Language Acquisition: Learning a Hierarchy of Phrases

Uri Zernik
Computer Science Department
University of California
Los Angeles, California 90024

Current Address:
GE Corporate Research and Development
Schenectady, New York 12301

Abstract

The *hierarchical* lexicon, in contrast to the traditional *flat* lexicon, enables a linguistic model to perform even in situations of incomplete knowledge: when a specific entry is missing, a more general entry can cover the gap. The question still remains regarding the construction of the lexicon itself.

Since the lexicon is organized as a hierarchy, and not as a flat structure, phrases cannot simply be placed in the lexicon: they must be interconnected with other phrases in the hierarchy at the appropriate level of generality. Furthermore, since input examples arc always given in terms of specific phrases, phrases must be propagated up and down the hierarchy, starting at the bottom level.

In this paper we describe a learning algorithm which is based on two existing machine-learning models: learning in a *version space* [Mitchell82], and learning by accumulating specific *episodes* in a dynamic memory [Kolodner84, Schank82]. The input required by the algorithm is a sequence of specific episodes, or training examples, from which lexical entries at various levels in the hierarchy arc *generalized* and *specialized.* The algorithm is embodied by the program RINA [Zernik87b] which models learning English phrases by a second language speaker.

1. INTRODUCTION

All language-related knowledge required by a linguistic model resides in its lexicon. Traditionally, the lexicon has been viewed as a *flat* list of lexical entries [Bresnan82]. Recently, however [Jacobs85, Langacker86], *hierarchy* has emerged as a method for organizing lexical information. Through the hierarchical representation, linguistic knowledge ranging form general grammar rules, to specific lexical entries can be covered uniformly. A significant advantage of this approach is the ability to cope with *incomplete knowledge-** problem faced by any running linguistic model, as well as by humans.

Two aspects of this problem must be investigated: performance-parsing and generation-and acquisition. In another paper [Zernik86b] we show how the hierarchical lexicon facilitates performance even in conditions of incomplete knowledge: through the hierarchy, general phrases can fill in for missing specific ones. However, how is the lexicon generated in the first place? In this paper we show how the hierarchy itself is constructed dynamically, as a by-product of parsing. The process is described in terms of the computer program RINA which acquires phrases by receiving examples from a user [Zernik85b, Zernik86a, Zernik85a].

1.1 The Linguistic Phenomenon

Four scenarios are distinguished here, according to two parameters: (a) the contents of the program's lexicon, and (b) the clues given by the user's example. The entire sequence illustrates how the program upgrades its parsing ability as experiences are being accumulated.

(1) Neither the Entry Exists in the Lexicon, Nor is the Context Provided by the User: Initially, the lexicon contains no entry to cope with the new phrase to push around.

User: Throughout their childhood,
Frank used to push Corinne around.
RINA: He moved her in circles?

Although the words push and around exist in the lexicon, the entire phrase push somebody around is yet unknown. Neither does the context, established by the text (throughout their childhood does not reveal the essence of the characters' relationship), provide any clue. Therefore, the program cannot cope with the new phrase, and it uses the literal interpretation of the single words: he moved her in circles?.

(2) No Entry Exists, but an Explanatory Context is Given: While no entry exists in the lexicon for push around, this time the context provided by the user is explanatory:

User: Frank, Corinne's elder brother, pushed her around.
He asked her to make his bed against her will.
RINA: Frank made Corinne carry out a job on his behalf.

Here, due to the context provided by the user, RINA forms an initial hypothesis. However, based on a single example, the hypothesis is not general enough (doing a job on behalf of a second person is not the essence of the meaning).

(3) A Similar Entry Exists, and a Context is Given: Next, RINA hears a similar phrase, to boss around.

User: Mary was hired on the same day as John, but she knew the manager. Therefore, she used to boss around her colleague, John.
RINA: Mary forced him to act.

At this point, RINA could identify the similarities between the two situations. Basically, RINA detected a common surface feature (the word around), and common semantic features (one person uses his power to impose his will on a second person). She constructs a generalization regarding the association between these features, and accordingly, she conveys a more general hypothesis to the user. This hypothesis may help the learner predict the behavior of other phrases involving the word around. However, since there are phrases involving around which do not have this meaning (e.g., stick around, get around to it, etc.), the acquired generalized phrase must carry discriminating conditions.

(4) No Context is Given, but a General Entry Exists: Next, RINA hears a third related phrase, to order around:

User: Doug, Linda's younger brother, told her to stop ordering him around.
RINA: Doug asked Linda to stop forcing him to act?

Like the first scenario, neither: (a), the phrase order somebody around is encoded explicitly in the lexicon, nor (b), any explanatory context is provided by the user. However, in absence of specific knowledge for parsing the new phrase, RINA applies here the general phrase acquired previously. RINA's hypothesis is not guaranteed to be correct, yet it enables the formation of a hypothesis regarding the unknown phrase.

Notice how learning improved the performance of the program.

- Before learning, RINA was not able to parse push around (without an explanatory context).
- After learning, even in conditions of minimal context, RINA could parse sentences which include, not only the phrase itself, but an entire set of similar phrases.

1.2 The Issues

Five issues must be addressed in acquiring such a hierarchical lexicon.

Determining Scope and Variability: The syntactic pattern of a new phrase is extracted initially from a single sentence.

Frank used to push Corinne around.

From that sentence, the program must determine scope and the variability of the pattern. However, the unknown pattern could take on many forms.

Frank pushed Corinne
X:person push:verb Y:person
X:person push:verb Y:person around
X:person use:verb to push:verb Y:person around

Which one of the forms above is appropriate as the phrase pattern?

Forming the Meaning: The meaning of the phrase is extracted form the context. However, the context includes many concepts, some appropriate, and some inappropriate for meaning formation. Thus, based on the given context of Frank and Corinne, what is the meaning of push around?

Generalization: From the first example, RINA extracted a narrow meaning:

RINA: Frank made Corinne carry out a job on his behalf.

By hearing a second example, RINA was able to generalize the meaning:

RINA: Mary forced John to act against his will.

This meaning pertains to general interpersonal relationships (power, authority) and can cover more situations in which the phrase may appear. (1) How can a generalization be formed across two apparently distinct situations (family feud vs. problems at the working place)? (2) How far should generalization be pursued without causing overgeneralizing (e.g., here is an overgeneralization: Frank did something negative to Corinne)?

Specialization: A new phrase order around is encountered without an explanatory context. Yet, although a specific lexical entry for that phrase does not exist, the phrase can be analyzed by a generalized phrase. How is generalized knowledge applied in coping with gaps in specific knowledge?

Phrase Disambiguation: Even when all the necessary information exists in the lexicon, parsing is not without problems. Consider the following pair of texts:

- Mary and John were ice skating in circles. He was pushing her around.
- John acted bossy with Mary. He was always pushing her around.

Since two phrases exist for push around, the program must select the appropriate one in each case. Individual lexical entries must provide the sufficient conditions for phrase discrimination.

1.3 The Approach

The phrasal lexicon [Becker75, Wilensky81, Fillmore87] contains not only single words (e.g., around), rather it contains entire phrasal entries (e.g., to get around to it, to suck around, around the clock, around 5pm). The phrasal approach has proved effective in parsing [Wilks75] and in generation [Jacobs85]. Yet the basic dilemma regarding inclusion of phrases in the lexicon is unresolved. Which linguistic elements belong in the phrasal lexicon, and which elements do not? On the one hand, productive phrases such as John gave Mary the spoon, or Mary went to school, should not reside in the lexicon, lest the lexicon will get oversized. On the other hand, non-productive phrases such as the big apple, or to throw the book at somebody must reside in the lexicon. Their meanings cannot be derived from their constituents. However, there is a gray area which includes phrases such as stick around, hang around, and sit around, for whom the inclusion question is unclear. These phrases are non-productive, since their meanings cannot be simply derived from the single words. Thus, they must be included. However, they all share semantic and syntactic features, thus they should not all be included as separate entries. Therefore in our model, such groups of phrases are clustered into generalized phrases, which represent their shared features. Such generalized phrases serve in predicting meanings of similar combinations even before they have been encountered.

The hierarchical representation facilitates learning as a continuous process of knowledge refinement. Two approaches have been integrated in our model. Using Mitchell's method [Mitchell82] we view the lexicon as a version space of rules. Since phrases are organized in a hierarchy by generalization, then both parsing and learning require determining location of phrases in that hierarchy. Kolodner [Kolodner84], on the other hand, has shown how learning is accomplished by organizing episodes in a dynamic memory. In our model, the lexical hierarchy itself is dynamic and nodes at all levels of generality are updated by receiving instance episodes at the bottom of the hierarchy.

2. REPRESENTING THE CONTEXT

The context is represented as a goal-plan situation at two levels: specific planning, and abstract planning.

Specific Planning Knowledge: Planning knowledge in each domain consists of goals-and acts which implement those goals. For example, consider the text:

> Frank needed to clean up his room. Corinne made his bed for him.

The events underlying the sentence are given in the following plan box [Schank77].

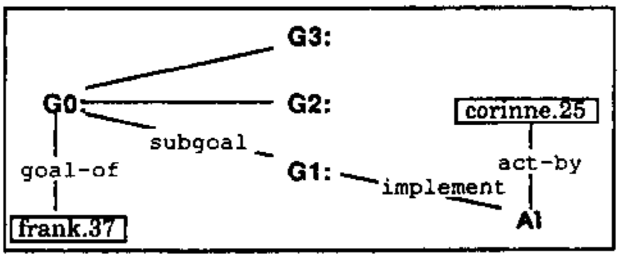

Figure 1: The Specific Plan Box

Making the bed (GI), wiping the floor (G2), clearing the desk (G3), are all subgoals for cleaning one's room (GO). In this case, although GO is a goal of Frank, Act AI which implements GI is executed by Corinne.

Abstract Planning Knowledge: Many specific plan boxes are required in covering planning situations across domains. In contrast, a relatively small set of abstract planning structures [Wilensky83] (e.g., goal-conflict, goal-competition, goal-concordance, plan agency, etc.), and relations (authority, power, friendship) can cover situations across all domains. In our example, due to the power relationship between characters X and Y, X causes Y execute an act A on X's behalf. This is shown in figure 2 below.

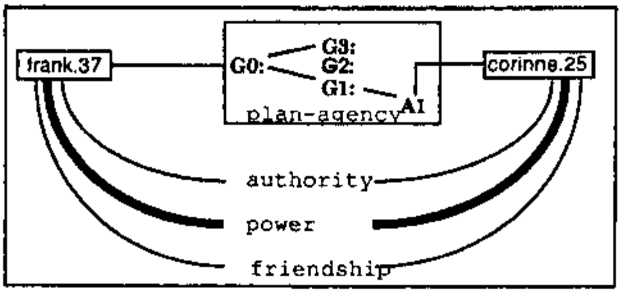

Figure 2: An Abstract Goal Situation

Corinne serves as a plan agent in executing a plan for Frank. Her motivation is neither (a) an authority relationship (Frank is not an authority for Corinne), nor (b) a friendship with Frank, but (c) the power relationship (due to his Frank's age). The abstract planning elements are used here to explain the planning situation. Planning knowledge at these two levels provide the semantics for lexical representation.

3. THE LEXICON

The phrasal lexicon [Zernik87a] is specified in two ways: by the contents of single lexical entries, and by the structure of the entire lexicon.

Single-Phrase Representation: A lexical entry, or a phrase, is a triple associating a linguistic pattern with its concept and a presupposition. For example compare the following sentences:

(1) John sat three days working on his thesis,
(2) John sat around three days working on his thesis.

What is the difference between their meanings, and what is the value added by the modifier around? The phrase to sit around is represented as a triple:

pattern:	X:person sit around
presupposition:	X dedicates time to an act P
concept:	the execution of P is a waste of time

Sentence (2), then, is parsed in three steps using the lexical phrase:

(1) The pattern is matched successfully against the text. Consequently, X is bound to the person called John.

(2) The presupposition is unified with the current context. In this case the variable P is taken by the context as the Ph.D work.

(3) Since both (1) and (2) are successful, then the pattern itself is instantiated, adding to the context: working on the PhD wasted John's time (which disabled execution of other acts by John).

Generalized Phrases: However, the representation given above for hang around is not satisfactory when considering two other similar phrases:

John used to stick around for hours.
John hung around, waiting for his girlfriend.

The lexicon cannot simply enumerate all word combinations. There must be a more general entry to account for all the phrases of the from "verb around" which convey this particular sense.

pattern: X:person V:act around
presupposition: V is an act of staying in location L for a period of time
concept: staying in L disables execution of other acts by X

This entry captures an entire set of such phrases. The presupposition part is used to discriminate phrases such as hang around, and stick around from phrases such as push somebody around and boss somebody around, by detecting necessary conditions in the context.

The Global Hierarchy: A hierarchy is defined, to accommodate for phrases at all levels of generality.

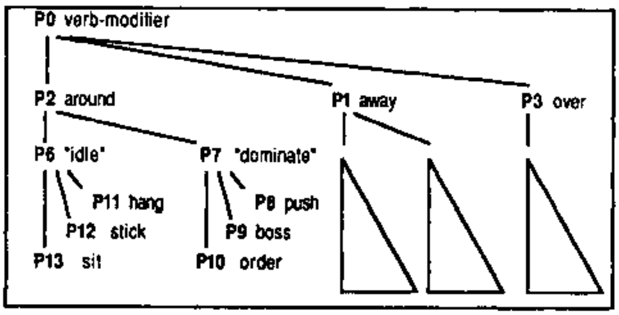

Figure 3: The Hierarchy for Verb-Modifiers

(The names of the nodes in this scheme are for reference only.) Phrases in this hierarchy are all given uniformly as pattern-concept-presupposition triples. For example, PI3 (sit) is the phrase for sit around, and P6 ("idle") is the phrase encompassing a set of V around phrases. Specific phrases, traditionally called "lexical rules", reside near the bottom. General phrase, or "grammar rules", reside at the top. P2 (around), for example, maintains general knowledge about the behavior of that modifier.

4 THE ALGORITHM

The learning algorithm accepts as input a sequence of episodes, from which it generates a subtree of phrases. The hierarchy before and after learning is given schematically below, where subtree T4 is the element added to the lexicon by learning.

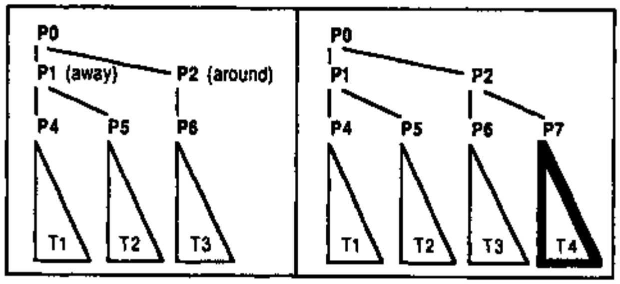

Figure 4: The Hierarchy Before and After

The process itself is shown schematically through a sequence of snapshots.

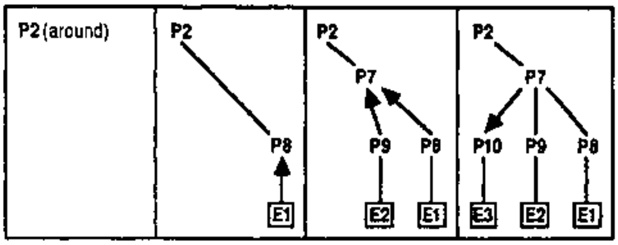

Figure 5: Four Snapshots in the Sequence

(1) Initially, the subtree includes only a the general node P2 (which stands for the modifier around).

(2) Node P8 is extracted from episode E1.

(3) Node P9 is extracted from episode E2. However, P7 is generalized by detecting the similarities between EI and E2,

(4) E3 is not informative enough for extraction of a node. However, node P10 can be specialized from the general node P7.

Thus, a subtree is added under P2. The specifications of the algorithm are given below. The given information is:

(1) An initial hierarchy. This hierarchy is not sufficiently developed, so parsing of certain sentences might fail.

(2) A sequence of episodes. An episode is a paragraph embedding a new phrase in an explanatory context.

(3) The semantics of the context. Semantic knowledge is given in terms of planning knowledge.

The objective is to refine the initial hierarchy. By adding on additional nodes, or by specializing and generalizing existing nodes, the new hierarchy should facilitate parsing of sentences on which the initial hierarchy has failed. The three basic steps of the algorithm are explained in regard to the sequence of scenarios in section 1.1.

4.1 Extracting a Phrase from a Single Example

Two simultaneous actions are involved in acquiring an unknown phrase from a training example (Frank pushed his younger sister around).

(1) The pattern is extracted from the input sentence.

(2) The meaning (both *concept* and *presupposition)* is extracted from the given context.

(1) Forming the Pattern

Two problems arise in pattern formation: determining the scope and determining the variability of the new pattern.

Determining the scope: Since the input sentence includes not only the bare phrase itself, but some additional elements, it is necessary to determine which elements in the sentence should be included in the pattern and which ones should be excluded. For example, the sentence under consideration contains two modifiers:

Throughout their childhood, Frank used to push Corinne around.

While throughout their childhood is excluded, around is taken as a part of the new phrase. The reason is that throughout their childhood can be interpreted successfully by the parser in any context. It does not cause a parsing failure. However, around does cause a parsing failure: "Frank moved Corinne around something"? Thus, the pattern is scoped as follows:

X:person push Y:person around

Determining variability: In converting an instance sentence into a general pattern, each element in the new pattern either becomes a variable, or it remains as a literal. In the conversion above, both Frank and Corinne were taken as generalized references to persons, X and Y. Accordingly, it appears that the rule was:

References to persons and to objects are converted into variables.

However, this rule is refuted by examples such as:

(1) Admiral Nellson went to Davey Jones' locker.
(2) He kicked the bucket in Trafalgar.

In these idioms, the marked references are taken as literals and not as variables, in violation of the rule above. Thus, the rule must be refined:

References which can be resolved in the context are converted into variables. References which cannot be resolved in the context are kept as literals.

Further strategies for pattern formation are described elsewhere [Zemik87b].

(2) Forming the Meaning

Initially, the program focuses at the level of specific plans.

G (cleaning the room) is goal of X (Frank).
P (making the bed) is a plan for G.
P is carried out by Y (Corinne).

The meaning captured at this level is: X carried out a job on Y's behalf. Consequently, the acquired phrase is:

pattern: X:person <push around> Y:person
presupposition: P is a plan for G
concept: Y is a plan agent in executing plan P for X

P8: The Specific Phrase for *push around*

However, many situations in which push around may be applied, are not included in P8's definition. Thus, P8 must be generalized.

4.2 Generalizing from Two Examples

The second scenario, Mary bossed her colleague around, does not share the specific features with the first example. Syntactically: the verb is boss and not push. Semantically: John did not carry out a job in Mary's behalf. Thus both the pattern and the meaning must be generalized.

The pattern is generalized as the verb becomes a variable. The meaning is generalized when specific planning elements are replaced by *abstract planning* structures which are shared by both episodes: Mary used her power (and not an authority), and Frank used his power to make Corinne act against her will. Therefore, the resulting phrases is:

pattern: X:person <V:act around> Y:person
presupposition: X presents a power for Y
concept: X causes Y to act against X's will

P7: The Generalized Phrase for "dominating"

Notice that although the general phrase P7 was generated, the specific phrases P8 and P9 have not been eliminated. Specific information is maintained even though it is sub* sumed by more general information.

4.3 Specializing a General Phrase

A specific phrase does not exist for order around in the fourth scenario. However, the sentence is parsed by using the generalized phrase P7. P7 is matched successfully in the episode E3, since:

(1) The syntactic pattern of P7 matches the input sentence (the form is X Verbed Y around).

(2) The semantic presupposition matches the input context (X presents a power to Y).

The specific phrase P10 (for *order around*) is added on to the lexicon, although it is subsumed by P7. For one thing, specific phrases enrich the program's vocabulary. In text generation they enable production of sentences which include specific variants of generalized phrases.

5 PREVIOUS WORK IN LANGUAGE ACQUISITION

Two previous models of language acquisition are related to our work. Granger's program FOUL-UP [Grangcr77], which acquired word meanings, and Pinker's model [Pinker84] which acquired language syntax.

FOUL-UP was devised as a mechanism for extending SAM's [Cullingford78] lexicon while performing conceptual analysis. SAM constructed meanings for English sentences by a two-step cycle: (a) select a script, and (b) instantiate the script. For example the following text:

> John's baby caught a cold. He called up his doctor, and made an appointment for the next morning. The nurse took his temperature, and it was 106 degrees. She realized immediately that he needed to be raggled.

involves the *Sclinic* script, which stands for a chain of events associated with a visit to the family doctor. Thus, the combination of child-being-sick followed by call-up-a-doctor causes the selection of Sclinic. Once this script is selected, references and events can be readily resolved. *Made an appointment* for example is taken as a standard Sclinic event. *The nurse* is bound to a designated role-holder in the script, and *take his temperature* also matches an anticipated action while in the clinic. The reference *his* (in *his temperature*) is bound to John's baby (and not to John) since the baby holds the role of the *patient* in Sclinic. FOUL-UP is activated by SAM at the point that the word *raggled* is encountered, a word which does not appear in SAM's lexicon. By knowing the rest of the possible events and the possible outcomes of Sclinic, FOUL-UP can predict that *to be raggled* means to taken to the hospital. FOUL-UP learned lexical entries assuming that the new single word appeared in the context of a script, and that the meaning of the word could be drawn from that script. FOUL-UP did not handle syntax acquisition, nor multi-word phrases.

Pinker [Pinker84] modeled child language acquisition, covering phenomena ranging from basic phrase structure identification to mastering phrase interaction-the issue being addressed here* Phrase interaction, or *complementation,* is restricted by two substantive constraints:

(a) The argument which might be missing in the complement is always the subject (e.g.: in *John persuaded Mary to leave*, the subject of the verb *leave* is not explicit in the text).

(b) The missing argument is *equated* with either one of:

  (1) an object of the controlling verb (*persuade* in the example above), if an object exists (which is the case in the example),

  (2) The subject of the controlling verb if the object does not exist (as in *John asked to go*).

This pattern of behavior holds for virtually all complement-taking English verbs. Nonetheless there are some exceptions such as *promise* in *John promised Mary to go*, or *make* in *John made Mary a fine husband* Such exceptions, although small in number, are significant since they could be used for testing the power of language-learning theories. If such exceptions did not exist, the learning of verb complementation would be trivial. Indeed this behavior (of *promise*) accounts for errors in children's language, as recorded by Carol Chomsky [Chomsky69] (as in *Jenny's father promised her to go to Disneyland* meaning *he promised her that she would go there*). At this point, an example in context (in which it is clear that *the father* is supposed to go) can correct such a hypothesis.

RINA draws from both these approaches. RINA's acquisition of lexical items concerns the mapping between syntactic patterns and their semantic concepts. However, Granger's model was restricted to single words and simple script-like contexts. RINA also addresses issues such as control and complementation. However, in our view, behavior of verbs reflects their semantics, and it is not just an arbitrary parameter which needs to be acquired to distinguish *promise* from *persuade*. Concepts representing such verbs must account for two facts. The verb means that (1) one party conveyed fact to a second party, and (2) each individual verb denotes a different speech act performed by conveying that fact. Thus learning the special syntax of *promise* is a by-product of understanding what *promise* means.

Pinker presupposes the existence of certain *innate* learning procedures. In his model, each language feature is accounted for by a custom-tailored procedure. For example, in learning the control aspect of complement-taking verbs, Pinker assumes this preexisting rule (Rule C3):

> Add to the lexical entry of the complement-taking predicate the equation X-COMP's SUBJ - (FUNCTION) where (FUNCTION) is the grammatical function annotated to the matrix argument that is coindexed with the missing complement subject in the contextually inferred semantic representation. ([Pinker84] pp. 213).

Pinker has chosen *Lexical-Functional Grammar* (LFG) [Bresnan82] as his linguistic framework. While LFG intends to denote a variety of linguistic phenomena, the rule C3 talks *about* LFG notation, but it certainly is not expounded in LFG terms. Moreover, if a language-learning program requires such a specific *procedural* rule for each language feature, then the purpose of modeling learning in the first place is defeated. Learning must be handled by a general *declarative* mechanism. In fact, the same *unification* mechanism which accounts for parsing and generation, should account also for learning.

6 CONCLUSIONS

Although the lexicon includes both specific and general phrases, the algorithm which constructs the hierarchy must require as input only specific episodes. This reflects human learning behavior. When communicating, people do not exchange syntax and semantics in form of rules, rather they communicate in terms of specific examples in context.

Accordingly we have shown a learning algorithm which augments a hierarchical lexicon as follows:

(1) A partial initial lexicon is given.

(2) The input is a sequence of specific examples.

(3) The result is an augmented lexicon.

The initial lexicon can be augmented by new specific prases. It can also be enhanced by generalizing and specializing existing entries. As a result of learning, a model can process sentences which could not parsed initially.

References

[Becker75] Becker, Joseph D., "The Phrasal Lexicon,'* pp. 70-73 in *Proceedings Interdisciplinary Workshop on Theoretical Issues in Natural Language Processing,* Cambridge, Massachusets (June 1975).

[Bresnan82] Bresnan, J. and R. Kaplan, "Lexical-Functional Grammar," in *The Mental Representation of Grammatical Relations,* ed. J. Bresnan, MIT Press, MA (1982).

[Chomsky69] Chomsky, C, *Acquisition of Syntax in Children from 5 to 10,* MIT Press, Cambridge MA (1969).

[Cullingford78] Cullingford, R. E., "Script Application: Computer Understanding of Newspaper Stories," 116, Yale University, Department of Computer Science, New Haven, Connecticut (1978).

[Fillmore87] Fillmore, C, P. Kay, and M. O'Connor, *Regularity and Idiomaticity in Grammatical Constructions: The Case of Let Alone,* UC Berkeley, Department of Linguistics (1987). Unpublished Manuscript.

[Granger77] Granger, R. H., "FOUL-UP: A Program That Figures Out Meanings of Words from Context," pp. 172-178 in *Proceedings Fifth IJCAI* , Cambridge, Massachusets (August 1977).

[Jacobs85] Jacobs, Paul S., "PHRED: A Generator for Natural Language Interfaces," UCB/CSD 85/198, Computer Science Division, University of California Berkeley, Berkeley, California (January 1985).

[Kolodner84] Kolodner, J. L., *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model,* Lawrence Erlbaum Associates, Hillsdale NJ (1984).

[Langacker86] Langacker, R. W., "An Introduction to Cognitive Grammar,'' *Cognitive Science* 10(1) (1986).

|Mitchell82] Mitchell, T. M., 4'Generalization as Search," *Artificial Intelligence* 18, pp.203-226 (1982).

[Pinker84] Pinker, S., *Language Learnability and Language Development,* Harvard University Press, Cambridge MA (1984).

[Schank77] Schank, R. and R. Abelson, *Scripts Plans Goals and Understanding,* Lawrence Erlbaum Associates, Hillsdale, New Jersey (1977).

[Schank82] Schank, R. C, *Dynamic Memory,* Cambridge University Press, Cambridge Britain (1982).

[Wilensky81] Wilensky, R., "A Knowledge-Based Approach to Natural Language Processing: A progress Report," in *Proceedings Seventh International Joint Conference on Artificial Intelligence,* Vancouver, Canada (1981).

[Wilensky83] Wilensky, R., *Planning and Understanding,* Addison-Wesley , Massachusetts (1983).

[Wilks75] Wilks, Y., "Preference Semantics," in *The Formal Semantics of Natural Language,* ed. E. Keenan, Cambridge, Cambridge Britain (1975).

[Zernik85a] Zernik, U. and M. O. Dyer, "Failure-Driven Aquisition of Figurative Phrases by Second Language Speakers," in Proceedings of the 7th Annual Conference of the Cognitive Science Society, Irvine CA (August 1985).

[Zernik85b] Zernik, U. and M. O. Dyer, 'Towards a Self-Extending Phrasal Lexicon," in Proceedings 23rd Annual Meeting of the Association for Computational Linguistics, Chicago 1L (July 1985).

[Zernik86a] Zernik, U. and M. O. Dyer, "Language Acquisition: Learning Phrases in Context," in Machine Learning: A Guide to Current Research, ed. T. Mitchell J. Carbonell R. Michalsky, Kluwer, Boston MA (1986).

[Zernik86b] Zernik, U. and M. O. Dyer, "Disambiguation and Acquisition through the Phrasal Lexicon," in Proceedings 11th International Conference on Computational Linguistics, Bonn Germany (1986).

[Zernik87a] Zernik, U. and M. G. Dyer, "The Self-Extending Phrasal Lexicon," The Journal of Computational Linguistics: Special Issue on the Lexicon (1987). to appear.

[Zernik87b] Zernik, U., "Strategies in Language Acquisition: Learning Phrases from Examples in Context," UCLA-AI-87-1, LA, CA (1987). Ph.D. Dissertation.