strength — it allows for efficient storage of the attributes and constraints of actions.

The question that remains then is, "What does this imply about human concept organization?" For one thing, it lends support to the idea that there can be multiple organization structures within the brain. There are clear advantages to having both types of concept rep-

resentation. In addition, it suggests that learning could proceed in one of two ways. If a child realizes that her idea of what a word means is wrong, she should look for concepts that are closely related in the hierarchy. If the child's hypothesized *constraints* for the word are wrong, she should change those constraints based on the structure of the hierarchy that is selected by the matrix links.

Again, only partially related psycholinguistic studies have been performed. Bloom et al. [1980] created a simple hierarchy of early verbs based on their case frames: Action and State verbs were the most general. Action verbs were broken down into Locative (specifying a source or destination) and Non-Locative verbs. Locative verbs were further broken down into Mover, Patient, and Agent Locatives. Then they analyzed the speech of children who were just beginning to use inflections. They found significant differences between verb groups in the frequency of inflection use and the order of emergence of the inflections. They concluded that, "The semantics of the verbs that the children were learning was the major influence on their learning of verb inflections." [Bloom *et al.*, 1980, p. 404] This supports the general notion of the relationship of semantic structure to learning.

Tomasello's [1992] analysis of a diary of his daughter's speech starts with the contrasting assumption that verbs are initially totally disjoint, i.e. that there *is not* an overarching structure which guides the child's learning of verbs, their argument structures, and inflections. From this "Verb Island Hypothesis", however, Tomasello suggests that the process of learning the relationships between the (initially disjoint) verbs is the key first step toward learning grammatical relations. Thus he is making the developmental argument that verb categories do not innately exist, but they are *learned*, and learning these verb categories is crucial to language learning in general.

CHAPTER 7

CONCLUSION

This thesis has presented a computational account of lexical acquisition from context and its psychological implications. The main research goal was to make the best possible inferences from context. The implementation of Camille has pursued that goal by concentrating first on leveraging the information found within an example sentence. Only after it completely exploited this knowledge source was the context expanded to include multiple sentences. As a result, this thesis provides the most thorough explication of the power of linguistic context in lexical acquisition to date. This chapter summarizes the specific contributions of this line of research.

7.1 Major contributions

The most fundamental observation that comes out of this work is the dichotomy between inferring noun meaning and inferring verb meaning (in section 3.1). This comes as a direct result of attachment of semantic constraints to the action concepts. This is not an artifact of Camille or LINK. It is a fundamental attribute of language. The verb and the action it corresponds to serve as the center of the representation of the sentence. Thus, they serve as the logical focus for the representation of constraints as well.

The constraints specify an upper bound on the concepts that can fill them. Therefore, if an NLP system does not know the meaning of a noun, but it can determine what slot it fills, it can deduce an upper bound on the interpretation of that noun. Conversely, if the meaning of the verb is not known, a lexical acquisition mechanism can only deduce a lower bound on the interpretation of that word. The primary focus of this research has been in developing a method to counteract the vagueness that results from the lack of an upper bound for unknown verb meanings.

The solution is for the system to *guess* the most specific consistent meaning for the verb. This supplies a tentative, highly falsifiable, upper bound that can be corroborated or rejected by later instances of the word in context. If the original hypothesis is rejected, the domain representation is searched for a different concept that is consistent with the slot fillers that have been encountered.

A corollary to this observation is that it is likely that general verbs will occur in text more frequently than will specific verbs (in section 3.2.3). This is a straightforward result of the fact that general words can occur with more slot-fillers. Because Camille infers the most specific consistent meaning, it is successful for highly specific verbs, which it may encounter infrequently, as well as for more general verbs, whose frequency of occurrence forces the hypotheses to the appropriate level of generality.

The theoretical observations about knowledge representation that are found in appendix A came about as a side-effect of this work. They are important general issues to consider when creating Natural Language Processing systems, especially those that learn.

Camille is an incremental system that learns words as they are processed. A batch system could, theoretically, produce better inferences than Camille by examining, at one time, all of the available evidence about that word's meaning. The batch approach, however, conflicts with the inherently sequential nature of language. Humans do not have the luxury of waiting to make hypotheses about linguistic elements. One can never know when another instance of an unknown word will be encountered. Furthermore, the storage and processing requirements of such an approach would be prohibitive. Camille's incremental graph search mechanism allows it to make the best possible inferences given the evidence it has encountered. By choosing the most specific concepts, it reduces the size, increases the usability, and maximizes the falsifiability of its hypotheses.

Unlike most of the other lexical acquisition systems that have been developed, Camille is fully automatic. It learns from example sentences as a side effect of understanding them. Although children make use of interaction when learning language, much of their linguistic acquisition is performed before they create multi-word utterances. Thus, although adults often alter their speech toward young children, they can't perform reactive training at this early stage. The fact that Camille does not rely on a trainer attests to the strength of its learning mechanism.

Another distinction of this work over previous approaches to lexical acquisition is its emphasis on maximizing the use of context from *within* sentences. Some of the early lexical acquisition systems largely ignored the information available from intra-sentence context. Instead they relied on script-like mechanisms. Unfortunately, these systems were not tested on real-world data. As the tests on Camille indicated, although scripts can be a powerful source of information upon which to base lexical inference, the application of scripts to texts is quite tricky if the system does not know the triggering verb's meaning.

This thesis has described, from several viewpoints, those learning environments that lead to rapid acquisition. Section 3.5 contains an analysis of the behavior of the basic system. Because it infers the most specific hypotheses, Camille learns verb meanings most efficiently if the examples that it encounters pair general verbs (that is, those at a high level in the hierarchy) with a wide variety of slot fillers. Section 4.2 explained why some corpora cannot be easily represented by scripts. Many texts do not contain the sequences of actions that scripts describe. Section 4.6 described aspects of the Terrorism corpus which made word learning difficult. The biggest problem was that the sentences were so complex that they were seldom completely parsed. Thus Camille was often forced to deal with missing or incorrect information about the examples. As will be further discussed below, Camille would benefit from a parsing system like that described in [Huyck, 1993] that heuristically combines the constituents of a sentence. This would increase the probability that the information given to Camille would be correct and complete. The features of the learning environment that relate to cognitive modeling will be discussed in the next section.

7.2 Cognitive modeling

In Chapter 6, the issue of whether or not to consider Camille a cognitive model was addressed. Camille performs a task that humans must perform, so, at some level, it must be a cognitive model. The question is, does it model human behavior well enough to be of predictive value? At a high level, it has already predicted a trend in psychological theory.

The observations about the differences between the treatment of nouns and verbs were made independently of any knowledge of the related psycholinguistic theories.

Section 4.1 described the implementation of a psychological theory and its application to learning verbs. This course of action is the most promising avenue for development of Camille as a cognitive model. The system can implement aspects of a proposed theory to see how it behaves. Although one could never be sure if an unexpected behavior was due to a flaw in the underlying psychological theory or to a difference between Camille and human learners, it can be used to signal that additional psychological tests should be performed. An example of this type of application will be described in the next section in reference to the Principle of Contrast.

The Mutual Exclusivity constraint was crucial to Camille's ability to add concepts to its knowledge representation. Although the biggest factor in concept acquisition is probably not linguistic — other modes (visual, aural, or even tactile) are likely to be more important — it is likely that linguistic input aids in concept acquisition, either by flagging a potential concept as important or through adding information to or adjusting the meaning of a concept.

The use of hierarchical mental structures in the brain is a topic of much discussion in the psychological literature ([Kaplan *et al.*, 1990] and [Keil, 1991], for example). Oddly, most of the previously developed lexical acquisition systems, even those with cognitive goals, have used rather simplistic knowledge representation structures. Camille's concept hierarchy not only gives it representational economy, but also allows it to make fine distinctions in its lexical inferences. Furthermore, the addition of the connections between concepts which describe the semantic constraints makes the representation structure "matrix-like" as Huttenlocher and Lui say it should be (see section 6.4).

7.3 Future work

The work described in this thesis has created a firm foundation for research in learning word meanings. This section describes ways that Camille can be extended, to enable it to better perform acquisition within the bounds of the information extraction task, and by applying it to new tasks. Camille can also be enhanced as a cognitive model and used to test aspects of psycholinguistic theories.

As mentioned above, Camille would benefit from integration with a heuristic parsing mechanism like Huyck's. Such a mechanism is more likely to create a complete parse, or at least properly combine parse fragments. This should greatly increase Camille's performance in domains with complex corpora.

Although the current version of Camille does include a rudimentary mechanism for recognizing and making use of the morphology of words, it could certainly be improved upon. One major weakness of the current system is that each definition that it makes only applies to the exact word as it was encountered in the text. Obviously, the definitions should apply to all of the forms of a word. This will be a straightforward extension.

The system's ability to handle noise can be improved by having it periodically reexamine its definitions. Because it keeps a record of prior slot fillers, Camille can recognize if one filler is inconsistent with the others. Then it could remove this filler and search for a more appropriate hypothesis.

As mentioned in section 4.2, many of the example texts did not fit well within a script representation because they did not contain a sequence of actions. One way to extend the script mechanism that could be especially productive for texts like those in the Assembly Line domain would be to include preconditions and results of actions, in essence before and after states of

the actions. If a script could link these states instead of requiring strict sequentiality, the connections between events would become more evident.

Another way to extend the script mechanism would be to couple it more tightly with the parsing procedure. Although LINK integrates the application of syntactic, semantic, and pragmatic constraints, the script mechanism, as currently implemented, is not invoked until after LINK is finished. This occasionally forced the script mechanism to reinvoke Camille's learning procedure if it cannot fit a hypothesized meaning into a script. Coupling the script application more tightly with the parser would allow it to impose discourse constraints sooner and rule out incorrect parses *and* word-meaning hypotheses.

Although Camille's development has been oriented toward the information extraction task, there is no reason it could not be used to learn words in more complex task situations. There are many ways that the system's knowledge representation and input mechanism could be extended to allow it to infer deeper meanings for words or to make distinctions that it currently cannot. An increase in the representational power of the language could make the knowledge acquisition bottleneck more severe — or it could increase the need for linguistic acquisition systems.

Knowledge about plans and goals (like that used by Wilensky [1978]) can be used not only to make inferences about information not described in a text, but also about what is described if some important words are not known. Discourse information (e.g. [Sidner, 1979]) could add similar knowledge about texts. Instead of describing only sequences of actions like scripts do, discourse information could specify the communication goals of the text. This could tell the system, for example, what types of background information are likely to be provided about the victims of terrorist attacks.

More low-level world knowledge could be used to disambiguate some of the concepts that take the same set of slot-fillers. Information about the frequency of occurrence of various events, for example that bombings occur more often than hijackings, would allow Camille to prefer more common actions as hypotheses for unknown verbs. Such information could also be made explicit to certain situations. For example, the system could store the knowledge that a certain terrorist organization likes to use a certain type of bomb.

Additional knowledge could be added to the system within the existing framework. Attributes of objects, like their weight, color, etc. could be represented as arcs on the DAGs. With a straightforward extension, action concepts could have constraints on their slot fillers that would restrict the applicable values of these attributes. For example, the Toss concept could specify that its OBJECT should weigh less than 20 pounds. Such additions to the system's knowledge base could become the basis for better word learning. They could also be the foundation for other generalizations like those made by Lebowitz' IPP system [1980], which used correspondences between situations (also taken from terrorist reports) to infer additional domain knowledge from context.

It would be very interesting to integrate Camille with a vision processor that could provide some sort of natural non-linguistic interpretation of the outside world. It may be some time, however, before the state-of-the-art in vision processing allows a reliable depiction of a scene. In the meantime, Camille could follow Moran, Child, and Davra and use some sort of provided description of a scene. This could greatly expand the information that Camille has to use as leverage for its learning, and thereby greatly increase what it can learn, especially in the area of concept acquisition.

Camille can also be extended in its role as a cognitive model. Several important questions remain about the use of the Mutual Exclusivity constraint. Under what circumstances is the constraint overridden? How long is it used? Is there a difference in the application of

the constraint between verbs and nouns?

Camille's use of Mutual Exclusivity could also be tested on some task used in psycholinguistic tests. This would reveal more about Camille's implementation and about its simulation of cognition in general.

As mentioned in section 4.1, Clark proposed a reinterpretation of Mutual Exclusivity [Clark, 1987] that separated it into three separate principles, some of which are abandoned after the initial learning period, and some of which are maintained. Camille's implementation can be extended to make the same distinctions. This will allow the testing of Clark's claims about the transient nature of part of this mechanism and the permanence of the rest of it.

Finally, Camille can be used to test the role of input and semantic structure in learning. As described in section 6.4, Fernald and Morikawa [1993] claimed that differences in the relative prevalence of nouns and verbs in maternal text causes differences in the ratios of nouns and verbs that their children use. Huttenlocher et al. [1991] showed correlation between amount of input and lexicon size. Bloom et al. [1980] showed the effect of semantic structure on acquisition of inflections. Tomasello [1992] described its effect on learning verb argument structures which, he suggests, serve as the groundwork for grammatical structure. Because Camille is an artificial system, its input and semantic structure can be precisely controlled. With more integration of the noun, verb, and concept learning mechanisms, Camille can be used to examine these aspects of children's language acquisition.

APPENDIX A

A DECONSTRUCTION OF THE KNOWLEDGE REPRESENTATION

This appendix contains some philisophical musings about knowledge representation that do not fit easily into the flow of the rest of the thesis but have a pervasive underlying effect on the work reported here, both in the type of representation used by the system and in what it learns. It also contains a set of specific conclusions about how knowledge is represented drawn from the experience of implementing Camille.

What is knowledge? Webster describes knowledge as "... facts or ideas acquired by study, investigation, observation, or experience." Human knowledge, then, encompasses an incredible range of complexity, from high-level concepts like Knowledge to low-level experiences like the feelings we get from seeing certain colors, or smelling certain aromas. Artificial intelligence has not even approached an implementation that can accommodate this range.[1] Even an incredibly ambitious project like Cyc [Lenat, 1990], which is aimed at encoding a massive amount of common-sense knowledge, draws the line at some level of detail, relying on *atoms* to bridge the gap between perception and basic concepts.

What is knowledge representation? Knowledge representation is a mechanism that computer scientists use to create a boundary around a task, creating an abstract version of a problem and of the information required to solve it. This appendix is an attempt to make clear some of the abstractions that this line of research makes and how they influence the learning process.

Where should the knowledge representation lines be drawn? Katz and Fodor [1963] and Barwise and Etchemendy [1989] approach this question from a theoretical perspective. Their conclusions about what a semantic model should include are based on considerations of what what it takes, *in theory*, to represent KNOWLEDGE. Allen [1981] made his task somewhat simpler by taking a more pragmatic approach. He considered what it would take to represent the verb " hide", and answer reasonable questions about it. Allen's answer to his question involved a temporal logic that could address "notions of belief, intention, and causality."[Allen, 1981, p. 81]

This work takes the same approach as Allen's, in effect rephrasing the previous question as, "In order to meet the functional requirements of the overall task, what does the NLP system need to know?" Instead of requiring the system to answer all possible questions about the consequences of actions, however, the system is only required to answer a fixed set of questions about a fixed set of actions — in short, the information extraction task.

[1] The model proposed by Kaplan, Weaver, and French [1990], however, does suggest a promising research direction.

What is required to successfully perform the information extraction task? In the MUC competitions [Sundheim, 1992], systems with greatly varying depths of knowledge representation performed at similar levels of efficiency. Some rather successful systems turned assumptions about the knowledge required for NLP on their ear by dispensing entirely with lexica and grammars and concept representations. Instead they reduced the task to a simple pattern-matching problem (for example, SRI's FASTUS system [Hobbs *et al.*, 1992]). When simple patterns were matched in the input, the appropriate part of the text was extracted.

Unfortunately, because these systems do not have full grammar, it is fairly easy to come up with examples that the pattern-matchers cannot handle. For example, consider these patterns and phrases:

`<HumanTarget> was injured`	"the reputation of the President was injured"
`<HumanTarget's> body`	"the attorney's body of evidence"

The philosophy behind the LINK system is that it is necessary to encode the grammar of the language and the concepts in the domain in order to adequately understand text. (Of course, this implies a need for a larger knowledge base, and therefore, a need for systems like Camille.)

This leads to three basic implications for knowledge representation as it relates to information extraction and lexical acquisition:

- **Level of atomicity:** There is a natural trade-off between the granularity of representation and the amount of knowledge required, for example, Pick-Up versus [Move-Hand, Tighten-Grip, Move-Hand]. LINK makes its atoms at the level of the basic actions and objects that are required by the task (and Camille is pledged to use this granularity). This affects learning in two ways. First, the system is unable to reason about parts of actions or features of objects. Second, the meanings that it learns are at the same level of granularity as the rest of the domain knowledge.

- **Compositionality:** Although LINK/Camille will not break atomic concepts down to a finer granularity, the system can (and routinely does) combine concepts to create more complex concepts (for example, the meaning of a sentence). The process of combining concepts is critical to Camille's learning task (and to language in general). The word-learning task can be viewed as inferring the missing component of a complex concept.[2]

- **Inheritance:** Conceptual knowledge in LINK forms a standard subsumption hierarchy. This allows parsimonious representation of constraints — they are connected with the concept at the highest level of abstraction to which they apply, and then are inherited, or made more specific, by the descendants. This is important for Camille because the structure that results from organizing the concepts forms the space that is searched for meanings of unknown words.

- **Concepts vs. Features:** Within the knowledge representation, there is a choice of methods for representing "attributes" of the concepts. Attributes are defined as features

[2] This brings up an interesting research issue: what would it take for a system to automatically learn word meanings that correspond to complex concepts, for example, sequences of actions? (This is a task that Huffman's system [Huffman *et al.*, 1993] is trained to do.) This will be left to future research.

of the concept which do not affect their set membership. For example, the color of a car may be important, but it doesn't affect whether or not it's labelled as a car. Attributes could be represented by atomic labels, as many syntactic features are, for example, (SYN VTYPE) = TRANS. Alternatively, since there is no limit on the number of parents a concept node can have, a separate concept could be created which subsumes the concepts which have that attribute, for example, Red-Things. In keeping with Camille's overall approach to language, attributes are not represented unless they are relevant to the task. This judgment has implications for the choice of representation. If the attribute is relevant, than it is likely that the system would benefit from explicitly knowing its members. For example, in the Assembly Line domain, Wiring-Harness and Drain-Hose are functionally dissimilar, but share an *important* attribute, that they are long and skinny and flexible. Hence, similar actions can be applied to them, like Uncoil and Route. Thus, the relevancy test leads to the choice of representing attributes as concepts instead of as features.

The importance of making these underlying knowledge representation issues explicit is that by doing so, we can better understand the functional performance limitations of the system. Then, if some additional behavior is desired that lay outside the boundaries, it will be obvious how the system and the underlying knowledge representation must be changed. For example, in order to make inferences about sub-atomic actions like Tighten-Grip, the level of atomicity of the knowledge representation must be changed.

APPENDIX B

TEST RESULTS

This appendix describes the details of the various tests that were applied to Camille and the results of those tests. Most of the analyses of the results are within the body of the thesis.

The appendix begins with a description of the original domain to which Camille was applied, the Assembly Line domain. The task for this domain and the knowledge representation are described. Then the protocol and results of the basic and extended tests are described. The third section compares the performance of Camille over its evolution. The fourth section describes its performance when the percentage of undefined words was varied. The tests on the Camille variations are described in section B.5.

B.1 The Assembly Line domain

Camille was originally applied to the Assembly Line domain. The MUC4 Terrorism domain, described within the body of the thesis, served to validate its results. In this section, the domain knowledge and corpus format of the Assembly Line domain is described. This domain contains descriptions of tasks for a human operator on an automobile assembly line.

This task is somewhat different from other information extraction tasks. Instead of populating a database from the text, the information from the sentences was used to create a set of dependencies in the form of enabling conditions and resulting effects. By combining all of the actions required of each single operator, a model of the entire assembly line was created. This model allowed plant designers to verify a design without actually setting it up.

Figure B.1 displays the action concepts included in this domain, and figures B.2 and B.3 display the objects. In this domain, the constraint definitions include the use of "=r" in addition to "=". The "=r" (for required) operator allowed Camille to distinguish those slots that were required for a particular concept from those that were optional (which is the default). For all versions of Camille after 1.1, the set of hypotheses was checked after sentence parsing, and the concepts whose required slots were not filled were removed from the set.[1]

As previously mentioned, the sentences in this domain consist of short descriptions of actions that the operator should perform, as shown in the examples below:

```
At bench, get inspection record.
Tear off inspection record.
Fold insp record.
```

[1] The "=r" constraint is also used in the Terrorism domain, but because of the complexity of the sentences, the parser rarely completely parses an entire sentence, so the constraint is seldom applied.

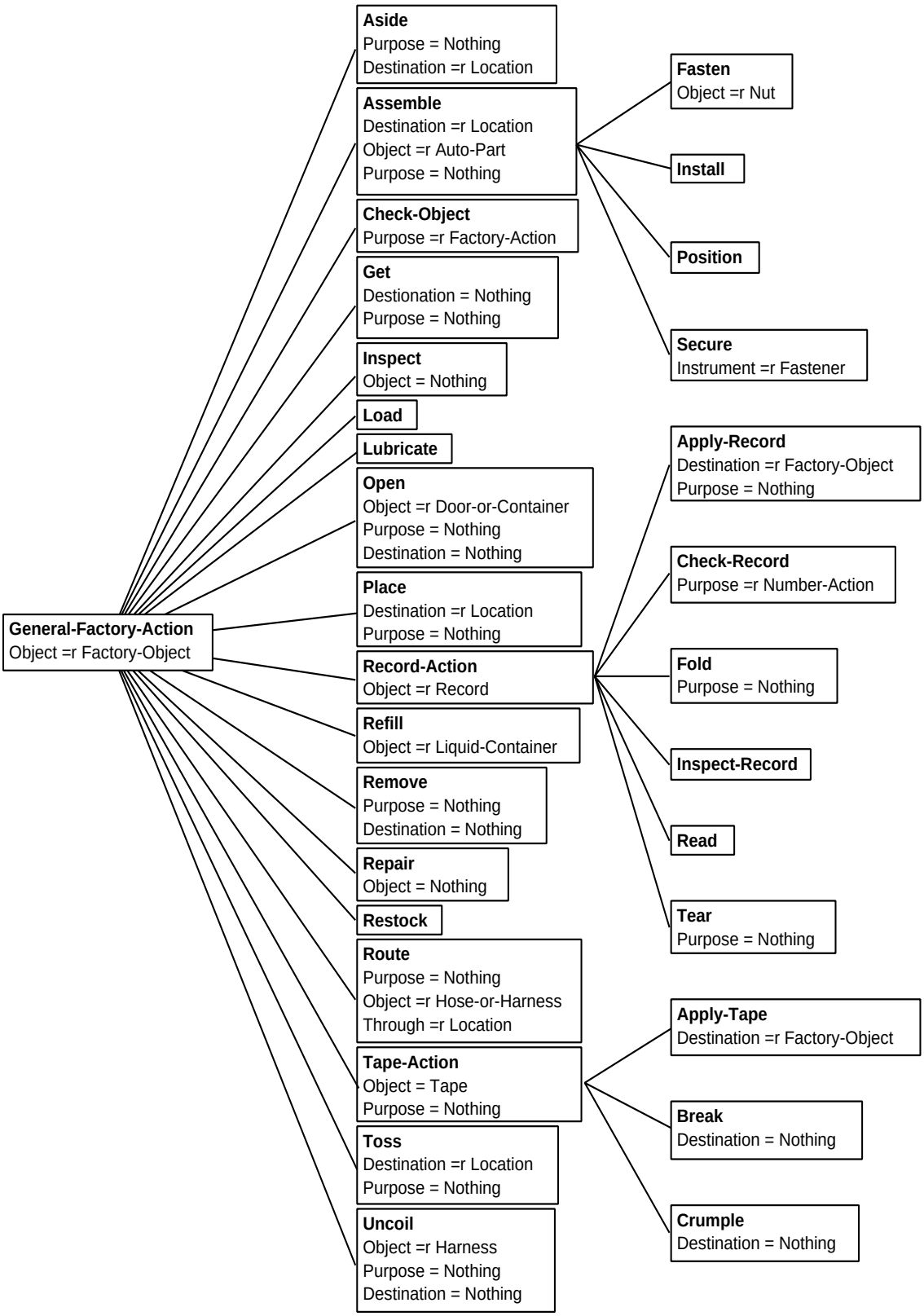


Figure B.1: The action hierarchy for the Assembly Line domain

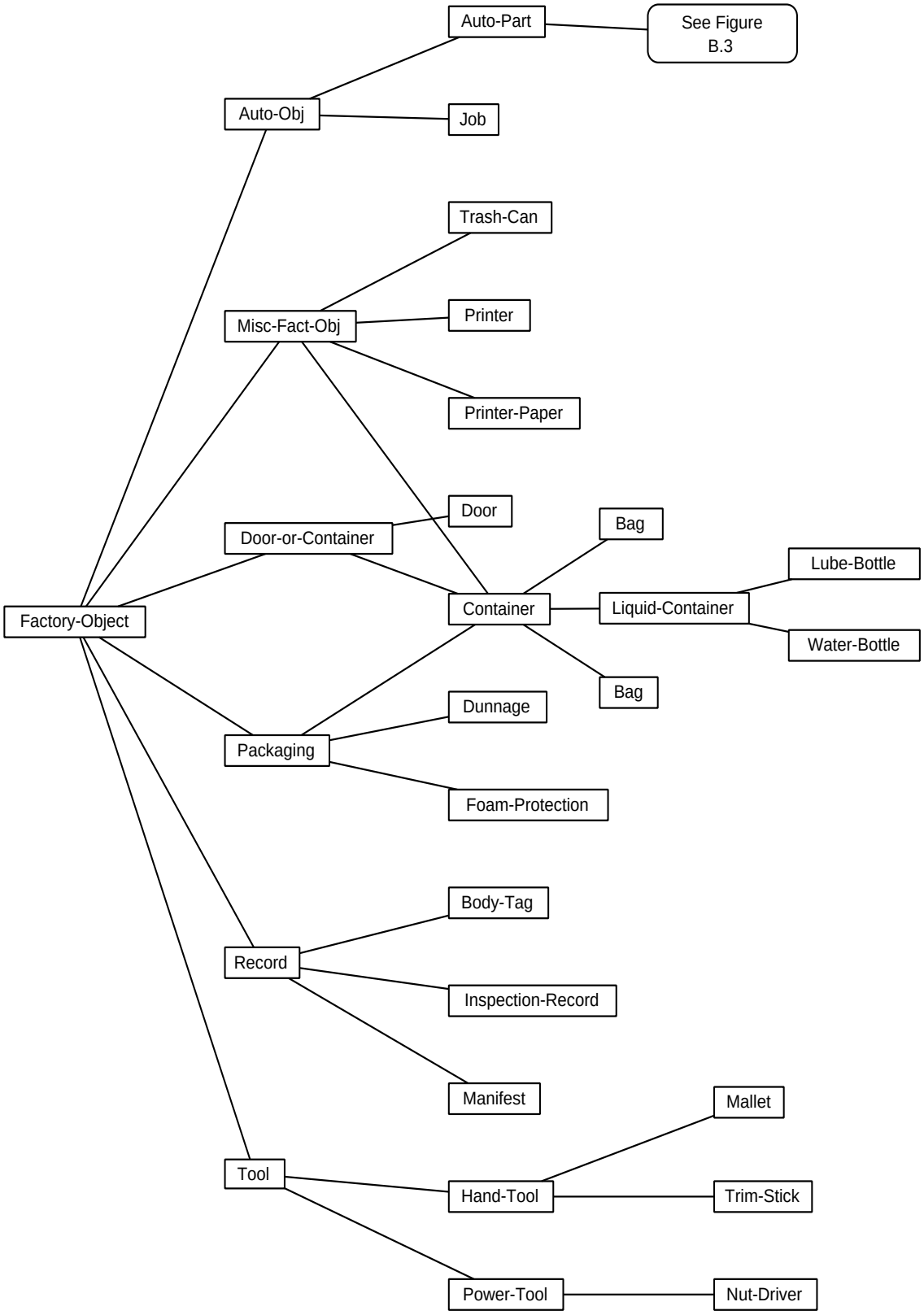


Figure B.2: The top of the object hierarchy for the Assembly Line domain

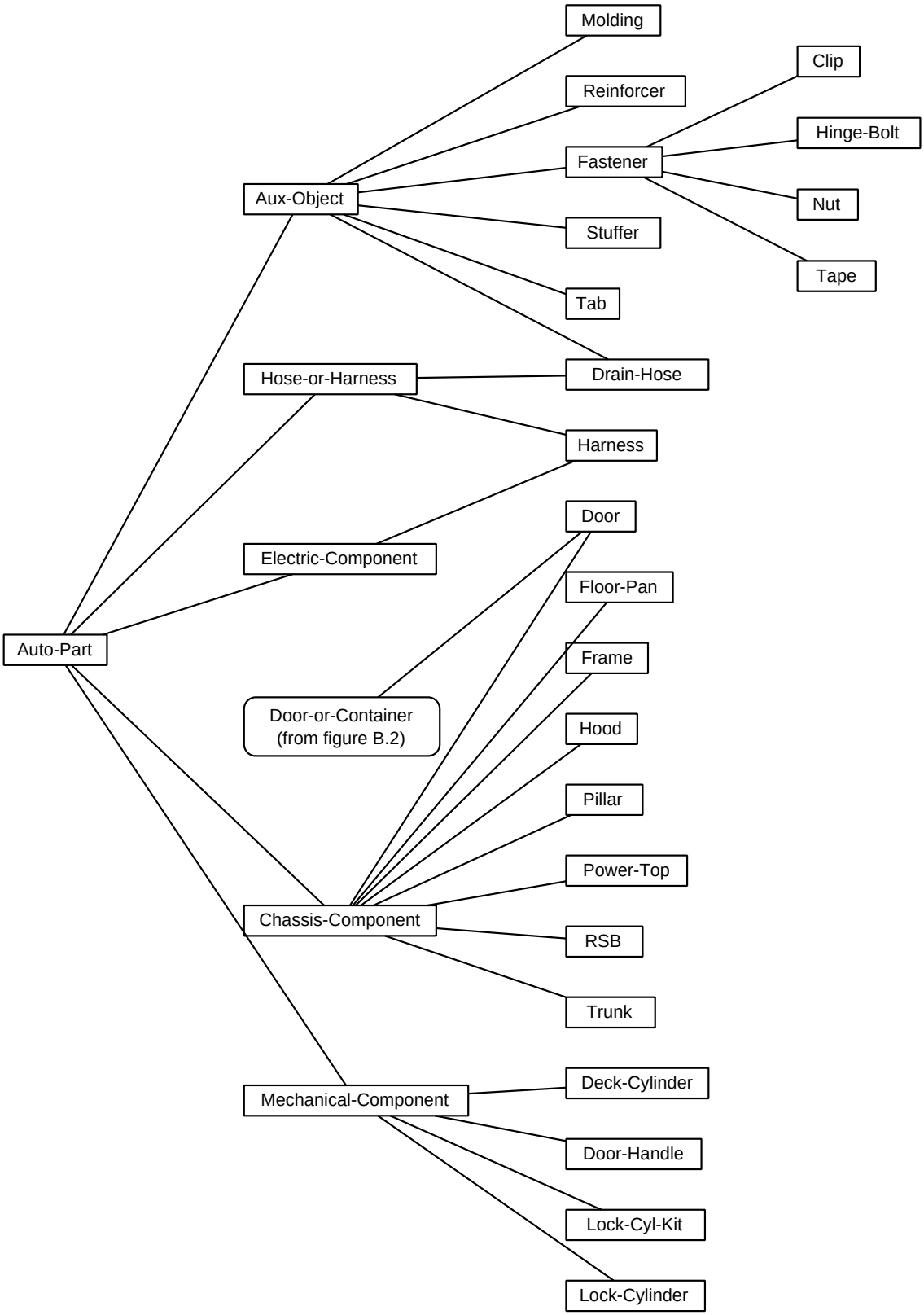


Figure B.3: Descendants of Auto-Part in the Assembly Line object hierarchy

```
At bench, get manifest.
Apply tape to manifest.
Visually check inspection record , manifest to match numbers.
Check manifest for Z7Q, 2DR.
Get lock cylinder kit.
Walk to front of job.
Get and read body tag to verify serial numbers.
Apply manifest to hood.
Walk to front door.
Toss insp record in job.
Walk to bench.
Get driver.
Walk to job.
Open door.
Toss left side lock cylinder to left side floor pan.
Allow to open lock cylinder bags (6 @ a time).
Route harness down right side floor pan through right side bolster.
```

B.2 The Basic Test

For each test a set of sentences was chosen from the corpus at random. In the Assembly Line test, 100 sentences were processed. For the Terrorism domain, 50 sentences were tested because the sentences were longer and had a larger number of verb occurrences per sentence (and parsing times were much longer). These sentences were randomly selected from the subset of sentences in the corpus which contained relevant verbs. A portion of the Terrorism domain test sentences is presented in section B.2.2. In order to simulate full-scale verb learning, all of the verb definitions were removed from LINK's lexicon. Then the sentences were processed by Camille and the resulting definitions were written to the lexicon. After all of the sentences were processed, the definitions inferred by Camille were compared to the correct definitions. This section details the results of each test.

B.2.1 Assembly Line domain

Table B.1 shows the concepts that Camille 1.0 inferred for the verbs in the Assembly Line test sentences. Each unknown word is shown with the set of concepts that Camille positted as its reference.

In table B.2, the verbs are grouped according to the result achieved. The 18 verbs in group 1 (82% of the 22 words for which a hypothesis was inferred) were assigned a correct meaning hypothesis by Camille. This measure is labeled Accuracy in this thesis. Each of the concepts in the hypotheses was consistent with the evidence provided by the example sentences, and the correct concept was in the set.

Group 2 contains verbs that were ambiguous. These verbs referred to two or more nodes in the semantic hierarchy. As described in section 4.3, the initial algorithm had no way of successfully handling such words. The verb "check" is also ambiguous in this domain, but Camille inferred the more general of the two possible concepts for it.

The verbs in group 3 were the victims of shortcomings in the implementation. "Allow" always occurs with a sentential object, e.g. "Allow to load paper to printers." This caused

Table B.1: Test results, Camille 1.0, Assembly Line domain

Verb	**Ordered meaning hypotheses**
allow	Apply-Tape Secure Install Position Step Walk
apply	Aside Check-Object Inspect Load Lubricate Open Place Repair Restock Route Toss
aside	Aside Check-Object Inspect Load Lubricate Open Place Repair Restock Route Toss
break	Apply-Tape Break Crumple Install Position Secure
check	Aside Check-Object Get Inspect Load Lubricate Open Place Remove Repair Restock Route Toss
crumple	Break Crumple
fasten	Fasten
fold	Apply-Record Check-Record Fold Inspect-Record Read Tear
get	Aside Check-Object Get Inspect Load Lubricate Open Place Remove Repair Restock Route Toss
install	Install Position
place	Install Position Secure
position	Install Position Secure
preload	Secure
reach	Reach
remove	Aside Check-Object Get Inspect Load Lubricate Open Place Remove Repair Restock Route Toss
return	Secure Install Position Secure Step Walk
route	Aside Check-Object Get Inspect Load Lubricate Open Place Remove Repair Restock Route Toss
secure	Install Position Secure
step	Secure Install Position Secure Step Walk
toss	Aside Check-Object Inspect Load Lubricate Open Place Repair Restock Route Toss
uncoil	Uncoil
walk	Secure Install Position Secure Step Walk

Table B.2: Grouping of verbs in test results

Group 1	aside, break, check, crumple, fasten, fold, get, install, position, reach, remove, return, route, secure, step, toss, uncoil, walk
Group 2	apply, place
Group 3	allow, preload

difficulty for Camille 1.0 because it could only handle one word at a time (notice that "load" does not appear in the results).

The word "preload" was only found in one sentence in this test set, so Camille's hypothesis was overly specific.

The results of this initial test suggested that a large portion of the meaning of unknown words could be inferred automatically using only very basic conceptual information

about the domain.

B.2.2 The Terrorism domain

A portion of the test sentence from the Terrorism domain is shown below (the actual texts were in all upper case letters):

```
The technical investigation commission has determined that some
military were reportedly involved in the assassination of the six
Jesuits and their two maids, which took place at daybreak on 16
November, as reported by President Alfredo Cristiani on 7 January.

Lopez Albujar, who left his post at the Ministry in May 1989, was
riddled with bullets as he was getting out of his car in the Lima
residential district of San Isidro.

Some 1600 Peruvians were murdered during the last quarter of 1989 due
to the political violence surrounding the 12 November municipal
elections.

Salvadoran Social Democratic politician Hector Oqueli Colindres was
kidnapped today in Guatemala City, his party reported in Mexico City.

The MNR reported on 12 January that heavily armed men in civilian
clothes had intercepted a vehicle with Oqueli and Flores enroute for
La Aurora airport and that the two political leaders had been
kidnapped and were reported missing.

Reportedly, Oqueli had been threatened with death by several people
who, through a government radio network, had accused him of being an
accomplice of the rebels.

Shots were fired from it at the sentry post from a distance of some
150 meters.

Today two people were wounded when a bomb exploded in San Juan
Bautista municipality.

They destroyed several power poles on 29th Street and machinegunned
several transformers.

Ordonez Reyes accused Jose Jesus Pena of masterminding the 7 January
assassination of Contra Commander Manuel Antonio Rugama.
```

After processing these and 40 more sentences from the domain, Camille 1.0 inferred the hypotheses in table B.3.

Table B.4 contains the grouping of the results in this domain. In this test, 15 verbs were assigned hypotheses by Camille, and 8, or 53%, were correct.

Table B.3: Test results, Camille 1.0, Terrorism domain

Verb	**Ordered meaning hypotheses**
accused	Accuse Ambush Bombing Injure Shoot Suspect
attacked	Bombing
claimed	Fight Threat
denied	Admit Report Request Think
destroyed	Bombing Destroy
dynamited	Bombing Destroy
kidnapped	Accuse Ambush Bombing Injure Kidnapping Murder Shoot Suspect
killed	Accuse Suspect
machinegunned	Bombing Destroy
murdered	Accuse Ambush Bombing Injure Kidnapping Murder Shoot Suspect
reported	Bombing
riddled	Shoot
stated	Admit Report Request Think
threatened	Accuse Ambush Bombing Injure Kidnapping Murder Shoot Suspect
wounded	Bombing

Table B.4: Grouping of terrorism verbs in test results

Group 1	accused, denied, destroyed, dynamited, kidnapped, murdered, riddled, stated
Group 2	killed, machinegunned, threatened, wounded
Group 3	attacked, claimed, reported
Group 4	died, exploded

The words in group 1 are those correctly inferred by Camille. The words in group 2 were victims of incomplete parses. Because the sentences were too complex for the grammar, these words were either assigned incorrect slot fillers or were missing slot fillers. The words in group 3 have rather general constraints, so Camille did not have enough instances of their use to converge on a correct hypothesis. Because of incomplete parses, no hypotheses were generated for the words in group 4.

Camille's overall performance was significantly lower for the Terrorism domain than for the Assembly Line domain. This is hardly surprising given the additional complexity of both the concept representation and the sentences in the corpus. The average number of words per sentence gives a rough measure of the corpus complexity. For the Assembly Line domain, it was just over 4 words per sentence, for the Terrorism test set, about 23 words per sentence.

B.2.3 Additional statistical analysis

The initial results of the test were encouraging, but not entirely satisfying. For one thing, the number of guesses for each word was quite high (about 6.2 concepts per word for the Assembly Line domain, 3.2 for Terrorism). In the extreme case, Camille could have guessed that every word referred to each concept, resulting in an Accuracy score of 100%. Ideally, the results should express both the accuracy of the hypotheses and their precision. In order to more adequately reflect this, the scoring mechanism used in recent MUC competitions (for which, it had been adapted from the Information Retrieval world) was adapted to the lexical

acquisition task.

This system of scoring describes performance from several viewpoints. First is a variation of the simple performance measure shown above, labelled Recall. This is formally defined as the number of correct answers divided by the total number of unknown verbs that appeared in the test set. This differs slightly from the Accuracy measure mentioned above. There were some cases in which Camille did not make a hypothesis for word that appeared in the test set due to inadequacies in the Camille implementation or difficulties with the parse. The Accuracy measure provides the ratio of correct answers to the number of hypotheses generated, that is, the number of verbs for which Camille actually produced a possible meaning.

The next measure, Precision, is defined as the number of correct hypotheses divided by the total number of concepts generated. Combining this measure with Recall reflects the desired trade-off between the number of good hypotheses and the total number of hypotheses generated.

Especially in domains with complex sentences, LINK frequently returned incomplete parses. Although parse fragments were extracted from the chart, they did not necessarily contain a given unknown word from the sentence. Without this information, Camille could not infer a meaning for the word. In order to measure the system's performance based on the words for which it could make hypotheses, the Accuracy measure is included in the following graphs.

In order to gauge the most basic inferential capabilities of the system, the ratio of the number of verbs in the test set to the number for which Camille generated a hypothesis was calculated. This is represented by the Production measure.

One final measure shows how many definitions Camille got exactly right. The Parsimony measure is the ratio of the correct answers which only contained one concept to the total number of possible definitions.

As displayed in figure B.4, Camille 1.0 running on the Assembly Line test scored a Recall of 51%, a Precision of 13%, and Production of 63%, Accuracy of 82%, and Parsimony of 9%. In the Terrorism domain (figure B.5), Camille scored a Recall of 47%, a Precision of 15%, Accuracy of 53%, Production of 88%, and Parsimony of 6%. These scores become more interesting when put into the context of the changing performance of the system over time as described in the section B.3.

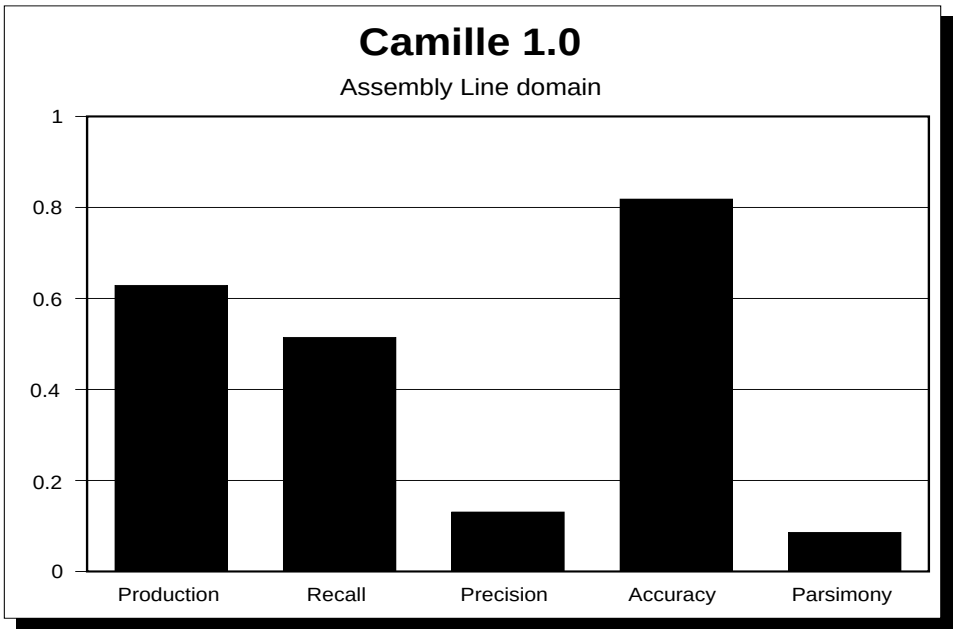


Figure B.4: Camille performance, Assembly Line domain

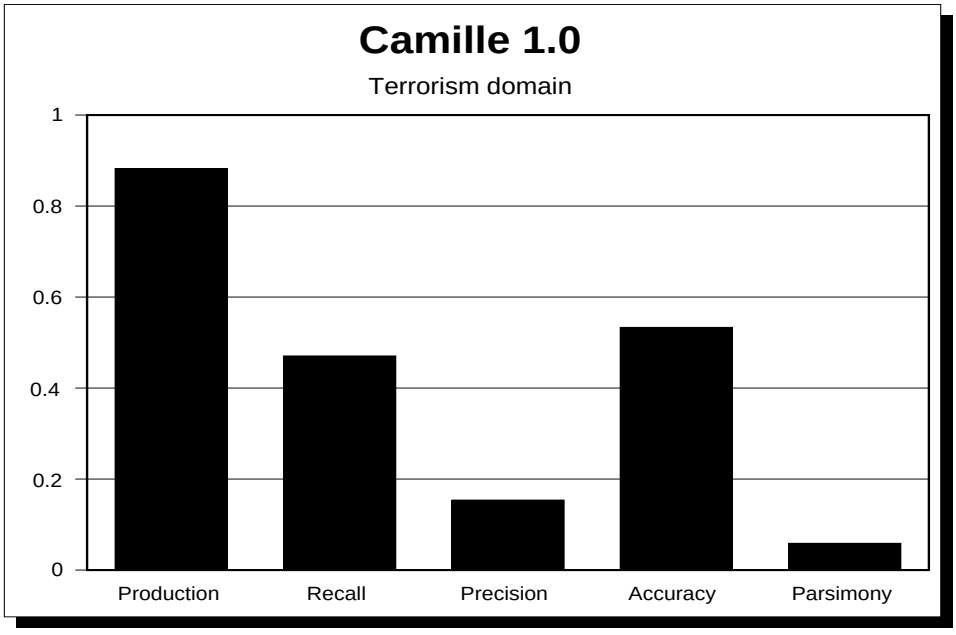


Figure B.5: Camille performance, Terrorism domain

B.2.4 Target word repetitions

As described in section 3.2.4, the number of instances of a word is important for Camille's performance. Table B.5 shows the number of repetitions of the words in the Assembly Line test set. Table B.6 shows the corresponding summary for the Terrorism domain. The average number of occurrences for the Assembly Line domain was 3.7, and for Terrorism it was 2.7.

Table B.5: Repetitions of words in the Assembly Line test set

allow	10	preload	1
apply	2	reach	1
aside	9	read	1
break	1	refill	1
check	3	remove	6
crumple	1	repair	2
fasten	1	return	6
fold	1	route	2
get	22	secure	3
inspect	2	step	2
install	5	stock	4
load	1	tear	1
lubricate	1	toss	3
match	1	uncoil	1
open	7	verify	1
place	2	walk	16
position	1		

Table B.6: Repetitions of words in the Terrorism test set

accused	2	machinegunned	2
attacked	4	murdered	2
claimed	1	reported	10
denied	2	riddled	1
destroyed	2	said	3
died	2	shot	5
dynamited	1	stated	1
exploded	2	struck	1
kidnap	1	threatened	1
kidnapped	5	told	1
kill	1	wounded	7
killed	4		

If the items are sorted by the number of occurrences, however, as in tables B.7 and B.8, the difficulty of the task is clear. In the Assembly Line domain, 15 of the 33 verbs (or 45%) occur only once in the test set, and in the Terrorism set, 9 of the 23 (39%) words occur only once, and 16 (70%) occur twice or less.

Table B.7: Sorted repetitions of Assembly Line verbs

break	1	place	2
crumple	1	repair	2
fasten	1	route	2
fold	1	step	2
load	1	check	3
lubricate	1	secure	3
match	1	toss	3
position	1	stock	4
preload	1	install	5
reach	1	remove	6
read	1	return	6
refill	1	open	7
tear	1	aside	9
uncoil	1	allow	10
verify	1	walk	16
apply	2	get	22
inspect	2		

Table B.8: Sorted repetitions of Terrorism verbs

claimed	1	died	2
dynamited	1	exploded	2
kidnap	1	machinegunned	2
kill	1	murdered	2
riddled	1	said	3
stated	1	attacked	4
struck	1	killed	4
threatened	1	kidnapped	5
told	1	shot	5
accused	2	wounded	7
denied	2	reported	10
destroyed	2		

B.2.5 The bigger test

As mentioned in Chapter 3, the basic test sets were intentionally kept fairly small to limit the number of repetitions of particular words. The goal was to see how well the system could do under rather extreme conditions. In order to completely test the system, however, it was applied to a larger test set in the Terrorism domain. 200 sentences were selected from the corpus which contained the undefined verbs from the basic test set. Then the sentences were processed and Camille's scores after each set of 50 sentences was calculated. The respective mean and median word repetitions for the increasing test sets were: 3.6 and 2, 6.9 and 5, 10.2 and 6, and 14.2 and 9. As mentioned in section 3.4, Camille's scores actually went down, as shown in figure B.6.

Most of the decreased performance was due to errors in the parsed sentence structure. To isolate the performance of the learning mechanism from the performance of the parsing

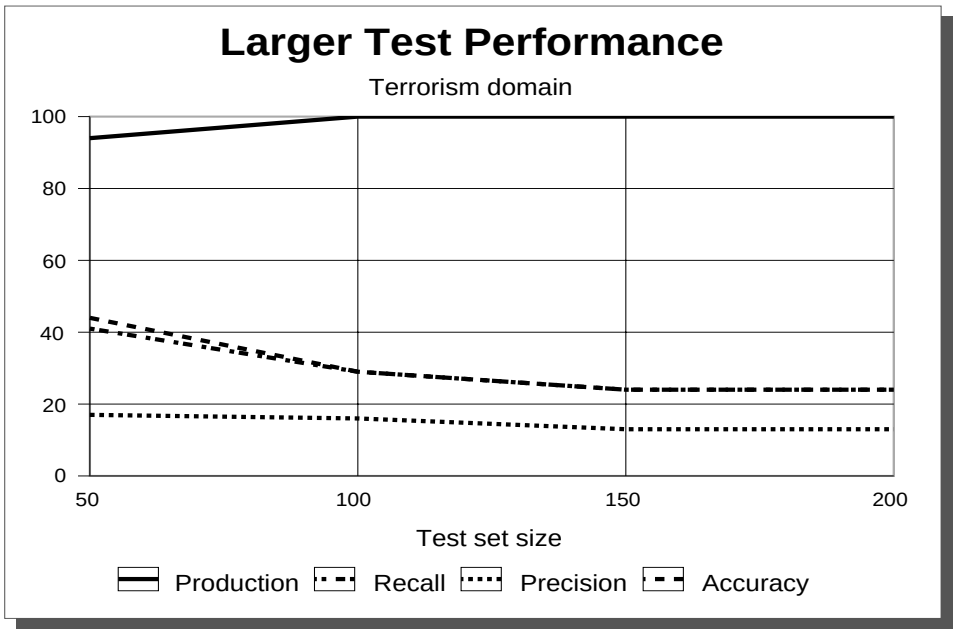


Figure B.6: Camille performance with larger test set

mechanism, this set of sentences was "hand-parsed", producing the actual slot fillers that the a correct parse would yield. Examples of three sentences and their representations are shown below:

```
Those accused of the assassination of six Jesuits will have a fair
trial and if found guilty, will be punished whether they are
civilians, military, or influential people, Supreme Court
President Dr. Mauricio Gutierrez Castro said.

(SAID (ACTOR POLITICAL-FIGURE) (OBJECT IGNORE-ACTION))

Salvadoran Social Democratic politician Hector Oqueli Colindres was
kidnapped today in Guatemala City, his party reported in Mexico City.

(KIDNAPPED (OBJECT POLITICAL-FIGURE) (TIME DATE) (LOCATION PLACE))

As a result of these attacks, several persons were wounded and others
died.

(WOUNDED (OBJECT CIVILIAN))
(DIED (ACTOR CIVILIAN))
```

The comparative results of the LINK-parsed and hand-parsed versions are shown in figure B.7.

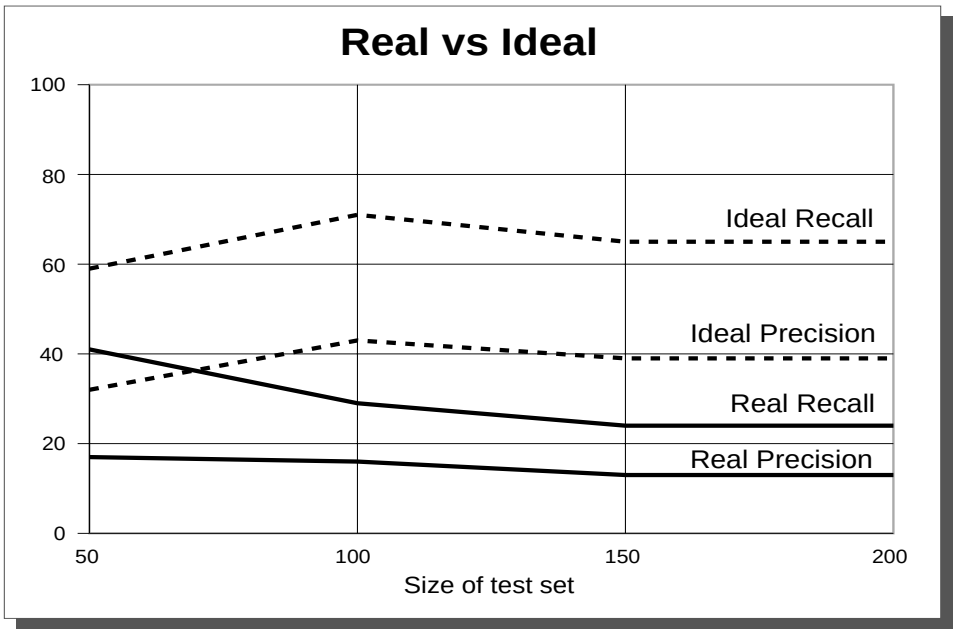


Figure B.7: Comparing LINK-parsed input to hand-parsed input

B.3 Plotting Camille's evolution

In this section, the test results of the different stages in Camille's evolution are presented. Camille 1.0 is the initial version. As described in section 3.3, Camille 1.1 ranked the hypotheses based on the tightness of their constraints fit to the slot fillers. Camille 1.2 (section 3.4) maintained a memory of the slot fillers so that the constraints could be accurately matched to additional instances. Due to the *post hoc* nature of the testing, however, Camille 1.1 and Camille 1.2 could not be tested separately. Therefore, separate test results are not included here for Camille 1.1.

The testing for the variations on the basic Camille implementation, Mutual Exclusivity, scripts, ambiguous words, and concept creation, are described in section B.5.

The enhancements made in versions 1.1 and 1.2 of Camille were intended both to fix its lapses in memory and to reduce the size of the concept sets that it hypothesized. Table B.9 shows the results of testing Camille 1.2 on the Assembly Line domain.

As shown in figure B.8, Camille 1.2 scored a Recall of 71%, Precision 22%, Production 94%, Accuracy 76%, and Parsimony 14% in the Assembly Line domain. This figure also contrasts the performance of the original version with the improved Camille 1.2. Although the Accuracy decreased slightly, the other measures all improved significantly. This reflects, to some extent, the trade-off between Recall and Precision. Camille 1.2 inferred hypotheses for 11 more verbs, but produced fewer concepts (112 as opposed to 138). By reducing the average number of concepts per hypothesis (3.4 as opposed to 6.3), Camille 1.2 eliminated some correct concepts but greatly increased Precision (22 from 13, a 70% increase).

Figure B.9 shows the combined difference in scores with an area graph. The four scores for both versions of Camille are added to show the overall increase in performance.

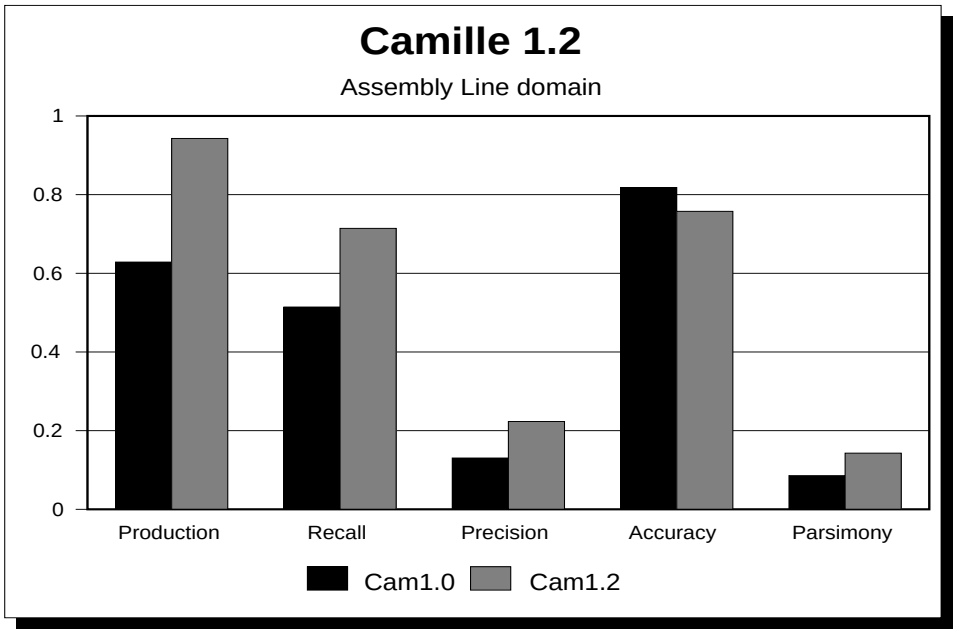


Figure B.8: Camille 1.2 performance, Assembly Line domain

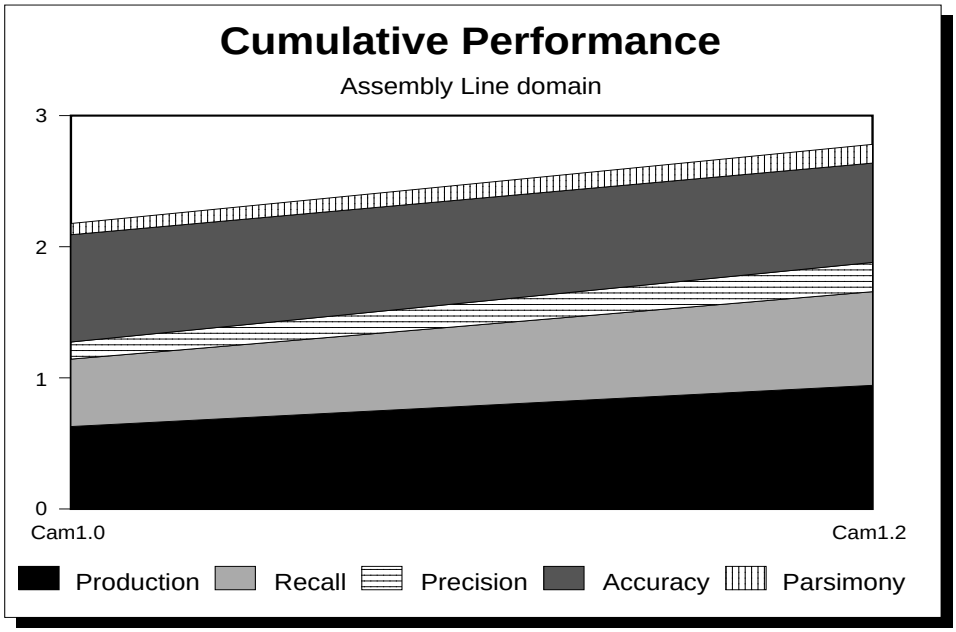


Figure B.9: Cumulative Camille performance, Assembly Line domain (area graph)

Table B.9: Test results, Camille 1.2, Assembly Line domain

Verb	**Meaning hypotheses**
allow	Inspect Repair
apply	Check-Object Load Lubricate Place Restock Toss
aside	Check-Object Load Lubricate Restock
break	Break Crumple
check	Check-Object Finish-Action Load Lubricate Restock
crumple	Break Crumple
fasten	Fasten
fold	Check-Record Fold Inspect-Record Read Tear
get	Check-Object Get Load Lubricate Remove Restock
inspect	Action
install	Prepare-Action
load	Check-Object Get Load Lubricate Remove Restock
lubricate	Check-Object Get Load Lubricate Remove Restock
match	Match Verify
open	Open
place	Install Position
position	Install Position
preload	Check-Object Get Load Lubricate Remove Restock
reach	Reach
read	Apply-Record Check-Record Fold Inspect-Record Read Tear
refill	Refill
remove	Check-Object Get Load Lubricate Remove Restock
repair	Action
return	Step Walk
route	Check-Object Get Load Lubricate Remove Restock Route
secure	Check-Object Load Lubricate Restock
step	Step Walk
stock	Check-Object Get Load Lubricate Remove Restock
tear	Check-Record Fold Inspect-Record Read Tear
toss	Check-Object Load Lubricate Place Restock Toss
uncoil	Uncoil
verify	Match Verify
walk	Step Walk

For the Terrorism domain, the set of hypotheses inferred by Camille 1.2 is displayed in figure B.10.

Table B.10: Test results, Camille 1.2, Terrorism domain

Verb	**Meaning hypotheses**
accused	Ambush Injure Shoot
attacked	Attack
claimed	Fight Threat
denied	Admit Report Request Think
destroyed	Destroy
dynamited	Destroy
kidnapped	Ambush Injure Kidnapping Murder Shoot
killed	Accuse Suspect
machinegunned	Destroy
murdered	Ambush Injure Kidnapping Murder Shoot
reported	Robbery
riddled	Shoot
stated	Admit Report Request Think
threatened	Ambush Injure Kidnapping Murder Shoot
wounded	Robbery

The results of testing Camille 1.2 on the Terrorism domain were: Recall 41%, Precision 19%, Production 88%, Accuracy 47%, and Parsimony 18%. Figure B.10 compares the performance of Camille 1.0 and Camille 1.2 on this test. Here, the Recall / Precision trade-off is much more evident. Camille 1.2 produced almost a third fewer concepts, resulting in an average number of concepts per hypothesis of 2.5 compared to 3.5. As an example of the effect of this reduction, instead of guessing both Destroy and Bombing for the meaning of "destroyed" and "dynamited", Camille guessed only the concept Destroy. Thus Precision was increased, but the number of correct concepts was decreased (by one), accounting for the drop in the Recall and Accuracy scores (and a slightly reduced gain in Precision).

Figure B.11 shows the combined difference in scores with an area graph. True to the nature of the Recall / Precision trade-off, the compound result barely changed from from one version to the other. This does not, of course, imply that the performance of the system didn't change. It demonstrates that by changing the behavior of the system to create narrower or broader concept sets, the system can be tuned to the needs of the application.

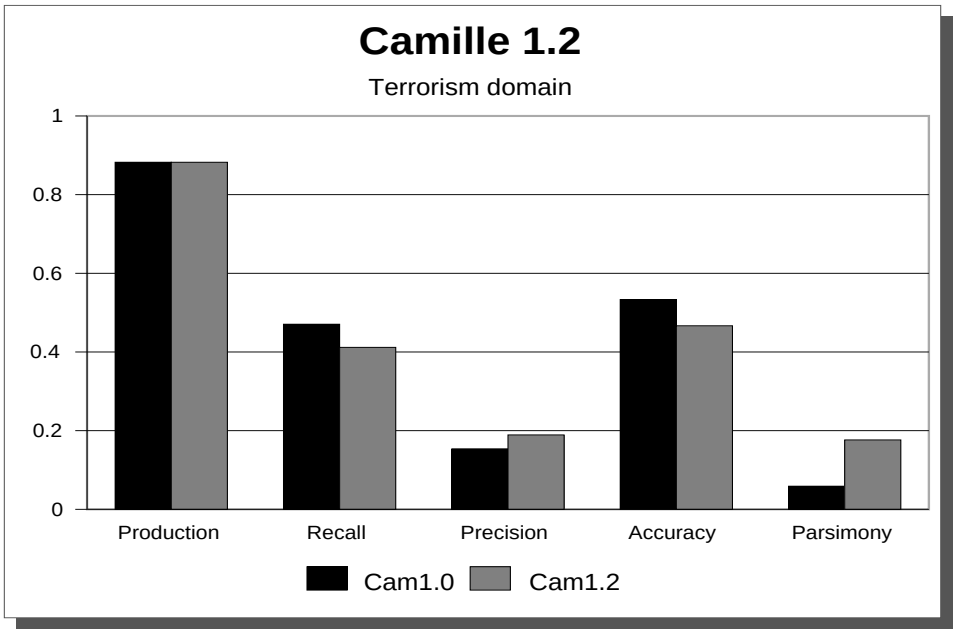


Figure B.10: Cumulative Camille performance, Terrorism domain

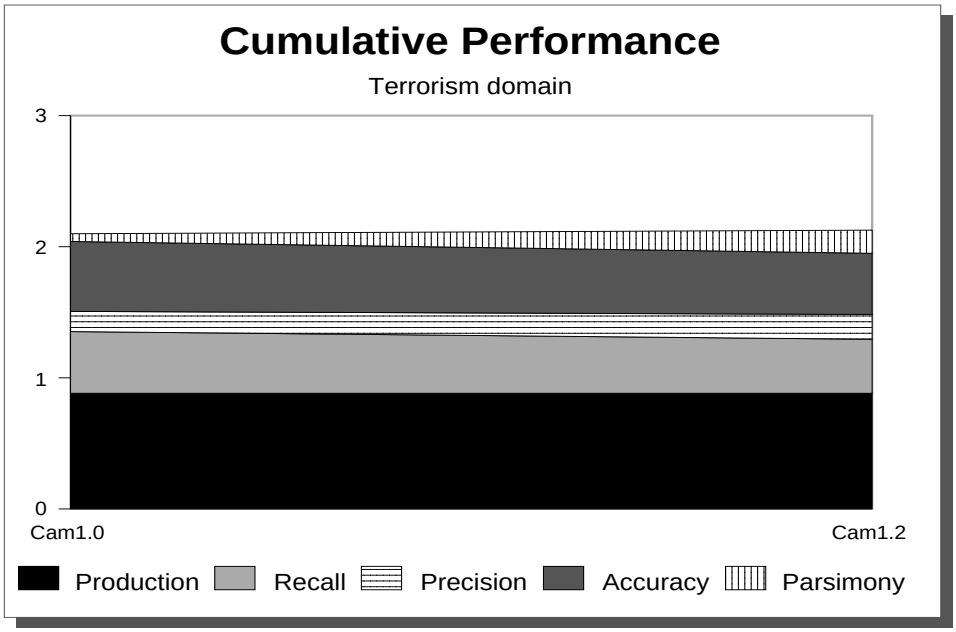


Figure B.11: Cumulative Camille performance, Terrorism domain (area graph)

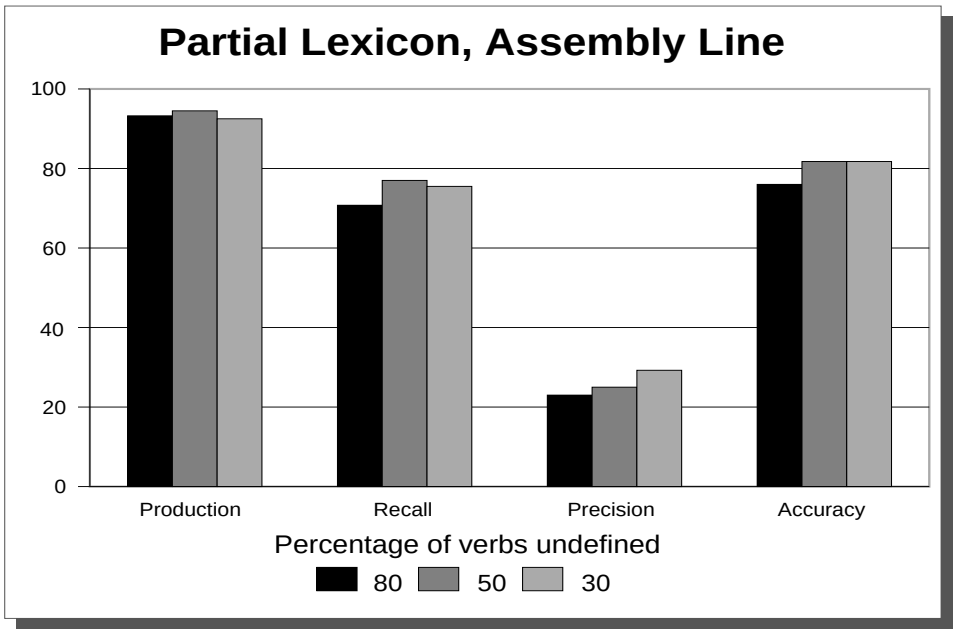


Figure B.12: Partial lexica Camille performance, Assembly Line domain

B.4 Testing Partial Lexica

In order to evaluate Camille's performance with partial lexica, a series of tests was run in the Assembly Line and Terrorism domains, varying the percentage of the original verb definitions that were deleted. Camille's performance was evaluated with 20%, 50% and 70% of the domain's verbs defined. Because the verbs which were defined were chosen randomly, the test at each level was run 4 different times. This enabled an analysis of the system's performance under various stages of development.[2] The results for the Assembly Line domain are shown in figure B.12, and the results for the Terrorism domain in figure B.13.

These graphs display the values of the test measures on the average of the four runs at each of the three levels of lexical ignorance. The only apparent trend from the Assembly Line data is the increase in Precision as fewer verbs are undefined. This stems from the fact that each successive test produced significantly fewer concepts. While the number of correct guesses went from 21 to 14 to 10, the total number of concepts produced went from 91 to 58 to 35. Part of the explanation for this behavior comes from the fact that certain verbs caused Camille to generate a much higher than average number of concepts per word. Although the average was 3.4, the median was 2. Thus, eliminating a one-word hypothesis and a six-word hypothesis from the set produces a downward effect on the average, but not enough to account for the large reduction in concept production. A word-by-word examination of the results reveals that the number of concepts hypothesized for each word did not change. Apparently the random definitions of words coincidentally defined (and thus kept Camille from hypothesizing) more words for which Camille had hypothesized higher numbers of concepts.

The Terrorism results show a downward trend in Recall, Precision, and Accuracy.

[2] The test could not be viewed as a complete test of varying development however, since the grammar and the lexicon for the rest of the words were at the same level of completeness for all of the tests. This was necessary, however, to enable direct comparison of the different versions

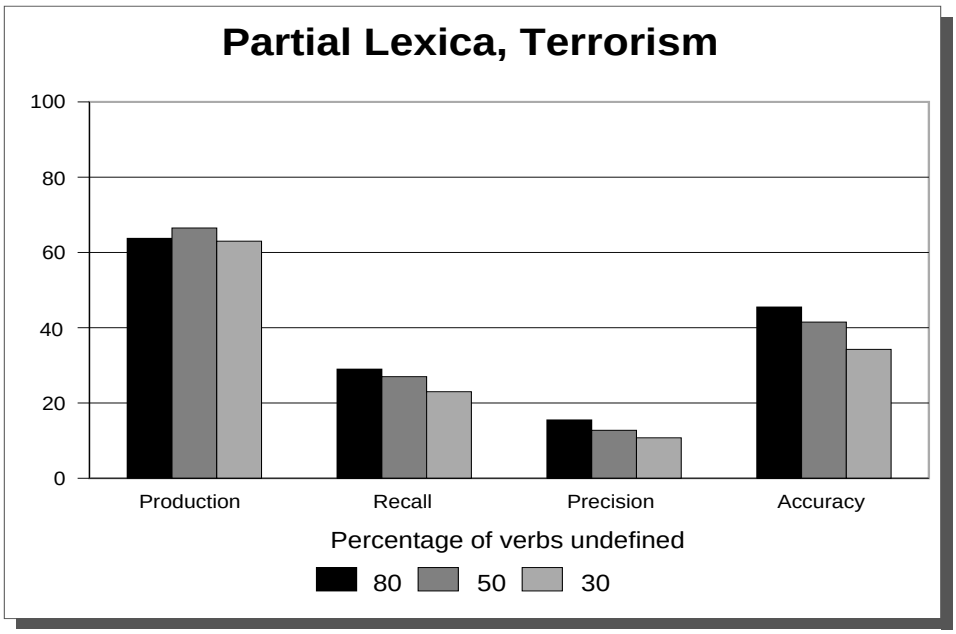


Figure B.13: Partial lexica Camille performance, Terrorism domain

Again this appears to be an artifact of particular testing results. Of the three basic measures, the number of correct guesses, the number of hypotheses generated, and the the total number of concept generated, each decreases accordingly as the number of words that are available for Camille to learn decreases (as the percentage of defined verbs increases). The rate of reduction in the number of correct guesses is higher than the others, and thus produces the downward trend in the calculated measures which depend in it. The noise in the data comes from one of the 70% defined tests. In this test, only 3 words were assigned hypotheses by Camille, compared to the average of 6.7 in the other three tests. Of these three guesses, none contained a correct concept, reducing the average number of correct guesses for these tests to 2.3 from the average of 3 for the other tests. If this test is removed, the overall result is as shown in figure B.14, where no trend is evident.

Two conclusions come from these tests. One is that sometimes small variations in the tests can lead to larger differences in Camille's behavior. This could potentially be addressed by running tests with larger amounts of undefined words and larger numbers of repetitions of those words. The goal of the tests however, was to examine the performance of the system under difficult conditions. A non-interactive agent has no control over the number of learning instances that it gets, and thus must be able to learn with scant data.

The more general conclusion is that the performance of the system is not significantly affected by changes in its lexical knowledge. This is important because the system should not have more difficulty learning when its knowledge of other words increases.

B.5 Testing the variations

This section describes the tests on the variations on the basic Camille system that were described in Chapter 4.

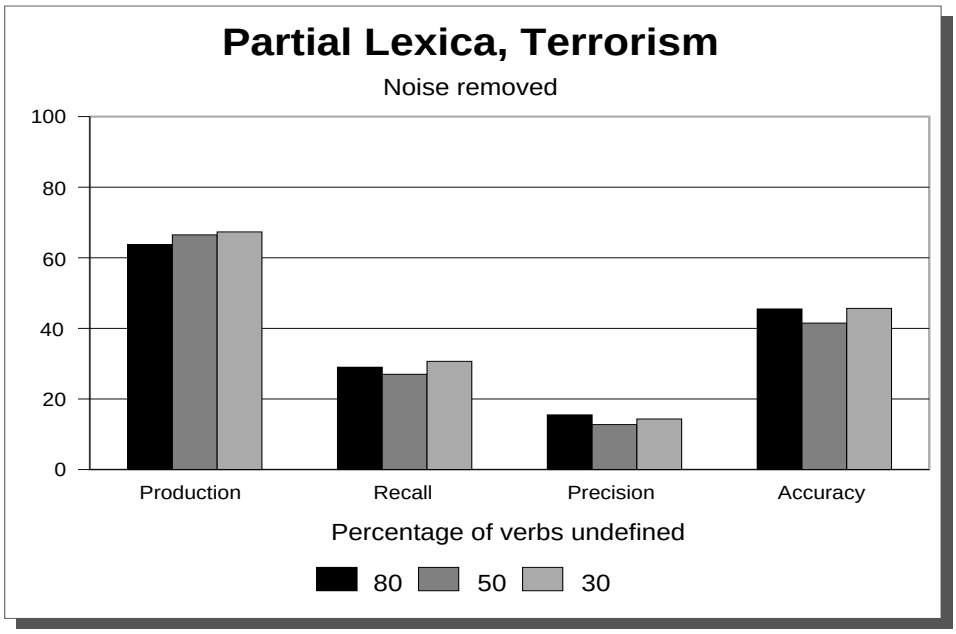


Figure B.14: Revised Partial lexica Camille performance, Terrorism domain

B.5.1 Mutual Exclusivity

The testing paradigm used for evaluating the Mutual Exclusivity implementation was the same as that used for testing the basic system. Camille processed the same test set with the Mutual Exclusivity switch turned on. The switch required Camille to take note of the mappings between concepts and words. Hypotheses linking an unknown word to a concept that was already the referent of another word were rejected. As mentioned in Chapter 4, the results were close to those from testing the basic system.

Table B.11 shows the results from testing in the Assembly Line domain. The only differences between these results and the ones from Camille 1.2 are in the definitions for "preload" and "uncoil". For both of these words, by the time they were processed, another word had already been assigned to their appropriate concept. For "preload", with the sentence, "Preload nut to driver," Camille tried (erroneously) to attach the word to the concept Fasten which was already assigned as the referent of the word "fasten". Camille was unable to find any other applicable referent for "preload" so it was assigned the most general concept, Action.

By the time that Camille processed the sentence, "Uncoil power top harness," it had already assigned the concept Uncoil to the word "route" (from "Route harness down right-side floor pan through right-side bolster."). Ironically, the hypothesis was later changed for "route" because a Hose was attached as its OBJECT.

The scores for the Mutual Exclusivity test on the Assembly Line domain were: Recall 69%, Precision 23%, Production 94%, Accuracy 73%, and Parsimony 14%.

Table B.12 shows the concepts hypothesized using Mutual Exclusivity for the Terrorism domain. The scores for Terrorism test set were 17% Recall, 21% Precision, 62% Production, 79% Accuracy, and 12% Parsimony. As expected, the system had difficulty inferring definitions for this test set because of the high numbers of synonyms in the corpus (see section 4.1 for more discussion). The system did, however, produce relatively precise definitions, generating only 19 concepts for its 15 hypotheses. This was a significant reduction over the basic system which

Table B.11: Test results, Camille 2.0, Assembly Line domain

Verb	**Meaning hypotheses**
allow	Inspect Repair
apply	Aside Check-Object Load Lubricate Place Restock Toss
aside	Check-Object Load Lubricate Restock
break	Break Crumple
check	Check-Object Finish-Action Load Lubricate Restock
crumple	Break Crumple
fasten	Fasten
fold	Check-Record Fold Inspect-Record Read Tear
get	Check-Object Get Load Lubricate Remove Restock
inspect	Action
install	Prepare-Action
load	Check-Object Get Load Lubricate Remove Restock
lubricate	Check-Object Get Load Lubricate Remove Restock
match	Match Verify
open	Open
place	Install Position
position	Install Position
preload	Action
reach	Reach
read	Apply-Record Check-Record Fold Inspect-Record Read Tear
refill	Refill
remove	Check-Object Get Load Lubricate Remove Restock
repair	Action
return	Step Walk
route	Check-Object Get Load Lubricate Remove Restock Route
secure	Secure
step	Step Walk
stock	Check-Object Get Load Lubricate Remove Restock
tear	Check-Record Fold Inspect-Record Read Tear
toss	Aside Check-Object Load Lubricate Place Restock Toss
uncoil	Action
verify	Match Verify
walk	Step Walk

produced twice as many concepts in 16 hypotheses. The fact that fewer of the hypotheses created by Camille 2.0 were correct prevented the system from making a dramatic gain in the Precision score.

B.5.2 Script testing

As mentioned in section 4.2, the testing paradigm for the acquisition mechanism using scripts was fundamentally different from that used for testing the basic system. Instead of presenting the system with a set of unconnected sentences, the test set for the script mechanism consisted of complete texts. For the Assembly Line domain, these were descriptions of the set of

Table B.12: Test results, Camille 2.0, Terrorism domain

Verb	**Meaning hypotheses**
accused	Ambush Injure
attacked	Attack
claimed	Threat
denied	Think
destroyed	Action
dynamited	Action
kidnapped	Kidnapping Murder
killed	Suspect
machinegunned	Action
murdered	Kidnapping Murder
reported	Threat
riddled	Shoot
stated	Think
threatened	Kidnapping Murder
wounded	Terrorist-Act

actions that a single operator would take on the line, for example (with abbreviations expanded for readability):

```
At bench get right-rear door handle reinforcer.
Walk to job.
Simultaneously get front door stuffer from apron.
Install stuffer at front door frame lower rear.
Open door.
Step into opening.
Get harness.
Route harness down right-side floor pan through right-side bolster.
Secure harness to right-side bolster with 2 clips.
...
Return to bench.
Aside tape to trash.
Allow to stock stuffers to apron.
Allow to lubricate stuffers.
Allow to refill lube bottle.
Allow to open boxes and stock water bottle.
Inspect and repair as required.
```

As discussed in section 4.2, the mechanism was tested with both specific scripts and more general ones. For both cases the top-level script for the Assembly Line domain was the same:

```
(define-sem job-script is-a (action-script)
   formulae (((1*) = assemble-script
              (2*) = finish-script)))
```

These scripts allow subscripts and iterations. The Job-Script is interpreted as: "Any number of iterations of Assemble-Script followed by any number of iterations of Finish-Script."

The Finish-Script allowed any Finish-Action: (Allow Lubricate Refill Inspect Repair).

The more specific Assemble-Script was defined with two children as follows (the items in the equations with *'s are previously defined nodes in the concept hierarchy):

```
(define-sem assemble-script-1
   is-a (assemble-script)
   formulae (((1) = *get*
              (2*) = manipulate-script
              (3) = *assemble*
              (4*) = assemble-subscript
              (1 object) = *factory-object*)))

(define-sem assemble-script-2
   is-a (assemble-script)
   formulae (((1) = *get*
              (2*) = record-script
              (1 object) = *record*)))
```

Either of these scripts could be an interpretation of any iteration of Assemble-Script in a Job-Script. Manipulate-Script allowed any Prepare-Action, (Walk Step Reach Uncoil Toss Place Crumple Break Apply-Tape Remove Open Load Check-Object). The Assemble-Subscript allowed any number of repetitions of Assemble actions. Record-Script could be either a Record-Action or a Tape-Action applied to a Record.

Unfortunately, due to the intertwining of these sequences of actions within the texts (as discussed in section 4.2), these scripts did not accurately describe the texts. They were replaced by the following more general script:

```
(define-sem assemble-script-1
   is-a (assemble-script)
   formulae (((1*) = manipulate-script)))
```

The Manipulate-Script referred to in this script allowed any of the larger set of actions which included the Prepare-Actions mentioned above as well as the Record-Actions and the Assembles.

Table B.13 shows the hypotheses generated for the Assembly Line domain using these more general scripts. The scores for the test were: 34% Recall, 18% Precision, 40% Accuracy, 86% Production and 6% Parsimony. Figure B.15 shows the cumulative scores of the system on the Assembly Line domain. The Mutual Exclusivity and Script tests are not directly comparable to the others, but are included for reference.

In the Terrorism domain, the tests were run on 20 uneditted messages from a newswire service. One of the test messages is included below:

> Salvadoran Social Democratic politician Hector Oqueli Colindres was kidnapped today in Guatemala City, his party reported in Mexico City. Oqueli Colindres is the secretary of the National Revolutionary Movement. The MNR is directed by

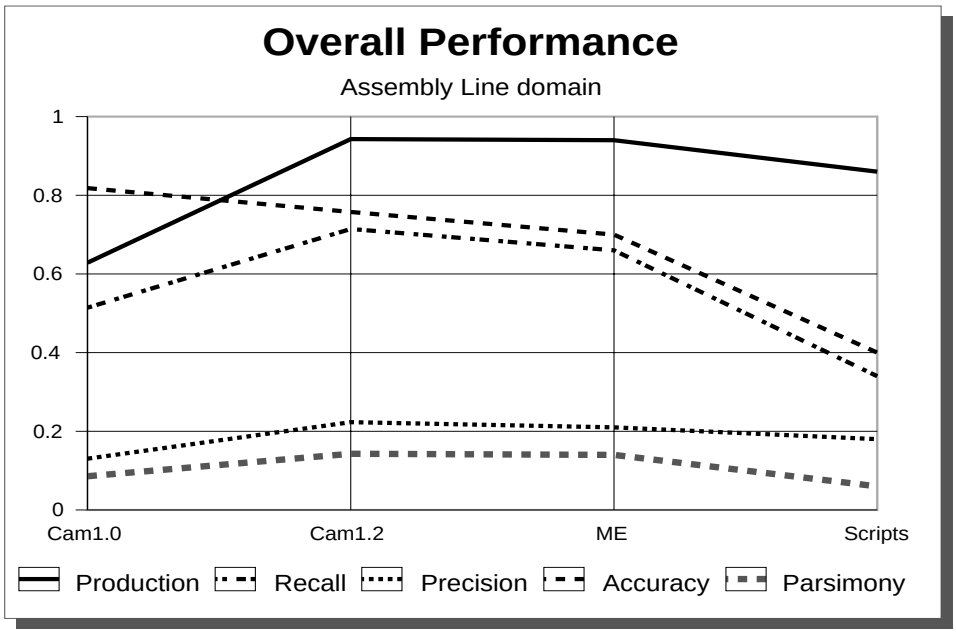


Figure B.15: Overall Camille performance, Assembly Line domain

Guillermo Ungo. Oqueli is also Socialist International secretary for Latin America. In a communique, the MNR said Oqueli had arrived in Guatemala on 11 January and was planning to travel today to Nicaragua as a member of a Socialist International delegation. The communique adds that Oqueli Colindres was kidnapped between 0630 and 0700 by heavily armed men while on his way to the airport along with Guatemalan Social Democratic leader Gilda Flores, who was also kidnapped. Oqueli, who returned last year to El Salvador after a long exile in Mexico, where he represented the Farabundo Marti National Liberation Front and the Revolutionary Democratic Front Political-Diplomatic Commission.

The scripts that were used to test the Terrorism texts were defined as follows:

```
(define-sem ter-act-script is-a (terrorist-action-script)
   formulae (((1) = nasty-action
              (2) = wound
              (3) = murder)))

(define-sem shooting-script is-a (terrorist-action-script)
   formulae (((1) = shoot
              (2) = murder
              (3) = wound)))

(define-sem bombing-script is-a (terrorist-action-script)
   formulae (((1) = bombing
              (2) = detonate
              (3) = explode
              (4) = destroy
              (5) = die
```

Table B.13: Test results, Camille 2.1, Assembly Line domain with scripts

Verb	**Meaning hypotheses**
allow	Allow
apply	Check-Object Load Place Toss
aside	Check-Object Load
break	Finish-Action
check	Check-Record Inspect-Record Read
crumple	Break Crumple
fasten	Aside
fold	Check-Record Fold Inspect-Record Read Tear
get	Get
install	Prepare-Action
load	Check-Object Get Load Lubricate Remove Restock
lubricate	Finish-Action
match	Match Verify
open	Check-Object Load Open Remove
place	Install Position
position	Install Position
preload	Finish-Action
reach	Inspect Repair
refill	Finish-Action
remove	Check-Object Load Remove
return	Inspect Repair
route	Uncoil
secure	Finish-Action
step	Step Walk
stock	Check-Object Get Load Lubricate Remove Restock
tear	Check-Record Fold Inspect-Record Read Tear
toss	Aside
uncoil	Finish-Action
verify	Prepare-Action
walk	Inspect Repair

```
        (6) = wound
        (7) = murder)))

(define-sem kidnapping-script is-a (terrorist-action-script)
   formulae (((1) = kidnapping)))
```

The scores achieved by Camille using these scripts were Recall 30%, Precision 43%, Production 60%, Accuracy 50%, and Parsimony 30%. Because of the differences in the testing procedure and the definitions that were collected, these results are not directly comparable to the basic test results. They do serve, however, to give a rough idea for the strengths and weaknesses of a script-based approach as discussed in section 4.2.

Table B.14: Test results, Camille 2.1, Terrorism domain with scripts

Verb	Meaning hypotheses
attacked	Destroy
destroyed	Destroy
kidnapped	Fight Threat
murdered	Murder
reported	Ambush Injure Kidnapping Murder Shoot
wounded	Murder

B.5.3 Ambiguous words

In order to test the learning of ambiguous verbs, the basic test was run with the addition of the mechanism described in section 4.3. The results of this test were exactly the same as for Camille 1.2 with the exception of the definition for open. As explained in section 4.3, this node was split into two different concepts, but the inferred hypothesis for both was Open.

As described in section 4.3.1, the test of the ambiguous noun acquisition mechanism required the removal of 9 word definitions from the corpus: branch, charge, lines, others, plant, post, quarter, state, and system. Examples of the 100 sentences which contained these words are shown below:

```
CHARGE:
According to Panamanian reports, charge d'affaires Luis Sandiga has
made statements on the situation that did not please the government.

According to a witness, the dynamite charge was placed by a young man.

STATE:
The Peruvian government today decreed a state of emergency and
military control in four Lima provinces and extended the measure in
another five.

As a shadow economy, it is trying to become part of the state
structure.

LINES:
We have broken the defensive lines of the enemy.

In the eastern part of the country, the Lempa River Hydroelectric
Executive Commission reported that one of the country's main power
lines was out of service on the morning of 1 June because a number of
pylons were destroyed.

PLANT:
U.S. policy, and all this is very well known in Latin America,
is based on destroying the plantations of a native plant in the
```

Table B.15: Test results, Camille 2.2, Assembly Line domain

Verb	**Meaning hypotheses**
allow	Inspect Repair
apply	Aside Check-Object Load Lubricate Place Restock Toss
aside	Check-Object Load Lubricate Restock
break	Break Crumple
check	Check-Object Finish-Action Load Lubricate Restock
crumple	Break Crumple
fasten	Fasten
fold	Check-Record Fold Inspect-Record Read Tear
get	Check-Object Get Load Lubricate Remove Restock
inspect	Action
install	Prepare-Action
load	Check-Object Get Load Lubricate Remove Restock
lubricate	Check-Object Get Load Lubricate Remove Restock
match	Match Verify
open	Open
open	Open
place	Install Position
position	Install Position
preload	Check-Object Get Load Lubricate Remove Restock
reach	Reach
read	Apply-Record Check-Record Fold Inspect-Record Read Tear
refill	Refill
remove	Check-Object Get Load Lubricate Remove Restock
repair	Action
return	Step Walk
route	Check-Object Get Load Lubricate Remove Restock Route
secure	Check-Object Load Lubricate Restock
step	Step Walk
stock	Check-Object Get Load Lubricate Remove Restock
tear	Check-Record Fold Inspect-Record Read Tear
toss	Aside Check-Object Load Lubricate Place Restock Toss
uncoil	Uncoil
verify	Match Verify
walk	Step Walk

```
continent, which is coca.

Coprefa reported that two soldiers were killed during a clash with
members of the Farabundo Marti National Liberation Front in Comasagua,
about 28 km to the southwest of Salvador, where a rebel attack on
a coffee processing plant was successfully repelled.
```

After the set of 100 sentences was processed, Camille 2.2 recorded the following definitions:

```
branch: Object-or-State
lines: Action
lines: Object-or-State
others: Place
others: Human
post: Human-or-Place
post: Energy
quarter: Object-or-State
state: Object-or-State (abstract notion of state)
state: State (government body / place notion)
system: Object-or-State
system: Energy
```

Although some of these definitions are questionable (especially Action for "lines"), those for "others", "post", and "state" are quite good. The inference of the meaning Object-or-State is of little use to an information extraction mechanism because it is so vague. An analysis of the semantic hierarchy reveals the reason for the vague inferences. The only verb with a constraint that references Object-or-State is Be. Thus, as suggested in the analysis of the limitations of the script mechanism, many of the sentences in the domain simply describe objects. This accounts for the inferences of the Object-or-State noun meanings in this test.

The system hypothesized 5 out of 9 ambiguous definitions, for a Production score of 56%. Recall, counting the correct definitions, was 8 of 18 possible definitions, or 44%. Precision and Accuracy were 8 out of 12, or 67%. Because there was only one concept in each sense of the ambiguous definitions, Parsimony was the same as Recall, or 44%.

B.5.4 Creating new nodes

As described in section 4.4, the creation of new concepts for the system relies on two other variations of Camille, Mutual Exclusivity and noun learning. The test on this mechanism was exactly the same as the "ambiguous" test. Instead of just attaching a new definition to an existing concept, however, Camille 2.3 created a new node in the hierarchy if Mutual Exclusivity had already assigned a word to that concept. The new concept was given the name of the word if no such concept-name already existed. Otherwise, a name was created based on the word. The results of the test were as follows (new concepts are flagged with a * and followed by their parents in the hierarchy):

```
branch:   Branch*          (Object-or-State)
lines:    Lines*           (Action)
lines:    Lines35764*      (Object-or-State)
others:   Place
others:   Other30078*      (Human)
post:     Human-or-Place
post:     Post*            (Energy)
quarter:  Quarter36164*    (Object-or-State)
state:    State33349*      (State)
state:    State32026*      (Object-or-State)
system:   System*          (Object-or-State)
system:   System32324*     (System*)
```

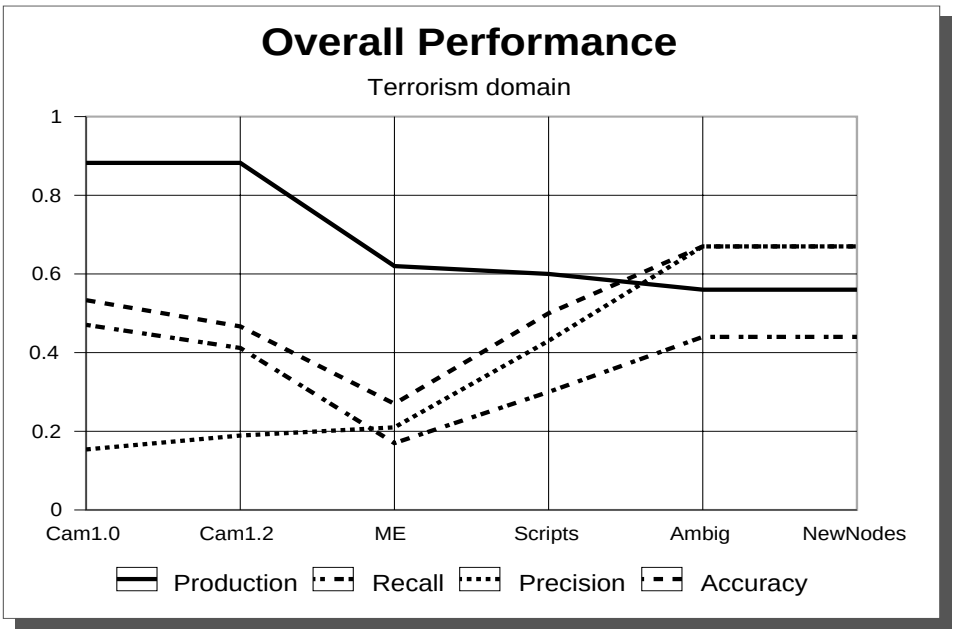


Figure B.16: Overall Camille performance, Terrorism domain

As previously mentioned, the Terrorism domain was a difficult one for Mutual Exclusivity. In a sense, this made it a good one for creating new nodes. Because most of the concepts in the domain were already referents of one or more words, the occurrence of an unknown word could signal Camille that it needed to further specify its concept structure.

The Production score was the same as for the ambiguous nouns test: 5 out of 9, or 56%. Because there were also 8 correct definitions, the Recall, Precision, Accuracy, and Parsimony scores were also the same as for the ambiguity test: 44%, 67%, 67%, and 44%. Of the 12 definitions that Camille created, only two did not result in the creation of a new concept. These two concepts, Place and Human-or-Place were so general that there were no other words in the lexicon that referred to them.

Figure B.16 summarizes the scores of the system for all of the different variations.

BIBLIOGRAPHY

[Allen, 1981] J. Allen. What's necessary to hide?: Modeling action verbs. In *Proceedings of the* 19^{th} *Annual Meeting of the Association for Computational Linguistics*, pages 77–81, 1981.

[Barwise and Etchemendy, 1989] J. Barwise and J. Etchemendy. Model-theoretic semantics. In M. Posner, editor, *Foundations of Cognitive Science*. MIT Press, Cambridge, MA, 1989.

[Behrend, 1990] D. Behrend. The development of verb concepts: Children's use of verbs to label familiar and novel events. *Child Development*, 61:681–696, 1990.

[Berwick, 1985] R. Berwick. *The Acquisition of Syntactic Knowledge*. MIT Press, Cambridge, MA, 1985.

[Bloom *et al.*, 1980] L. Bloom, K. Lifter, and J. Hafitz. Semantics of verbs and the development of verb inflection in child language. *Language*, 56(2):386–412, 1980.

[Bowerman, 1976] M. Bowerman. Semantic factors in the acquisition of rules for word use and sentence construction. In D. Morehead and A. Morehead, editors, *Normal and deficient child language*. University Park Press, Baltimore, 1976.

[Bowerman, 1983] M. Bowerman. How do children avoid constructing an overly general grammar in the absence of feedback about what is not a sentence? In *Proceedings of Research on Childrens Language Development*, volume 22, 1983.

[Brent, 1991] M. Brent. Automatic acquisition of subcategorization frames from untagged text. In *Proceedings of the* 29^{th} *Annual Meeting of the Association for Computational Linguistics*, pages 209–214, 1991.

[Brent, 1993a] M. Brent. From grammar to lexicon: Unsupervised learning of lexical syntax. *Computational Linguistics*, 1993. in press.

[Brent, 1993b] M. Brent. Surface cues and robust inference as a basis for the early acquisition of subcategorization frames. *Lingua*, 1993. in press.

[Cardie, 1993] C. Cardie. A case-based approach to knowledge acquisition for domain-specific sentence analysis. In *Proceedings of the* 11^{th} *National Conference on Artificial Intelligence*, pages 798–803, 1993.

[Carey and Bartlett, 1978] S. Carey and E. Bartlett. Acquiring a single new word. *Papers and reports on child language development* (Department of Linguistics, Stanford University), 15:17–29, 1978.

[Carey, 1978] S. Carey. The child as word learner. In M. Halle, G. Miller, and J. Bresnan, editors, *Linguistic theory and psychological reality*, pages 264–293. MIT Press, Cambridge, MA, 1978.

[Chinchor, 1992] N. Chinchor. MUC-4 evaluation metrics. In *Proceedings of the Fourth Message Understanding Conference*, San Mateo, CA, 1992. Morgan Kaufmann Publishers, Inc.

[Chomsky, 1981] N. Chomsky. Principles and parameters in syntactic theory. In N. Hornstein and D. Lightfoot, editors, *Explanation in Linguistics: The Logical Problem of Language Acquisition*. Longman, London, 1981.

[Chomsky, 1985] N. Chomsky. *Knowledge of Language*. Praeger Publications, New York, 1985.

[Chomsky, 1986] N. Chomsky. *Barriers*. MIT Press, Cambridge, MA, 1986.

[Church and Hanks, 1990] K. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16, 1990.

[Clark, 1987] E. Clark. The principle of contrast: A constraint on language acquisition. In B. MacWhinney, editor, *Mechanisms of Language Acquisition*. Lawrence Erlbaum Inc, Hillsdale, NJ, 1987.

[Clark, 1989] E. Clark. On the logic of contrast. *Journal of Child Language*, 15:317–335, 1989.

[Corter and Gluck, 1992] J. Corter and M. Gluck. Explaining basic categories: Feature predictability and information. *Psychological Bulletin*, 111(2):291–303, 1992.

[Cullingford, 1977] R. Cullingford. *Organizing World Knowledge for Story Understanding by Computer*. PhD thesis, Yale University, New Haven, CT, 1977.

[Dennett, 1978] D. Dennett. *Brainstorms*. MIT Press, Cambridge, MA, 1978.

[Fernald and Morikawa, 1993] A. Fernald and H. Morikawa. Common themes and cultural variations in Japanese and American mothers' speech to infants. *Child Development*, 64:637–656, 1993.

[Gentner, 1978] D. Gentner. On relational meaning: The acquisition of verb meaning. *Child Development*, 49:988–998, 1978.

[Gleitman, 1990] L. Gleitman. The structural sources of verb meaning. *Language Acquisition*, I(1):3–55, 1990.

[Goldin-Meadow *et al.*, 1976] S. Goldin-Meadow, M. Seligman, and R. Gelman. Language in the two-year-old. *Cognition*, 4:189–202, 1976.

[Golinkoff *et al.*, 1987] R. Golinkoff, P. Hirsh-Pasek, K. Cauley, and L. Gordon. The eyes have it: Lexical and syntactic comprehension in a new paradigm. *Journal of Child Language*, 14:23–46, 1987.

[Gopnik and Choi, 1990] A. Gopnik and S. Choi. Do linguistic differences lead to cognitive differences? a cross-linguistic study of semantic and cognitive development. *First Language*, 10:199–215, 1990.

[Graesser *et al.*, 1987] A. Graesser, P. Hopkinson, and C. Schmid. Differences in interconcept organization between nouns and verbs. *Journal of Memory and Language*, 26:242–253, 1987.

[Granger, 1977] R. Granger. Foul-up: A program that figures out meanings of words from context. In *Proceedings of Fifth International Joint Conference on Artificial Intelligence*, 1977.

[Grimshaw, 1979] J. Grimshaw. Complement selection and the lexicon. *Linguistic Inquiry*, 10:279–326, 1979.

[Grimshaw, 1981] J. Grimshaw. Form, function and the language acquisition device. In C. L . Baker and J. J. McCarthy, editors, *The logical problem of language acquisition.* MIT Press, Cambridge, MA, 1981.

[Hastings and Lytinen, 1991] P. Hastings and S. Lytinen. Automatic acquisition of word meanings. In D. Powers and L. Reeker, editors, *Proceedings of the AAAI Spring Symposium on Machine Learning of Natural Language and Ontology, Document D-91-09*, University of Kaiserslautern, FRG, 1991. DFKI.

[Hastings and Lytinen, in press] P. Hastings and S. Lytinen. Acquiring new words from context. *Heuristics: The Journal of Knowledge Engineering*, in press.

[Hastings *et al.*, 1991a] P. Hastings, S. Lytinen, and R. Lindsay. Learning words from context. In L. Birnbaum and G. Collins, editors, *Machine Learning: Proceedings of the Eighth International Workshop*, pages 55–59, San Mateo, CA, 1991. Morgan Kaufmann.

[Hastings *et al.*, 1991b] P. Hastings, S. Lytinen, and R. Lindsay. Learning words: Computers and kids. In K. Hammond and D. Gentner, editors, *Proceedings of the 13th Annual Conference of the Cognitive Science Society*, pages 251–256, Hillsdale, NJ, 1991. Lawrence Erlbaum Associates.

[Hastings *et al.*, 1991c] P. Hastings, S. Lytinen, and R. Lindsay. Psycholinguistic implications of a computational language-learning model. In D. Powers, L. Reeker, and B. Humm, editors, *Proceedings of the Workshop on Natural Language Learning of the 12th International Joint Conference on Artificial Intelligence*, 1991.

[Heibeck and Markman, 1987] T. Heibeck and E. Markman. Word learning in children: An examination of fast mapping. *Child Development*, 58:1021–1034, 1987.

[Hindle, 1990] D. Hindle. Noun classification from predicate-argument structures. In *Proceedings of the* 28^{th} *Annual Meeting of the Association for Computational Linguistics*, pages 268–275, 1990.

[Hirsh-Pasek and Golinkoff, 1993] K. Hirsh-Pasek and R. Golinkoff. Skeletal supports for grammatical learning: What infants bring to the language learning task. *Advances in Infancy Research*, 8:299–338, 1993.

[Hobbs *et al.*, 1992] J. Hobbs, D. Appelt, M. Tyson, J. Bear, and D Israel. SRI International: Description of the FASTUS system used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference*, San Mateo, CA, 1992. Morgan Kaufmann Publishers, Inc.

[Huffman *et al.*, 1993] S. Huffman, C. Miller, and J. Laird. Learning from instruction: A knowledge-level capability within a unified theory of cognition. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, 1993.

[Huttenlocher and Lui, 1979] J. Huttenlocher and F. Lui. The semantic organization of some simple nouns and verbs. *Journal of verbal learning and verbal behavior*, 18:141–162, 1979.

[Huttenlocher *et al.*, 1991] J. Huttenlocher, W. Haight, A. Bryk, M. Seltzer, and T. Lyons. Early vocabulary growth: Relation to language input and gender. *Developmental Psychology*, 27(2):236–248, 1991.

[Huyck, 1993] C. Huyck. Efficient heuristic natural language parsing. In *Proceedings of the* 11^{th} *National Conference on Artificial Intelligence*, pages 386–391, 1993.

[Jackendoff, 1983] R. Jackendoff. *Semantics and cognition.* MIT Press, Cambridge, MA, 1983.

[Kaplan *et al.*, 1990] S. Kaplan, M. Weaver, and R. French. *Active symbols and internal models: Towards a cognitive connectionism.* Springer Verlag, London, 1990. Springer Series on Artificial Intelligence and Society.

[Katz and Fodor, 1963] Jerrold J. Katz and Jerry A. Fodor. The structure of a semantic theory. *Language*, 39, 1963.

[Keil, 1991] F. Keil. Theories, concepts, and the acquisition of word meaning. In J. P. Byrnes and S. A. Gelman, editors, *Perspectives on language and thought: Interrelations in development.* Cambridge University Press, Cambridge, 1991.

[Lebowitz, 1980] M. Lebowitz. Generalization and memory in an integrated understanding system. Research Report No. 186, Yale University, New Haven, CN, October 1980.

[Lehnert, 1990] W. Lehnert. Description of the CIRCUS system as used for MUC-3. In *Proceedings, Third Message Understanding Conference (MUC-3)*, pages 223–233, San Diego, CA, 1990. Morgan Kaufmann Publishers.

[Lehnert, 1992] W. Lehnert. University of Massachusetts: MUC-4 test results and analysis. Talk given at Fourth Message Understanding Conference (MUC-4), June 1992.

[Lenat, 1990] D. Lenat. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.* Addison-Wesley Publishing Company, Reading, MA, 1990.

[Lytinen and Roberts, 1989a] S. Lytinen and S. Roberts. Lexical acquisition as a by-product of natural language processing. In 11^{th} *International Conference on Artificial Intelligence*, 1989. Lexical Acquisition Workshop.

[Lytinen and Roberts, 1989b] S. Lytinen and S. Roberts. Unifying linguistic knowledge. AI Laboratory, Univ of Michigan, Ann Arbor, MI 48109, 1989.

[Lytinen *et al.*, 1992a] S. Lytinen, S. Bhattacharyya, R. Burridge, P. Hastings, C. Huyck, K. Lipinsky, E. McDaniel, and K. Terrell. The LINK system: MUC-4 test results and analysis. In *Proceedings of the Fourth Message Understanding Conference*, pages 159–163, San Mateo, CA, 1992. Morgan Kaufmann Publishers.

[Lytinen *et al.*, 1992b] S. Lytinen, S. Bhattacharyya, R. Burridge, P. Hastings, C. Huyck, K. Lipinsky, E. McDaniel, and K. Terrell. Description of the LINK system used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference*, pages 289–295, San Mateo, CA, 1992. Morgan Kaufmann Publishers.

[Lytinen *et al.*, in press] S. Lytinen, R. Burridge, P. Hastings, and C. Huyck. Description of the LINK system used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference*, San Mateo, CA, in press. Morgan Kaufmann Publishers.

[Lytinen, 1988] S. Lytinen. Are vague words ambiguous? In S. Small and G. Cottrell, editors, *Lexical Ambiguity Resolution*, pages 109–128. Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[Lytinen, 1990] S. Lytinen. Robust processing of terse text. In *Proceedings of the 1990 AAAI Symposium on Intelligent Text-based Systems*, pages 10–14, Stanford, CA, 1990.

[Lytinen, 1991] S. Lytinen. A unification-based, integrated natural language processing system. *Computers and Mathematics with Applications*, 23(6-9):403–418, 1991.

[MacGregor, 1990] R. MacGregor. The evolving technology of classification-based knowledge representation systems. In J. Sowa, editor, *Principles of Semantic Nets: Explorations in the Representation of Knowledge*. Morgan Kaufmann Publishers, San Mateo, CA, 1990.

[Mandler, 1988] J. Mandler. How to build a baby: On the development of an accessible representational system. *Cognitive Development*, 3:113–136, 1988.

[Mandler, 1992] J. Mandler. How to build a baby: II. Conceptual primitives. *Psychological Review*, 99:587–604, 1992.

[Markman, 1990] E. Markman. Constraints children place on word meanings. *Cognitive Science*, 14(1):57–77, Jan–Mar 1990.

[Markman, 1991] E. Markman. The whole object, taxonomic, and mutual exclusivity assumptions as initial constraints on word meanings. In J. P. Byrnes and S. A. Gelman, editors, *Perspectives on language and thought: Interrelations in development*. Cambridge University Press, Cambridge, 1991.

[Marr, 1982] D. Marr. *Vision*. W. H. Freeman and Company, San Francisco, CA, 1982.

[Mitchell, 1977] T. Mitchell. Version spaces: A candidate elimination approach to rule learning. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 305–309, 1977.

[Naigles, 1990] L. Naigles. Children use syntax to learn verb meanings. *Journal of Child Language*, 17:357–374, 1990.

[Newell, 1990] A. Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA, 1990.

[Pinker, 1984] S. Pinker. *Language learnability and Language Development*. Harvard University Press, Cambridge, MA, 1984.

[Pylyshyn, 1989] Z. Pylyshyn. Computing in cognitive science. In M. Posner, editor, *Foundations of Cognitive Science*. MIT Press, Cambridge, MA, 1989.

[Quinlan, 1992] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1992.

[Resnik, 1992] P. Resnik. A class-based approach to lexical discovery. In *Proceedings of the* 30^{th} *Annual Meeting of the Association for Computational Linguistics*, pages 327–329, 1992.

[Riloff, 1993] E. Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the* 11^{th} *National Conference on Artificial Intelligence*, pages 811–816, 1993.

[Salveter, 1979] S. Salveter. Inferring conceptual graphs. *Cognitive Science*, 3:141–166, 1979.

[Salveter, 1980] S. Salveter. Inferring conceptual graphs. In *Proceedings of the* 18^{th} *Annual Meeting of the Association for Computational Linguistics*, pages 13–15, 1980.

[Schank and Abelson, 1977] R. Schank and R. Abelson. *Scripts, plans, goals, and understanding.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

[Schank, 1973] R. Schank. Identification of conceptualizations underlying natural language. In Roger Schank and K.M. Colby, editors, *Computer Models of Thought and Language.* W.H. Freeman, San Francisco, 1973.

[Schank, 1981] R. Schank. *Inside Computer Understanding.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.

[Selfridge, 1986] Mallory Selfridge. A computer model of child language learning. *Artificial Intelligence*, 29:171–216, 1986.

[Selfridge, 1991] M. Selfridge. How do children learn to recognize ungrammatical sentences? In D. Powers and L. Reeker, editors, *Proceedings of the AAAI Spring Symposium on Machine Learning of Natural Language and Ontology, Document D-91-09.* DFKI, University of Kaiserslautern, FRG, 1991.

[Shatz and Ebeling, 1991] M. Shatz and K. Ebeling. Patterns of language learning-related behaviours: evidence for self-help in acquiring grammar. *Journal of Child Language*, 18:295–313, 1991.

[Sidner, 1979] C. L. Sidner. *Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse.* PhD thesis, MIT, 1979.

[Siskind, 1990] J. Siskind. Acquiring core meanings of words. In *Proceedings of the* 28^{th} *Annual Meeting of the Association for Computational Linguistics*, pages 143–156, 1990.

[Siskind, 1991] J. Siskind. Dispelling myths about language bootstrapping. In D. Powers and L. Reeker, editors, *Proceedings of the AAAI Spring Symposium on Machine Learning of Natural Language and Ontology, Document D-91-09*, University of Kaiserslautern, FRG, 1991. DFKI.

[Spelke, 1982] E. Spelke. Perceptual knowledge of objects in infancy. In J. Mehler, E. Walker, and M. Garrett, editors, *Perspectives on mental representations.* Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, 1982.

[Sundheim, 1992] B. Sundheim. Overview of the fourth message understanding evaluation and conference. In *Proceedings of the Fourth Message Understanding Conference*, San Mateo, CA, 1992. Morgan Kaufmann Publishers.

[Tomasello, 1992] M. Tomasello. *First Verbs: A case study of early grammatical development.* Cambridge University Press, Cambridge, England, 1992.

[Turing, 1950] A. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

[Waxman *et al.*, 1991] S. Waxman, E. Shipley, and B. Shepperson. Establishing new subcategories: The role of category labels and existing knowledge. *Child Development*, 62:127–138, 1991.

[Wilensky, 1978] R. Wilensky. Understanding goal-based stories. Research Report 140, Department of Computer Science, Yale University, 1978.

[Winograd, 1987] T. Winograd. *Language as a Cognitive Process. Vol. 1: Syntax.* Addison-Wesley Publishing, Reading, MA, 1987.

[Yarowsky, 1992] D. Yarowsky. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proceedings, COLING-92*, 1992.

[Zernik, 1987a] U. Zernik. Strategies in language acquisitions: Learning phrases from examples in context. Technical Report UCLA-AI-87-1, UCLA, January 1987.

[Zernik, 1987b] Uri Zernik. How do machine language paradigms fare in language acquisition. In *Proceedings of the Fourth International Workshop on Machine Learning*, Los Altos, CA, 1987. Morgan Kaufmann.

[Zernik, 1991] U. Zernik. Train1 vs. train2: Tagging word senses in corpus. In U. Zernik, editor, *Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon.* Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, 1991.