THE SELF-EXTENDING PHRASAL LEXICON*

Uri Zernik**
Michael G. Dyer

Computer Science Department
University of California
Los Angeles, California 90024

Lexical representation so far has not been extensively investigated in regard to *language acquisition*. Existing computational linguistic systems assume that text analysis and generation take place in conditions of complete lexical knowledge. That is, no unknown elements are encountered in processing text. It turns out however, that *productive* as well as *non-productive* word combinations require adequate consideration. Thus, assuming the existence of a complete lexicon at the outset is unrealistic, especially when considering such word combinations.

Three new problems regarding the structure and the contents of the phrasal lexicon arise when considering the need for dynamic acquisition. First, when an unknown element is encountered in text, information must be extracted in spite of the existence of an unknown. Thus, *generalized* lexical patterns must be employed in forming an initial hypothesis, in absence of more *specific* patterns. Second, senses of single words and particles must be utilized in forming new phrases. Thus the lexicon must contain information about single words, which can then supply clues for phrasal pattern analysis and application. Third, semantic clues must be used in forming new syntactic patterns. Thus, lexical entries must appropriately integrate syntax and semantics.

We have employed a Dynamic Hierarchical Phrasal Lexicon (DHPL) which has three features: (a) lexical entries are given as entire phrases and not as single words, (b) lexical entries are organized as a hierarchy by generality, and (c) there is not separate body of grammar rules: grammar is encoded within the lexical hierarchy. A language acquisition model, embodied by the program RINA, uses DHPL in acquiring new lexical entries from examples in context through a process of hypothesis formation and error correction. In this paper we show how the proposed lexicon supports language acquisition.

1. INTRODUCTION

Examination of the language acquisition task sheds light on the nature of the lexicon, illuminating issues which have been ignored by existing linguistic systems [Wilks75, Kay79, Bresnan82b, Gazdar85]. Current systems restrict their account to analysis and generation of text, by making the assumption that a fixed, complete lexicon exists at the outset. This assumption proves unrealistic for two reasons: First, due to the huge size of the lexicon (especially when including idioms and phrases) it is difficult to manually encode the entire lexicon. This problem is further aggravated as people continuously invent new idiosyncratic word combinations, which are then introduced into general speech. Second, word meanings must often be custom tailored to the domain (e.g., bug in computer applications), since people assign different meanings to words in various jargons. Therefore, computational linguistic models are required to learn lexical items in context, the way people learn new words and phrases.

Learning commonly occurs when the learner detects a gap in his or her knowledge. In analysis, such a discrepancy can be detected when a new word or phrase is encountered. Learning involves three issues: (a) detecting the discrepancy in the first place, (b) forming an initial hypothesis about the new phrase, and (c) refining and generalizing this hypothesis through a process of error correction [Granger77, Langley82, Selfridge82, Zernik85b]. These three issues impose new requirements on the lexicon, regarding (a) its

*This research was supported in part by a grant from the Initial Teaching Alphabet (ITA) Foundation.

**Uri Zernik's new address is: General Electric, Research and Development Center, P.O. Box 8 Schenectady NY 12301.

0362-613X/ 87 /030308-327$03.00

contents—the way individual entries are encoded, and (b) its structure—the way entries are organized.

The need to detect discrepancies affects the *contents* of the lexicon. Both semantic and syntactic discrepancies must be detected, and correction strategies must be associated with various types of errors. Thus, lexical entries should not be underspecified, lest they will allow discrepancies to slip by unnoticed.

The need to generalize affects the *structure* of the lexicon. In order to make an initial hypothesis about a new element, it is important to glean from the text as much information as possible. This requirement is problematic: the text cannot be analyzed since an element is unknown; but on the other hand, for the element to be acquired, the text must be analyzed. The solution for this bootstrapping problem is to employ a lexical hierarchy by generality. When a specific pattern does not exist for a precise matching against the new element, one can apply a more general pattern, which albeit being less informative, does match the new element.

Thus, we propose employing a Dynamic Hierarchical Phrasal Lexicon (DHPL) which has three features: (a) lexical entries are given as entire phrases and not as single words, (b) phrases are organized in a hierarchy by generality, and (c) there is not separate grammar; grammar is encoded in general lexical phrases. The program RINA [Zernik86b, Zernik87a] employs DHPL in modeling language acquisition. In particular, the program models second language acquisition of English phrases and idioms. The linguistic concepts being acquired are complex enough, so that neither a human learner, nor a computer program can acquire their complete behavior through a single example. Thus the initial hypothesis might be incorrect. Capturing incorrect hypotheses generated by humans, and simulating them by the computer program is essential for practical and theoretical reasons. First, the human user of the program will relate to the human-like errors generated by the program. Consequently he may present the program with constructive counterexamples. Second, human errors, such as errors of overgeneralization, reveal otherwise inaccessible cognitive processes and internal structures. Thus, errors made by human learners play a central role in constructing a cognitive model of acquisition. Subsequently, observed human behavior is analyzed in terms of the computer program RINA.

1.1 THE LINGUISTIC BEHAVIOR

RINA receives examples from a user who teaches her new phrases. When RINA encounters a new phrase, she creates a hypothesis about its behavior, and accordingly she generates an example to demonstrate her state of knowledge. Communication between the program and the user is only through a sequence of examples – there is no way to discuss syntax and semantics explicitly.

(1) LEARNING NEW PHRASES

In the following dialog, RINA encounters an unknown phrase, `throw the book at somebody`.

User: Al Capone went on trial.
The judge threw the book at him.
RINA: He threw a book at him?
User: No. The judge threw the book at him.
RINA: He punished him severely?

RINA is familiar with the single words `throw` and `book`. However, the entire figurative phrase is not in her lexicon. RINA first attempts a literal interpretation using a phrase existing in the lexicon (`throw an object`). When this interpretation fails she realizes the existence of an unknown, and tries to form the meaning of the new phrase by using (a) the context, and (b) the single words in the phrase.

(2) PROCESSING AN UNKNOWN

In the next dialog, RINA encounters a new word, `goggled`.

User: Jenny goggled John to come over.
RINA: Jenny told John that he must/can/will come to her.

RINA manages to extract useful information from the sentence in spite of the missing element. In particular, RINA's hypothesis includes three points:

(a) Jenny's unknown act is a kind of *mtrans** (an act of communication).
(b) The actor of the communicated event (`coming over`) is John.
(c) The communicated event is a conditional plan for the future (in contrast, for example, to the case: `Jenny goggled John that he came over`).

Normally, properties of an embedded phrase (e.g., `to come over`) are determined by the definition of the embedding verb. RINA manages to make an initial hypothesis even though the embedding verb (`goggle`) is unknown, by using generalized knowledge of phrase interaction. (The structure of a sentence: `Person1 goggled Person2 to do Act3` implies *mtrans* such as `ask`, `tell` and `instruct`, in contrast to `Person1 goggled to do Act2` which implies an *mbuild*, such as `decide`). The hypothesis must be abstract, since RINA cannot determine at this point whether this mtrans act comes in the sense of `allow` (can come over), or `instruct` (must come over). Yet, even this hypothesis may turn out to be incorrect. For example, `goggle` could mean seduce, or influence in some other way. In either, it is important to come up with a hypothesis which provides a basis for further modification.

*Conceptual classes such as mtrans, mbuild, select-plan, are based on semantic representation. Several of these elements are taken from Schank's [Schank77] system of primitive acts, goals and plans. Mtrans for example represents the transfer of mental information, and mbuild represents the construction of new information in memory. The particular scheme chosen is not so important as the fact that syntactic classes (such as verbs) are organized phrasally in terms of conceptual categories.

(3) RESOLVING AN AMBIGUITY

As with human listeners, computer parsers must also be able to interpret text successfully only when supplied the appropriate context [Zernik86a]. Consider the following sentence:

User: She took it up with her dad.
RINA: ?

Imagine a person hearing a fragment of a conversation between two unknown people, or alternatively, a computer program being given this sentence in isolation. Clearly, in the absence of a context, this sentence does not make complete sense. The pronouns, `she` and `it` cannot be resolved in absence of referents which have been introduced in the discourse. In addition, the same phrase will mean different things in different contexts. Consider these two examples.

User: Jenny wanted to buy a new car.
She took it up with her dad.
RINA: She discussed the issue with her dad.
User: Jenny started jogging.
She took it up with her dad.
RINA: She started an activity with him.

Since the same sentence can be interpreted in two ways in two different contexts, a question is raised regarding disambiguation. What is the impact of the context on phrase selection?

1.2 ISSUES IN LANGUAGE ACQUISITION

Three lexical representation issues must be addressed in modeling language acquisition.

(1) USING GENERALIZATIONS

As shown in the sentence below,

Jenny goggled John to come over.

the system must cope with unknown elements. Parts of the text must be examined to some extent, in spite of the presence of the unknown. Ideally, each element in the text is matched by a lexical phrase. Since no such phrase exists for a precise matching of the unknown element, a generalized phrase must be used to recover at least partial information. However, by the nature of generalization, the more generalized the matching phrase, the less informative it is.

Typical errors of overgeneralization were generated in a version of this paper by the first author, who is a second language speaker:

- The third phrase requires to generalize the initial notion. (Section 6)
- Wilensky suggested to represent knowledge as a database of rules. (Section 3.2)

In both cases, the learner applied the wrong generalized phrase, which accounts for verbs such as `decide` and `plan` (John decided to go home). This behavior does not capture verbs such as `suggest`, `require` or `tell` (*John told to go* sounds incorrect). The speaker faced a generation task in presence of incomplete lexical knowledge about `suggest` and `require`, and he resorted to using generalized knowledge. Using such knowledge, an idea could be communicated, albeit grammatically incorrectly.

Therefore, the lexicon must maintain phrases at various level of generality, to cope with different degrees of partial knowledge.

(2) USING LINGUISTIC CLUES

Meaning representation is extracted from the context. For example, given the text below,

Al Capone went on trial.
The judge threw the book at him.

RINA guessed that `throw the book at somebody` means to punish that person severely. However, the context might consist of many concepts, some appropriate and some inappropriate (e.g.: did the judge acquit Al or did he punish him?). Thus, a basic task is feature extraction. In extracting features, the system must utilize clues provided by single words. For example, what is the significance of the particle `at`? How does it contribute to the construction of the meaning? An experiment with second language speakers reveals, predictably, that using a different preposition leads to a different learning result. When the given text is:

Al Capone went on trial.
The judge threw the book **to** him.

language learners formed the hypothesis that the judge actually acquited the defendant. Thus, the lexicon must maintain senses for single words such as `at` and `to` that could be used as linguistic clues in feature extraction.

(3) USING SEMANTIC CLUES

The system must hypothesize the scope and variability of the new phrases. Which one of the phrases below best captures the syntax of the new phrase: the judge threw the book at him?

He threw **something** at him.
He threw **a book** at him.
He threw **the book** at him.

Each one of these patterns could be the specification of the new phrase. In determining degree of specificity the system must consult semantic clues extracted during parsing. For example, since no actual book exists in the context, then the reference the book is assumed to be a fixed literal. In contrast, consider the context below:

The judge was holding the third volume of tax law.
He threw the book at Al.

In this context, an instance of a book is found in the context (i.e., the third volume), and a different hypothesis is made about the the generality of the new pattern. Thus, semantic discrepancies in parsing must be utilized in determining both scope and generality of syntactic patterns.

2. Accounting For Idiomacity In The Lexicon

What are the contents of the lexicon to be acquired? Traditionally, the lexicon has been viewed as a list of words, specifying syntactic and semantic properties for each entry. However, since in our theory, the lexicon provides the sole linguistic database, it must include a variety of linguistic knowledge types, not just properties of single words. Here the lexicon is extended in two ways: towards the specific by bringing in idioms, and towards the general by including grammar.

2.1 IDIOMS AS EQUAL CITIZENS

Are idioms, such as `throw the book at`, a class apart, to be distinguished from "normal" phrases, which abide by grammar rules? The first to proclaim "equal rights" for idioms was Becker [Becker75], who called for a systematic treatment for the variety of phrases in the language. Consider these phrases:

> We **will be looking forward to** seeing you guys.
> He is cheap. He will not pay $5 **let alone $8.**
> **So much for** superficial solutions.
> Productive **as well as** non-productive phrases should reside in the lexicon.

These phrases defy traditional text-book grammar analysis, however, they possess their own grammar. For example, it sounds odd to say `He is cheap; He will not pay $8 let alone $5` [Fillmore87]. (Is the behavior of `as well as` analogous to the behavior of `let alone`?) Such linguistic phenomena cannot be ignored merely by tagging it as idiomatic, since idioms turn out to be ubiquitous in people's speech. Hardly can a sentence be found which behaves according to textbook grammar. There is a need therefore for a systematic treatment of idiosyncracy [Fillmore87]. Furthermore, linguistic knowledge cannot be strictly divided into grammar rules and lexical items. Rather, there is an entire range of items: some very specific, in the sense that they pertain to a small number of instances, and some very general, pertaining to a large number of instances. The former have been called "lexical items", and the latter "grammar rules". However, it is not possible to define a clear borderline between such two distinct groups, as elements could be found at all levels of generality, not just at the two ends of the spectrum. On one end, the phrase `it is raining cats and dogs` is very idiomatic. On other end, the phrase in `John took the spoon from Mary` is an instance of a general verb, `to take`, which may appear in many other ways. However, consider the phrase `John took the issue up with his dad`. Is this an idiom, or is it just an instance of the general verb `to take`?

2.2 PRODUCTIVE VS. NON-PRODUCTIVE PHRASES

In the *phrasal approach* [Wilensky84] rather than maintaining lexical entries for single words, the lexicon maintains entire phrases. For example, the lexicon will contain many phrases involving the word `throw`. Consider these phrases as they appear in the following sentences.

(1) He threw her off by a single inaccurate clue.
(2) He threw a wild party for her graduation.
(3) He threw up his whole breakfast.
(4) He threw his weight around.
(5) He threw a temper tantrum.
(6) He threw a stone at the kitchen window.
(7) He threw out that old chapter of his dissertation.
(8) He threw out the garbage.
(9) He threw the banana peel away.
(10) He threw in the towel.
(11) He threw the book at his students.
(12) He threw it. His answer was totally incorrect.

To a certain extent, all the phrases above derive their meanings from the meaning of the verb `to throw`. However, the issue here is whether a single generic lexical entry for `throw` can suffice to produce the meanings of all those sentences. In example (6) (he threw a stone), the phrase for `throw` is used in its generic form and meaning: to throw a physical object means to propel that object through the air. Sentence (9) (he threw away a banana peel) too can be interpreted using the generic phrase. In sentence (8) (he threw out the garbage), on the other hand, the derivation of the meaning using the generic phrase is less direct, as it requires analysis at the level of plans and goals. Throwing an object *causes* the object to become inaccessible. Thus throwing out the garbage does not necessarily mean throwing it in the air as much as getting rid of it.

The meanings of the other sentences are even more detached from the generic meaning. The meaning of *throw the book* (11) at is not a mere composition of the meanings of the single words, but requires extraneous knowledge from the trial situations. Neither a person, nor a computer program can produce the meaning of the phrase if the context is not given. Sentence (4) (he threw his weight around) introduces a metaphor [Lakoff80] in which a person's authority is compared to a weight, being used in a careless way. Sentence (2) (he threw a party) as well as sentence (5) (he threw a temper tantrum), use a different meaning of throw (to throw an event) which can hardly be related to its original meaning. Finally, sentence (12) (he threw it) represents a novel, yet still understandable, use of the word `throw` (as in `he blew it`).

Non-productive phrases are those in which the meaning of the entire phrase cannot be produced from the meanings of its constituents. Such phrases should be maintained in the lexicon as distinct entries. In fact,

even *productive* phrases, such as *to throw out the garbage*, should be maintained as distinct entries. Even if the meaning can be produced each time from the single words, an objective of an *efficient* system is to compile knowledge whenever possible, and to minimize unnecessary derivations. Thus, phrases in the lexicon can be viewed as *linguistic episodes* indexed and compiled for further use. Such knowledge is redundant in regard to language parsing (the meaning could be derived from the constituents again and again). However, this is not the case in language generation, where unless the phrase is stored, it is unlikely to be generated again by the system. Thus, both productive and non-productive phrases must be stored in the lexicon.

2.3 FIXED VS. VARIABLE PHRASES

As another example of lexical phrases, consider phrases involving the word `at`:

(13) John left school at noon.
(14) He actually stayed at school for an hour.
(15) He dabbled at the piano for a while.
(16) John aimed the ball at Mary.
(17) The criminal is still at large.
(18) Mary did not feel at ease in John's presence.
(19) This is what I am trying to get at.
(20) Did you understand anything at all?
(21) Please come at once!
(22) John looked at Mary.
(23) Fred lives at New-York. (produced by a second language speaker.)

Certain phrases are *fixed,* in the sense that they do not take any variation. For example, `at large`, `at all`, or `at once` are such fixed phrases. One cannot say, for example, `at twice`. However, other phrases might be mutated and still maintain their basic meaning. For example, `at noon`, `at midnight`, `at the hour`, etc., convey a meaning of sharp timing. Another meaning shared among a set of phrases is described by the following sentences:

(15) He dabbled at the piano for a while.
(24) He nibbled at the corn.
(25) He is playing at AI programming.

The use of the proposition `at` here implies an aimless, unfocused activity marking the difference between playing the piano and playing at the piano. Similarly, the set of sentences:

(22) John looked at Mary.
(26) Spot sniffed at Mary.
(27) Mary glanced at John.

share the implication that the sensory act was directed at the object.

Which ones of these phrases should be maintained in the lexicon? Fixed, idiosyncratic phrases such as `at large`, `at once`, and `at all` must be maintained in the lexicon. Otherwise they cannot be predicted by the system. However, the dilemma arises regarding variable phrases, such as in (22), (26) and (27). The question is whether to maintain all instances of a certain variable phrase or to maintain a single generalized entry which encompasses them all. We argue that *both* must be maintained. Specific phrases must be maintained as compiled, easy to access knowledge, while general phrases, which can derive many specific phrases, must be maintained too so that the system has a *predictive* power. Using such generalized phrases, the system can handle instances which have not been previously encountered.

In fact, specific "canned" phrases could not account for the following generation task, concerning the selection of appropriate prepositions in the following sentences:

(28) There is one teacher {in on at} our school, which I really like.
(29) I stayed late {in on at} school.

Notice that since both sentences involve the word `school`, it could not be used as a discriminator. Unless the lexicon maintains general predicates for the use of `in`, `at`, and `on`, the generator cannot select the appropriate preposition in each case. Clearly, it is difficult to capture the intuition of a native speaker in forming the general senses of these prepositions. An approximation of this intuition can be captured by modeling a second-language speaker who might "incorrectly" generate a sentence such as (23) above:

(23) Fred lives at New York.

Although it does not sound right to an English speaker, this sentence reflects the notion of that particular speaker.

2.4 OVERSPECIFICATION AND UNDERSPECIFICATION

Lexical entries should not be either underspecified or overspecified. Unless the lexical phrases are fully specified, they cannot serve in disambiguation. On the other hand, overspecification should also be avoided. Indeed, in encoding lexicons there is a temptation to overspecify. Consider the following pairs of examples in regard to lexical constraints:

He kicked the bucket.	The bucket was kicked.
Mary was taken by the car dealer.	The car dealer took her.
He put his foot down.	He put down his foot.
She laid down the law.	She laid the law down.
He took on Goliath.	He took on him.

There is a tendency to incorporate in the lexicon syntactic restrictions which will prevent the instances on the right. For example, `kick the bucket` would be marked as *active-voice-only*. This is in contrast to the phrase `bury the hatchet` which maintain its figurative flavor also in the passive voice: the hatchet was buried by Israel and Egypt.

We believe that this behavior is not dictated by an

arbitrary, ad hoc syntactic restriction, rather it reflects the conceptual representation of the phrase as it has been shaped in the acquisition process [Zernik87b]. The acquisition of the phrase `bury the hatchet` was based on a metaphor, and generalized from single-word meanings. `Bury` was generalized into *disenable-use*, and the referent `the hatchet` was generalized to a tool, the availability of which is a precondition for an active conflict. Therefore, the reference `the hatchet` stands for a certain generalized object. On the other hand, `kick the bucket` was learned as a whole chunk, since the underlying metaphor remained unresolved. Thus, the referent `the bucket` is maintained as a literal not associated with any concept. Due to this difference, there may arise a discourse function for passivizing `bury the hatchet`. However, since there is no referent for `the bucket`, there will never occur the need to passivize that phrase. Therefore, marking the phrase pattern as active-voice-only is redundant (albeit correct).

Another issue is verb-modifier separation, i.e.: David took on Goliath vs. He took him on. How can the lexicon account for this separation phenomenon? A grossly overspecified rule claims that pronouns (and only pronouns) separate such two-word verbs. However, there are counterexamples such as:

He took that ugly giant on.

(where the separation is by a lengthy reference). Therefore the rule must be revised to relate the phenomenon to *given* and *new* references. A given, or an already resolved reference, can separate, while a new reference cannot be placed between the verb andits modifier. We believe that this behavior should not be specified by the lexicon, rather the generation decision is according to discourse functions.

Overspecified lexical entries can always be contradicted by instances in context. In order to avoid the such contradictions we take the approach of maintaining syntactic specifications of lexical entries at appropriate levels, and use conceptual representation to account for apparently syntactic restrictions.

3. Lexical Representation: Previous Work

DHPL is a continuation of efforts in three distinct areas. First, in integrating the underlying situation as part of the lexical entry, we extend previous work on lexical presupposition. Second, we modify Wilensky's method of lexical representation for use in language acquisition. Third, we examine Bresnan's system of linguistic representation, which proves problematic in light of the acquisition task, and compare it to DHPL's representation.

3.1 Lexical Presupposition

A message might be conveyed by an utterance beyond its straightforward illocution. That message, called the *presupposition* of the utterance, is described by Keenan (1971) as follows*:

> The presuppositions of a sentence are those conditions that the world must meet in order for the sentence to make literal sense. Thus if some such condition is not met, for some sentence S, then either S makes no sense at all or else it is understood in some nonliteral way, for example as a joke or metaphor.

Despite this definition of presupposition as a *condition* for application of lexical knowledge, presupposition has been studied as a means for generation and propagation of inferences, reversing its role as a condition. In [Gazdar79, Karttunen79, Keenan71] the goal has been to compute the part of the sentence which is already *given*, by applying "backward" reasoning, i.e.: from the sentence the `king of France is bald` determine if indeed there is a king in France, or from the sentence `it was not John who broke the glass`, determine whether somebody indeed broke the glass. Rather than using presuppositions to develop further inferences, we investigate how presuppositions are actually applied according to Keenan's definition above, namely, in determining appropriate utterance interpretations.

Fillmore [Fillmore78] introduced lexical presupposition to describe situations in which lexical items may appear. He described the meanings of judgement words such as `accuse`, `criticize`, `blame`, and `praise`, by separating the entire meaning into (a) a statement (the *illocutionary act*), and (b) a presupposition. We illustrate this distinction by comparing the meanings of `criticize` and `accuse` in the following sentences:

(30) John criticized Mary for adjourning the meeting.
(31) John accused Mary of adjourning the meeting.

In both sentences, John referred to a hypothetical act, namely adjourning the meeting. In (30), it is **presupposed** that Mary committed the act (a test for determining presupposition is invariance under negation: John did not criticize Mary of adjourning the meeting still implies that Mary committed the act), while it is **stated** that the act is judged negatively. In (31), on the other hand, it is **stated** that Mary committed the act, while it is **presupposed** that the act is negative.

We believe Fillmore's approach is suitable also for the task of language acquisition, since learning involves factoring out the statement of a phrase from the entire surrounding context. We have further pursued Fillmore's notion in utilizing lexical presupposition in specific tasks such as disambiguation, indexing, and accounting for communicative goals [Gasser86a].

Presupposition must be distinguished from precondition. Consider the following text.

John ran into a pedestrian on a red light.
He managed to **explain it away** in court.

*(See also [Grice75] and [Fauconnier85] Ch. 3)

The lexical phrase under consideration is `explain away`. The *presupposition* for the *application of the phrase* is the entire situation in which the phrase typically appears. A person is attempting to justify a certain planning failure. The *precondition* for the *enablement of the act*, on the other hand, is a planning element from the domain itself. One precondition in the story above could be the judge's permission for John to stand up in court and defend his own case. Another trivial example is the sentence below.

John threw a rock at Mary.

There is no presupposition for the generic phrase `person throw phys-obj`. This phrase may appear in almost any context. However, from a planning point of view, for a person to throw a rock she must first grasp the rock in her hand. In contrast to presupposition, such planning information should not reside in the lexicon. In fact, any information which could be derived by means of general world knowledge does *not* belong in the lexicon.

Dyer [Dyer83] has described text comprehension as an integrated cognitive process. Parsing, he claimed, cannot be separated from other cognitive tasks such as memory update and retrieval. Accordingly, *search demons* were introduced in lexical entries to perform memory retrieval. For example, consider the difference between the two sentences.

(32) John made up his mind.
(33) He decided to go swimming.

In parsing sentence (33) the selected plan, namely going swimming, is mentioned explicitly. However, in sentence (32) neither the plan nor the problem to be resolved are mentioned explicitly. Therefore, a search demon associated with the phrase `make up one's mind` is dispatched to retrieve from memory the problem under consideration by the actor of the phrase. One of the objectives of DHPL's representation is to eliminate such procedural knowledge. Lexical presupposition serves the task of memory retrieval. The mechanisms we use are unification and variable binding.

3.2 LANGUAGE AS A KNOWLEDGE-BASED SYSTEM

Wilensky [Wilensky81] promoted the view of language processing as a knowledge-based task. Accordingly, he suggested representing linguistic knowledge as a database of rules given at various levels of generality. The basic representation element is called a *phrase*, given as a *pattern-concept* pair. For example, the phrase in the sentence:

John dropped out of police academy.

is given as the phrase

pattern ?x:person drop out of ?y:school
concept goal of person ?x, pursue-education at institute ?y, terminated unsuccessfully

Parsing is viewed as a process of rule (phrase) application. When more than one rule is applicable (ambiguity), selection is by *specificity*, namely, the most specific phrase is selected.

An additional layer was added to this work by Jacobs [Jacobs85] who noticed the need for inheritance and hierarchy in the lexicon. Concepts in memory are organized in a hierarchy of categories, through which more specific concepts can inherit features from more general ones. Concepts in the lexicon, namely lexical items, should be organized through the same general discipline. This approach enjoys three advantages:

- **Modularity:** Adding a new entry does not require any global modification.
- **Declarativeness:** The representation is neutral with respect to parsing and generation. The representation does not reflect any programming style (beyond basic slot-filler notation) and it does not reflect the mechanism of any particular parser.
- **Uniformity:** Modifying the level of generality of a phrase does not require a change of the phrase beyond the single feature being updated (generalized or specified).

These properties make the system more amenable to modeling language processing [Kay79] and acquisition [Mitchell82].

3.3 LFG AND LANGUAGE ACQUISITION

Bresnan's [Bresnan82a] linguistic representation, *lexical functional grammar* (LFG), is a system with a "flat" lexicon, which does not define a hierarchy of generalizations. LFG is contrasted here with DHPL's hierarchical approach, and it is examined here in regard to learning [Pinker84]. In LFG there are two lexical entries representing the word *ask*, as it appears in the following sentences.

(34) John asked to leave.
(35) John asked Mary to leave

The corresponding lexical entries are given respectively below.

```
ask: v:pred = "ask(subj, v-comp)"
     subj   = v-comp's subj              (subj-equi)

ask: v:pred = "ask(subj, obj, v-comp)"
     obj    = v-comp's subj              (obj-equi)
```

Figure 1: LFG representation of ASK

The meaning of *ask* is given as the predicate *ask* which takes either two or three arguments. There is no general notion which captures the similarities in the behavior of the two specific entries. In the hierarchical approach, on the other hand, the behavior of ask is described in the broader context of the infinitive interaction between phrases. The schematic hierarchy is given in Figure 2 below:

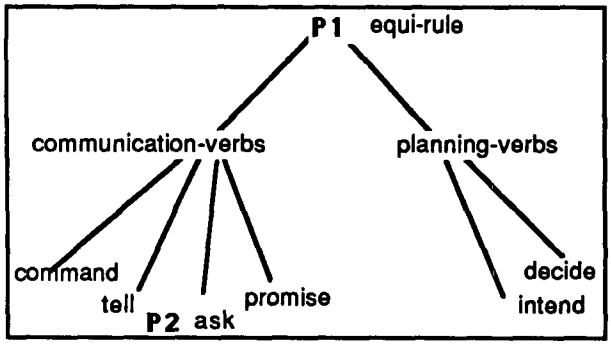

Figure 2: ASK as Part of a Broader Hierarchy

In this scheme, there is a single phrase for ask (P2). This phrase draws properties from a more general phrase (P1) which defines the general *equi rule* in complement-taking English verbs. In this representation, the behavior of ask is inherited from the general phrase P1 and there is no need to duplicate specific cases.

LFG current theory does not facilitate such hierarchies. In absence of hierarchy and inheritance, there is a need for duplication of the learning effort which can lead to serious flaws in modeling human behavior. For example, the word `promise` presents an exception to the general equi rule. Consider `John promised Mary to go`, in contrast to `John asked Mary to go`. The latter implies that John is the actor of the future act of going (John promised that **he** will go, but John asked that **Mary** go). In learning this behavior of promise, children make an error by hypothesizing the default equi rule, thus committing an error of *overgeneralization* (a child might say: Dad promised Tommy to drive the big car alone meaning "Tommy will drive the car"). In LFG it is impossible to model this behavior since generalizations do not exist. Indeed, Pinker [Pinker84] accounted for this error, but the equi rule he resorted to is not part of the LFG system itself. Moreover, through LFG it is impossible to *recover* from overgeneralization. Normally people recover from overgeneralizations by being given a counterexample (No. Dad promised Tommy to take him to Disneyland). However, since neither Bresnan nor Pinker attempt to represent meanings of words such as `take` and `drive` – the meanings are actually represented as the symbols "take" and "drive" – it is impossible to make the necessary semantic inferences for error recovery. Thus, without the ability to generalize and without an appropriate representation of concepts, LFG as currently defined, cannot account for these behaviors in learning.

4. Representing The Context

The semantics of entries in the lexicon draw from the various contexts in which they have been applied. Here we represent contexts using scripts, plans, goals, and relationships [Schank77, Dyer83, Dyer86b]. Consider the context in reading the text:

Al Capone went on trial.
The judge threw the book at him.

The underlying knowledge is the the *trial script,* which captures the basic events taking place in court.

(a) The Prosecutor communicates his arguments.
(b) The Defendant communicates his arguments.
(c) The Judge decides (select-plan) either:
  (1) Punish (thwart a goal of) Defendant.
  (2) Do not punish him.

Figure 3: The Acts in $Trial

This script, as shown in Figure 3, consists of a sequence of four events, in which the characters are the judge, the prosecutor, and a defendant. In addition, there is knowledge of the character's goals. The prosecutor is interested in thwarting a preservation goal – p-freedom, p-property of the defendant. The defendant attempts to block this goal thwart. Both parties advance their cases by trying to convince the judge. By this representation the meaning of the phrase to throw the book at somebody means to *punish him severely,* based on events (a) and (1) in the script.

Another situation, involving the same script, is presented in the following text.

John ran over a pedestrian.
He failed to **explain it away** in court,
and he went to jail

In this case the phrase explain away pertains to the underlying goal-plan situation, given in Figure 4 below. John experienced a planning-failure (failed plan of driving safely). John's preservation goal of freedom is threatened. A plan for preserving this goal is convincing the judge as to why John himself was not at fault. This second plan is executed and it fails also. Thus, his p-goal fails.

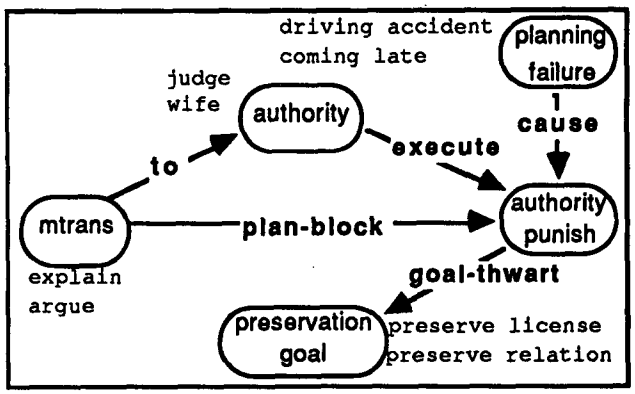

Figure 4: The Goal-Plan Structure for *explain away*

Notice that the same goal-plan schema exists also in the case of the next story:

> Joe forgot to put away the dirty dishes.
> When his wife came home, he **argued it away**
> by telling her he had been working.

The phrase `argue away` also involves a prior plan failure, a thwarted p-goal (*p-social-relation*) and a recovery plan of convincing the other party. This underlying schema is a presupposition. It holds whether Joe fails to argue it away or whether he manages to argue it away. Since the same plan-goal schema underlies both phrases (up to the specific plan: argue vs. explain), they both can be viewed as instances of a more general phrase.

Many other phrases draw their meanings in terms of such general plan-goal structures. Consider the phrases in the next sentences:

> This machine was **idling away** for hours.
> They stayed at home, and **argued away** for hours.
> The class was boring. John sat near the window
> **dreaming away.**

In all these sentences there is a similar underlying situation, shown in Figure 5 below.

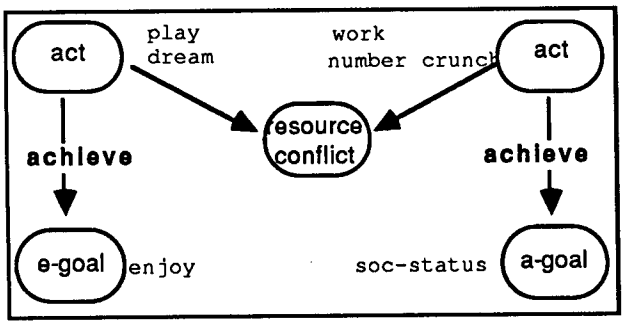

Figure 5: The Goal-Plan Structure for *idle away*

In this schema a resource competition (the resource is time) exists for an agent between two competing tasks, and that agent subordinates the important goal.

The fact that phrase representation can be elevated to a level of general plans and goals is very significant. It implies that *a relatively small number of structures can represent phrases whose instances can be used across many domains.*

5. Organizing The Lexicon

Retrieval and update are the operations required of memory [Kolodner84], and of the lexicon in particular. The objective in DHPL is to retrieve lexical entries at various levels of generality. The structure of the lexicon is specified by (a) the structure of a single lexical element, and (b) the global structure in which elements are organized.

5.1 Basic Phrase Structure

Consider the marked clause in the following text.

> For years they tried to prosecute Al Capone.
> Finally, **a judge threw the book at him**
> for income-tax evasion.

This clause is derived from a lexical phrase which is given as the following simplified template:

phrase pattern: Person1 throw the book at Person2.
presupposition: Person1 is an authority for Person2.
concept: Person1 punishes person2 severely.

This lexical *phrase* is a triple associating a linguistic *pattern* with its semantic *concept* and *presupposition*. The *pattern* specifies the syntactic appearance in text. The *presupposition* specifies the surrounding *context,* while the concept specifies the meaning added by the phrase itself. Phrase presupposition, distinguished from phrase concept, is introduced in DHPL's representation since it solves three problems: (a) in *disambiguation* it provides a discrimination condition for phrase selection, (b) in *acquisition* it allows the incorporation of the context of the example as part of the phrase, and (c) in *generation* it provides an indexing scheme for phrase discrimination and triggering.

The role of the three slots in a phrase template may be better understood by the way they are applied in parsing the text above. The clause is parsed in four steps:

(1) The pattern is matched successfully against the text. Consequently, Person1 and Person2 are bound to the judge and to Al Capone respectively (as the *person* class *restrictions* imposed by the pattern are satisfied).
(2) The presupposition associated with the pattern is validated using the concepts in the context. Using knowledge of human relationships, it is inferred that the judge presents an authority to Capone.
(3) Since both (1) and (2) are successful, then the pattern itself is instantiated, adding to the context: *The judge punished Al Capone severly.*
(4) Steps (1)-(3) are repeated for each relevant lexical entry. If more than one entry is instantiated, then the concept with the best match is selected.

(1) ACTUAL SLOT-FILLER NOTATION

The actual representation of the phrase is implemented using GATE's [Mueller87] slot-filler language, as shown below. In particular notice in that notation that the representation of a phrase, which is a linguistic object, is not different than the representation of other objects in the database.

```
comment X throw the book at Y
pattern   ?x throw ( the book ) ( at ?y)
presupposition
          (authority      high ?x
                          low ?y)
concept   (auth-punish    from ?x
                          to ?y)
```

Figure 6: The Phrase Notation

Notice that the phrase consists of three main parts: pattern, concept and presupposition (the comment is for reference only).

(2) CASE-FRAME REPRESENTATION

The pattern of the phrase above can be written as:

?x throw <the book> <at ?y>

This is an abbreviation which stands for the full notation given below.

```
subject (case-frame
                class person
                instance ?x)
verb    (verb
                root throw)
object1 (case-frame
                determiner the
                root book)
object2 (case-frame
                marker  at
                class   person
                instance ?y)
```

This full notation has three features:

(1) The pattern is constructed of four **case frames** [Carbonell84].

(2) Case frames are **named.** For example, *object2* is the name of the case frame given as:

```
marker    at
class     person
instance  ?y
```

This case is referred to as the lexical subject to be distinguished from the *surface subject* (the element actually preceding the verb in the text).

(3) Case frames are **unordered,** namely no order is imposed among the case frames. In no place in the case frame is it mentioned, for example, that the lexical subject should precede the verb or follow it (or not appear at all). Case ordering, thus, is inherited from general linguistic patterns, as shown later in this paper.

(4) Case frames contain both semantic and syntactic properties. For example, *object1* defines the named constituents the and book, while *object2* defines the class person.

Since not all properties are given explicitly within the pattern itself, there is a need for an *inheritance* scheme. Properties such as case order (e.g. active and passive voice), and word-order of the syntactic constituents within cases (e.g. the determiner the precedes the root book) are inherited from general linguistic patterns.

5.2 THE GLOBAL STRUCTURE

While varying in generality, lexical entries are represented uniformly throughout. The lexicon can be viewed as a collection of triples (Pattern-Concept-Presupposition), as shown in Figure 7, which are retrieved for parsing and for generation tasks, and become operational by unification.

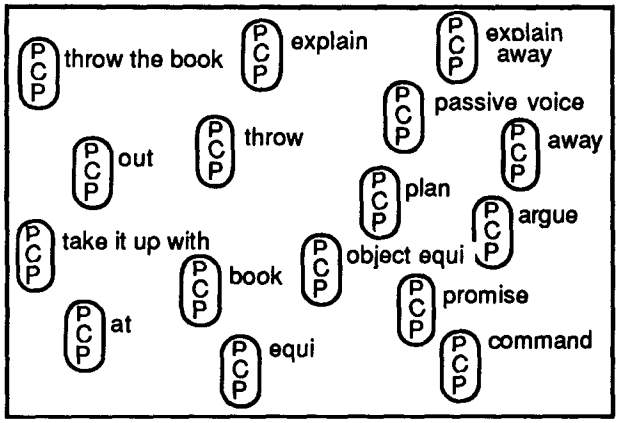

Figure 7: The Lexicon as a Collection of Triples

To facilitate learning, these triples are organized in hierarchies by generality. In a hierarchical scheme, the bottom nodes are very specific and idiomatic while the ones at the top are more general. Phrases may reside at, and inherit from, more than one hierarchy. For example, the phrase `to take on` can inherit from the hierarchy of `take` as well as from the hierarchy of `on` (a hierarchy which defines properties of verb modifiers). Four operations, implemented as forms of unification, and are defined by this representation. They are: (a) *interaction* between two unrelated phrases, (b) *inheritance* between two related phrases (one more general than the other), (c) *generalization,* and (d) *discrimination* of a phrase, which both update its level of generality. Three hierarchy schemes are given in the following sections to demonstrate three aspects of the system: (a) phrase interaction through the infinitive construction, (b) word-sense representation, and (c) case-order.

6. Representing The Infinitive

Consider the following pair of clauses in the sentences below:

Judge Wilson threw the book at him.
Judge Wilson decided to throw the book at him.

Parsing the first sentence is carried out simply as a lexicon lookup: a phrase is found in the lexicon, and its concept is instantiated. Parsing the second sentence is more complex since no single lexical phrase is matched for `throw`. For one thing, the subject does not precede the verb throw as anticipated by the lexical pattern. Identifying the implicit subject involves knowledge of phrase interaction. Properties of phrase interaction (through the infinitive form [Kiparsky71]) are represented by a hierarchy below.

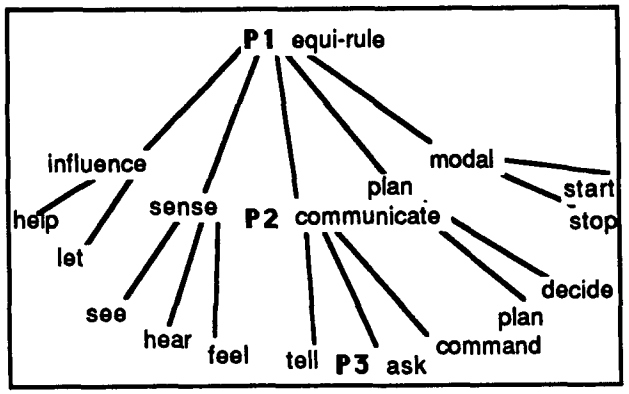

Figure 8: The Hierarchy for Phrase Interaction

The names of the individual nodes are mnemonic, and are used for reference only. Each such node is a full pattern-concept-presupposition triple (the presupposition may not appear). The nodes in Figure 8 are described as follows:

(a) The most general node (P1) denotes the basic equi rule, which stands for the following object:

```
comment     the general equi behavior
pattern
   (subject   instance ?x)
   (verb      root ?v)
   (object    instance ?y)
   (comp      pattern
                 subject   instance (and ?x ?y)
                 verb      form infinitive
              concept ?z)
concept
   (act       actor ?x
              object ?z)
```

In this phrase, notice in particular the *complement* (comp), which defines the embedded phrase. The implicit subject of the embedded phrase is taken as either (1) the object of the embedding phrase, if that object exists, or (2) the subject of the embedding phrase, if the object does not exist.

(b) Middle-level nodes encompass classes of verbs. For example, P2 encompasses *communication* verbs such as `ask`, `tell`, `instruct`, etc., share certain features. It is represented as follows:

```
comment     communnication verbs
pattern
   (subject   instance ?x)
   (verb      root ?v)
   (object    instance ?y)
   (comp      pattern
                 subject   instance (and ?x ?y)
                 verb      form infinitive
              concept ?z)
concept
   (mtrans    actor     ?x
              object    plan ?z)
```

This phrase is similar to the phrase P1. However, it includes information specific to that class of verbs. It defines shared syntactic features: subject, verb, object, complement (where the *complementizer is* to). It also defines shared semantic properties: (a) the equi-rule, (b) the concept of the complement, which is a hypothetical, future plan communicated by the actor.

(c) *Specific* nodes give the behavior of individual verbs, such as the phrases for decide (a *planning* verb) and command (a *communication* verb).

```
comment X decide to Z
pattern  (subject   instance ?x)
         (verb      root decide)
         (comp      pattern
                       subject   instance ?x
                       verb      form infinitive
                    concept ?z)
concept  (select-plan actor     ?x
                      object    plan ?z)
```

```
comment X command Y to Z
pattern  (subject   instance ?x)
         (verb      root command)
         (object    instance ?y)
         (comp      pattern
                       subject   instance ?y
                       verb      form infinitive
                    concept ?z)
presupposition
         (authority
                high ?x
                low ?x)
concept  (mtrans actor     ?x
                 to        ?y
                 object    (goal    instance ?z
                                    goal-of ?x))
```

Each one of these phrases adds on the information specific to the denoted verb. According to this representation ?x command ?y to ?z means that ?x

who presents an authority to ?y, tells ?y that ?z is a goal of ?x.

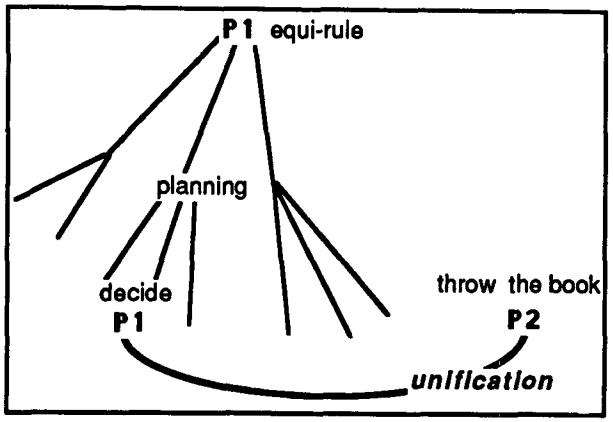

Figure 9: Interaction of Two Specific Phrases

(d) *Episodes* such as P4, which include specific instances of a phrase, are indexed to the phrase. For example, P4 is the situation in which God commands Moses to approach the Mountain. This episode contains the semantic ingredients constituting the meaning of the phrase.

The hierarchy of Figure 8 is used by four processing tasks.

6.1 PHRASE INTERACTION

The analysis of the sentence below:

Judge Wilson decided to throw the book at him.

involves the interaction of two specific phrases, as shown schematically in Figure 9. The two specific lexical phrases involved are the entries for `decide` (the embedding phrase, P1, elaborated in item (c) at the beginning of Section 6 above) and for `throw the book` (the embedded phrase, P2, described in Figure 6 above). The *unification* of these two phrases guarantees that: (a) the subject of P1 is the subject of P2, and (b) the concept of the P2 (denoted by ?z) is plugged in the plan slot of P1. The interaction of these two phrases yields the compound concept:

```
select-plan
        actor wilson.1
        plan  (auth-punish
                        actor wilson1
                        to    capone.2)
```

This concept conveys the meaning of the entire sentence.

6.2 PARSING AN UNKNOWN

In contrast to the previous example, consider the analysis of a sentence in which an unknown word is

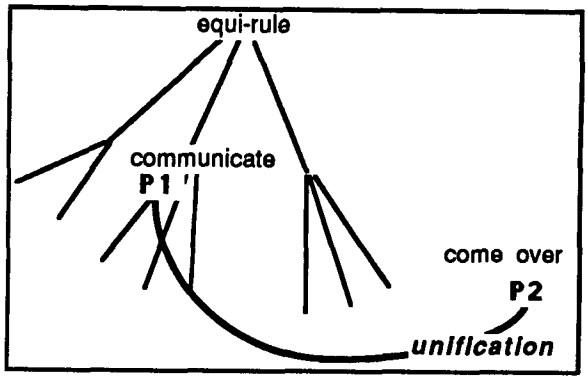

Figure 10: Interaction with a Generalized Phrase

included:

Mary goggled John to come over.

In analyzing this sentence, no lexical phrase is found to account for the word `goggle`. Therefore, the meaning of the entire sentence cannot be produced. Yet, even a partial meaning cannot be produced for the known clause, to come over, since it is intertwined with the unknown clause `Mary goggled John`. In order to overcome this obstacle, the interaction involves a more general phrase as shown in Figure 10. In contrast to Figure 9, here no specific phrase could be found for `goggle`, and it is necessary to select the generalized phrase, P1, which encompasses *communication* verbs in general. For come over, on the other hand, there exists a specific entry in the lexicon, P2, thus a generalization is not sought for. The *partial* meaning constructed for the sentence, in absence of a phrase for goggle is:

```
mtrans
        actor   mary.1
        to      john.2
        object  (ptrans
                        actor john.2
                        to    mary.1)
```

Thus, even when the particular phrase does not exist, the parser is able to construct an initial hypothesis, based on a generalization.

In fact, the selection of the generalized phrase is not unambiguous. The nature of the selected phrase is restricted by two schemes: (a) the hierarchy in Figure 8 above, and (b) the *persuade plan box* [Schank77] which provides the planning options available for a person in persuading another person to act (overpower, threaten, promise, steal, etc.). Accordingly, *goggle* could have as well conveyed meanings such as:

(36) Mary pushed John to come over. (influence verb)
(37) Mary let John come over. (help verb)
(38) Mary threatened John to come over. (promise verb)

Indeed option (38) is not available in English, however, since the phrase is yet unknown to the learner, this option must be given consideration.

6.3 OVERGENERALIZATION AND RECOVERY

In the case that the word promise does not exist in the lexicon, the program behaves as follows:

User: John promised Mary to come over.
RINA: John told Mary that she must/can come to him.

In using the generalized phrase, RINA unified inappropriately the roles. This is an error of overgeneralization which is typical of children learning new vocabulary items.

6.4 ERROR RECOVERY

The user can correct the program by giving an explicit example.

User: No. John promised Mary to come to her place.

By using few inferences (e.g., person ?x does not come to the same person ?x), RINA figures out the confusion in the role-binding and corrects appropriately the phrase for promise, as given below:

```
comment X promise Y to Z
pattern   (subject    instance ?x)
          (verb       root promise)
          (object     instance ?y)
          (comp       pattern
                        subject   instance ?x
                        verb      form infinitive
                      concept ?z)
presupposition
          (goal
                      goal-of ?y)
concept   (mtrans     actor   ?x
                      to      ?y
                      object  (plan  ?z))
```

Notice two interesting points regarding the semantics of promise: (a) ?x (the embedding subject) is always the subject of the embedded phrase, and (b) the act ?z is presupposed to be a goal of ?y. ?x is the subject of the embedded act, and the act ?z is presupposed to be a goal of ?y.

7. Handling Word Senses

By its nature, the phrasal approach is oriented towards the representation of entire groups of words. However, single words, such as up, at, and away must also be represented. Three issues are involved in representing such words.

7.1 ASSIGNING MEANINGS TO PARTICLES

Compare the following two sentences:

(39) John looked **up** at Mary.
(40) John looked at Mary.

The meanings of the two sentences are given below*:

```
(39)                          (40)
attend                        attend
    object    eyes                object    eyes
    actor     john.3              actor     john.3
    to        mary.4              to        mary.4
    direction vertical-positive
```

The contribution of the particle up is given as (**direction vertical-positive**). The role of the particle in the next sentence is less obvious.

(41) John **flew away** from the scene of the crime.

What is the contribution of the word away to the meaning of sentence (41)? For instance, how is the meaning of sentence (41) different than the meaning of sentence (42) below?

(42) John **flew** to Alaska.

7.2 RESOLVING WORD-SENSE AMBIGUITY

Is the contribution of away identical in all the sentences (43)-(46), or are there several meanings involved?

(43) John **flew away** from the scene of the crime.
(44) John did not **put away** the clean dishes.
(45) He managed to **argue it away** with his wife.
(46) This machine was **idling away** for hours.

For example, consider two appearances of the production argue away which involve two different senses of away:

(47) His lawyer can **argue away** any tax violation.
(48) He is a bum. He can **argue away** for hours without convincing anybody.

The first sense implies success in deceiving the authorities (as in get away with), while the second sense implies a waste of time (as in idle away). If there is more than one sense for away, then how is the appropriate meaning selected in each instance? In our lexicon, there are two phrases for argue away, which are disambiguated by matching their presuppositions with the context. The two phrases are:

*Another phrase, John looked up to Mary, in contrast to John looked up at Mary, is not processed as a simple production of the particles, since it involves the entire phrase "X look up to Y".

pattern ?x ⟨ ?v away ⟩ ?y
presupposition
?y is a planning failure by ?x
?g is ?x's goal thwarted by authority punishment ?z
?v is a communication act by ?x to avert ?z
concept
act ?v is successful, and ?z is averted

---

pattern ?x ⟨ ?v away ⟩
presupposition
act ?v serves no goal of ?x
act ?v consumes a useful resource (time)
concept
act ?v is selected by ?x

Figure 11: Two Different Senses for argue away

The appropriate phrase is selected in each context by matching the presupposition.

7.3 DETERMINING LEVEL OF GENERALITY

Which is the appropriate alternative for representing the phrase in sentence (49)?

(49) He managed to **argue it away** with his wife.

(a) Is it as "fixed" phrase as given below?

pattern: ?x away ?y ?z
concept: ?x managed to explain event ?y to person ?z by arguing.

(b) Or is it a "variable" phrase as given next:

pattern: ?x away ?y
concept: ?x managed to explain event ?y to person ?z by act ?v.

Answers for these dilemmas are given by the hierarchy in Figure 12 below:

(a) The most general phrase (P1) denotes the general properties of English verb modifiers. The modifier follows the verb, but separation is allowed (i.e.: `he explained it away` vs. `he explained away his latest goof`).

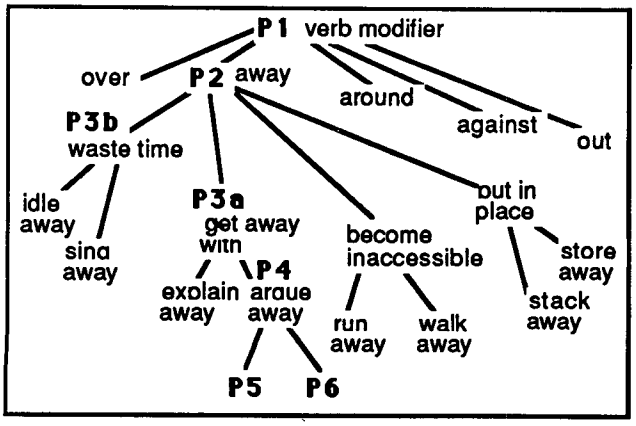

Figure 12: The Hierarchy for "Away"

ing conveyed by words such as `away` (P2), `up` and `down`. The pattern for P2, for example is <?v away> where ?v can be any verb.

(c) Nodes at the third level convey word senses which encompass classes of specific phrases. For example, P3a (*convince*) conveys the meaning encompassing both `explain it away` and `argue it away`, while P3b (*waste time*) conveys the meaning encompassing both `idle away` and `sing away`. These two phrases (P3a and P3b) are elaborated here:

pattern ?x ⟨ argue away ⟩ ?y
presupposition
?y is a planning failure by ?x
?g is ?x's goal thwarted by authority punishment ?z
?v (argue) is a communication act by ?x to avert ?z
concept
act ?v (arguing) is successful, and ?z is averted

---

pattern ?x ⟨ argue away ⟩
presupposition
act ?v (arguing) serves no goal of ?x
act ?v consumes a useful resource (time)
concept
act ?v (arguing) is selected by ?x

These two phrases generalize respectively the phrases in Figure 11.

(d) Nodes at the next level denote specific phrases, or *productions*, such as `run away`, `argue away` (P4), `idle away`, etc. Such phrases are given in Figure 11 for two cases of `argue away`.

(e) Nodes at the bottom level describe *episodes* in which instances of phrases were encountered (e.g., the instances `Al Capone argued it away in court` (P5), `John Smith argued it away with his wife` are indexed to the phrase <?x argue ?y away>).

On the face of it, it seems that levels (a) and (d) are sufficient for all parsing and generation purposes. What is the function of levels (b), (c), and (e)?

7.4 ANALYZING A NEW PRODUCTION

These intermediate levels of generalization facilitate the analysis of new productions such as:

(50) John tried to **describe it away** in court.

Sentence (50) introduces a new production to the reader of this paper. Yet, the reader should be able to resolve the new production by using the generalized linguistic pattern P3a in Figure 12.

7.5 LEARNING FROM EXAMPLES

In the previous example we have assumed an existing generalized phrase P3a, which was used in predicting a specific phrase. When such a *generality* does not exist, learning must be done by induction from *specific* examples. The following set of examples provide episodes

from which RINA can hypothesize the meaning of the phrase to `take on`.

(51) David took on Goliath.
(52) The *Celtics* took on the *Lakers*.
(53) Finally, I took on the hardest question on the midterm.

So far we have shown two ways of deriving new phrases: First, a new phrase can be *generalized* from indexed episodes (which include instances in context). However, learning is easier when a generalized template already exists, in which case learning is accomplished by applying a generality [Zernik85a].

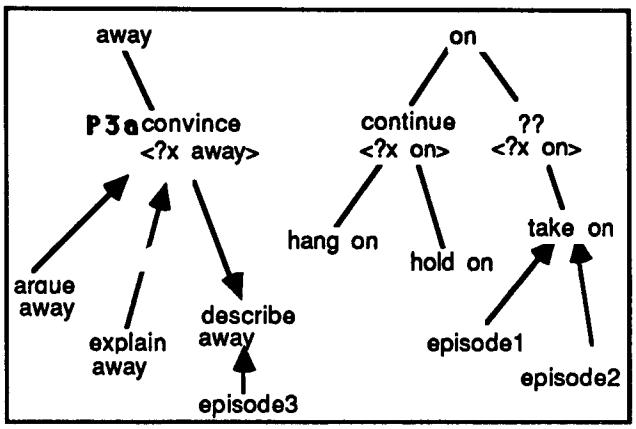

Figure 13: Top-Down vs. Bottom-Up Propagation

Figure 13 shows two learning processes: `describe it away` is deduced top-down from an existing general concept (P3a). On the other hand, `take on` is induced bottom-up from the set of specific episodes such as David and Goliath, the *Celtics* vs. the *Lakers*, and the midterm. There is no generalized concept which could serve as a short cut.

8. Inheriting Case Order

Consider the lexical pattern given as a set of four unordered case-frames:

PO: ?y throw < book> <at ?x>

Since ordering is not specified explicitly in pattern P0, then how can this pattern match sentences such as:

(54) The judge threw the book at Al. (active voice)
(55) The book was thrown at him. (passive voice)
(56) Al he decided to throw the book at, but John he gave a break. (left dislocation)
(57) "Take it easy!" said the prosecutor. (right dislocation)

Under what condition does the *lexical subject* precede the verb, and when can the lexical subject be omitted altogether? This information is contained in a case-order hierarchy (Figure 14 below) in the lexicon.

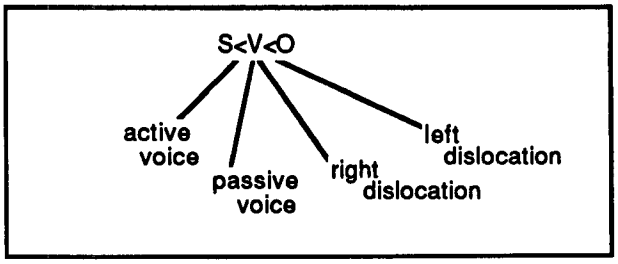

Figure 14: Case-Order Hierarchy

The patterns for the *passive* and the *active* voice, for example, are given in the figure below.

P2:
subject (location bef) (marker none)
verb (location ref) (voice active)
object1 (location aft)
object2 (location aft)

P3:
subject (location any)
verb (location ref) (voice passive)
object1 (location bef) (marker none)
object2 (location aft)

In matching sentences (54) and (55) above, the pattern P0 inherits case-order properties from these general linguistic patterns. For example, after inheriting the passive voice for matching sentence (55), the pattern augmented by inheritance from P3 would be:

P1:
subject (location aft) (marker by) (class person)
verb (location ref) (voice passive) (root throw)
object1 (location bef) (marker none) (root book)
object2 (location aft) (marker at) (class person)

An even more general pattern exists which captures the basic SVO structure of the language. This phrase is given at the top of the hierarchy:

P0:
pattern
subject (location bef) (marker none) (instance ?x)
verb (location ref)
object1 (location aft) (marker none) (instance ?y)
object2 (location aft) (marker ?m) (instance ?z)
concept
act (actor ?x) (recepient ?y) (?m ?z)

What is the use of that general SVO phrase? This phrase is called for in absence of more specific knowledge. Children who have not yet mastered specific case-structure patterns resort to this pattern. For example, a 2-year-old child might incorrectly understand:

Mary was fed by John.

as if Mary actually fed John. Adults too, in case of missing knowledge, might resort to this generality in making sense out of sentences.

9. FIGURATIVE PHRASE ACQUISITION: A PROCESS MODEL

So far, we have assumed the existence of necessary phrases in the lexicon. However, in reality a program may encounter new phrases in the text. Thus, the program must accomplish two objectives: (a) parse the text in spite of the unknown element, and (b) acquire information about the unknown element for future encounters. Consider the situation in which the figurative phrase is first encountered.

User: The mobster eluded prosecution for years.
Last month, they threw the book at him for income-tax evasion.
RINA: The prosecutor propelled a book at him?
User: No. A judge threw the book at him.
RINA: The judge threw the book at him. He found him guilty.

And later on:

User: The dean of the school threw the book at John.
RINA: He punished him.

There are three stages in the acquisition process:

(1) Apply the literal interpretation.
(2) Acquire the figurative phrase.
(3) Generalize the new phrase beyond the specific context.

9.1 LITERAL INTERPRETATION

In the absence of the appropriate phrase in the lexicon, RINA utilizes other available knowledge sources, namely (a) the literal interpretation and (b) the context. The literal interpretation is given by the phrase:

```
pattern   ?x:person throw ?y:phys-obj ⟨ at ?y ⟩
concept   propel    actor  ?x
                    object ?y
                    to     (location-of ?z)
```

Figure 15: Propel a Phys-Obj

This phrase describes propelling an object in order to hit another person. Notice that no presupposition is specified. General phrases such as `take`, `give`, `catch`, and `throw` do not have a expressed presupposition since they can be applied in many situations.* The literal interpretation fails by plan/goal analysis. In the context laid down by the first phrase (prosecution has active-goal to punish the criminal), "propelling a book" does not serve the prosecution's goals. In spite of the discrepancy, RINA spells out that interpretation above with a question mark, `The prosecutor propelled a book at him?` to notify the user about her current state of knowledge, and the fact that a discrepancy has been detected.

9.2 LEARNING BY FEATURE EXTRACTION

In constructing the new hypothesis, the program must extract the relevant features from the given episode.

(a) The initial phrase *presupposition* is taken to be the entire *trial* script.
(b) The *pattern* is extracted from the sample sentence.
(c) The *concept* is extracted from the script.

In extracting either the pattern or the concept, the problem is to distinguish between features which are relevant and should be taken in as part of the phrase, and features which are irrelevant and thus should be left out. Moreover, some features should be taken as is, where other features must be abstracted before they can be incorporated.

9.3 FORMING THE PATTERN

Four rules are used in extracting the linguistic pattern from the sentence:

Last month, they threw the book at him
for income-tax evasion.

(1) Initially, use an existing literal pattern. In this case, the initial pattern is:

---

pattern1:
?x:person throw: ?z:phys-obj <at ?y:person>

---

(2) Examine other cases in the sample sentence, and include cases in the pattern which could not be interpreted by general interpretation. There are two such cases:

(a) Last month could be interpreted as a general time adverb (i.e.: last year he was still enrolled at UCLA, the vacation started last week, etc.).

(b) For income-tax evasion can be interpreted as a *element-paid-for* adverb (i.e.: he paid dearly **for his crime,** he was sentenced **for a murder he** did not commit, etc.).

Thus, both these cases are excluded.

(3) Variablize references which can be instantiated in the context. In this case ?x is the Judge and ?y is the Defendant. They are maintained as variables, as opposed to case (4):

(4) Freeze references which cannot be instantiated in

* Notice the distinction between preconditions and presupposition. While a precondition for "throwing a ball" is "first holding it", this is not part of the phrase presupposition. Conditions which are implied by common sense or world knowledge do not belong in the lexicon.

the context. Since no referent is found for the reference the book, that reference is taken as a frozen part of the pattern instead of the case ?z:phys-obj.

The resulting pattern is:

---

pattern2:
?x:person throw: <the book> <at ?y:person>

9.4 FORMING THE CONCEPT

In selecting the concept of the phrase, there are four possibilities, namely the events shown in Figure 3 (Section 4). The choice of the appropriate one among these four events is facilitated by linguistic clues. As opposed to the phrase `they threw the book to him` which implies cooperation between the characters, the phrase `they threw the book at him` implies a goal conflict between the characters. `At` implies not taking acknowledgement protocols into consideration. E.g., `x throws the rock to y` implies that x catches y's attention, and gets acknowledgement for y's receipt of the rock. On the other hand, `x throws the rock at y` implies that y may not be aware or ready to receive the rock. This analysis applies also to `talk at` vs. `talk to`, etc. Since this property is shared among many verbs, it is encoded in the lexicon as a *general* phrase:

```
pattern   ?x:person ?v:verb ?y:phys-obj ⟨ at ?y ⟩
concept   propel    actor   ?x
                    object  ?y
                    to      (location-of ?z)
                    mode    no-acknowledge
```

Figure 16: Propel At, a General Phrase

Notice that rather than having a specific root, the pattern of this phrase leaves out the root of the verb as a variable. From lack of acknowledgement, a goal conflict may be inferred.

```
goal
     class    p-health
     status   thwarted
     goal-of  ?z
```

Using this concept as a search pattern, the "punishment-decision" is selected from *$trial*. Thus, the phrase acquired so far is:

```
pattern   ?x:person throw ⟨ the book ⟩ ⟨ at ?y ⟩
concept   auth-punish     actor    ?x
                          to       ?y
presupposition
          trial
                          judge    ?x
                          defendant ?y
```

Figure 17: The Acquired Phrase

9.5 PHRASE GENERALIZATION

Although RINA has acquired the phrase in a specific context, she might hear the phrase in a different context. She should be able to transfer the phrase across specific contexts by generalization. RINA generalizes phrase meanings by analogical mapping. Thus, when hearing the sentence below, an analogy is found between the two contexts.

> The third time he caught John cheating in an exam, the professor threw the book at him.

The trial-script is indexed to a general *authority* relationship. The actions in a trial are explained by the existence of that relationship. For example, by saying something to the Judge, the Defendant does not dictate the outcome of the situation. He merely informs the Judge with some facts in order to influence the verdict. On the other hand, by his decision, the Judge does determine the outcome of the situation since he presents an authority. Three similarities are found between the $trial and the scene involving John and the professor.

(a) The authority relationship between ?x and ?y.
(b) A law-violation by ?y.
(c) A decision by ?x.

Therefore, the phrase presupposition is generalized from the specific trial-script into the general *authority-decree* situation which encompasses both examples.

10. Current Status And Limitations

The lexical theory (DHPL) described in this paper underlies the program RINA described in the first author's dissertation [Zernik87c]. The program RINA is currently implemented in T [Rees84] (a dialect of SCHEME), on an APOLLO workstation using GATE's [Mueller87] unification language. RINA's lexicon includes more than 200 phrases including grammatic forms, word senses, and idioms. A "micro" version of the program, which carries out basic parsing and learning functions is included as an appendix of the dissertation [Zernik87c]. RINA can engage in learning sessions by using a variety of learning strategies. However, RINA's performance is limited in four ways.

(1) Only fragments of the English grammar have been implemented in the current version of DHPL. More work is required for extending systematically the set of constructs handled by DHPL.
(2) Text generation has not been at the focus of our work, and RINA's generation capabilities need to be enhanced.
(3) Aspects of discourse and dialog have been accounted for by simple heuristics. The input/output dialogs of the program are restricted to a small number of examples.
(4) *Transfer* plays a major role in second-language acqisition, as elements in language I are manifested

in performing in language II. Accordingly, multiple languages in one model must not be represented in isolation. The aspect of transfer and cross-linguistic interference is modeled by Michael Gasser [Gasser86b].

11. Future Work

Open research issues are (a) generation of examples, (b) learning and forgetting, (c) concept generalization, and (d) handling irony.

11.1 GENERATION OF EXAMPLES

We have identified a difference between generation tasks in general, where the generator describes a state of affairs in the world, and our specific task of *example generation*. In example generation, the program is required to demonstrate its own state of knowledge. For instance, one dialog given earlier proceeds as follows:

User: Greg wanted to buy a new car.
He took it up with his dad.
RINA: He took up **the car** with his dad?

The explicit reference `the car` is important since it conveys RINA's failure in acquiring the phrase. How could a program decide to generate the car (and not it) in contrast to he (and not Greg)? The research issue is: how a program or a person can test out its notion of a phrase. Examples must be generated to examine the boundary conditions in which the phrase can still be applied. This issue has not been investigated so far.

11.2 LEARNING AND FORGETTING

Two related issues are system *stability* and *obsolescence*, or forgetting. Stability concerns the ease with which well-established knowledge can be modified. If the behavior of the program is too dynamic, then it might easily get thrown off by one esoteric, or incorrect use of a phrase. It is not desirable that an adult native speaker would get his lexicon ruined by listening to a second language speaker. Forgetting involves inaccessibility of unused phrases, or getting rid of incorrect hypotheses. Are incorrect hypotheses simply destroyed, or is there a more realistic model of obsolescence? These two issues involve quantitative reasoning which require implementation of strength of links and activation. These kind of problems demonstrate the limitations of a strictly *qualitative* approach, such as ours, which rely on manipulation of logical propositions, and it raises the need for quantitative approaches such as *connectionism* [Waltz85, McClelland86], and *spreading activation* [Anderson84, Charniak83].

11.3 CONCEPT GENERALIZATION

Proliferation of knowledge is the process we try to approximate. The ubiquitous dilemma in comparing two concepts is whether a generalization exists for both, or whether they are distinct concepts. For example, consider the following sequence of examples in teaching the phrase to `take on`.

(57) David took on Goliath.
(58) I took on my elder brother.
(59) I took on a new job.
(60) We took on a new systems programmer.
(61) This piece of paper took on the shape of a butterfly.

The second phrase can share the concept acquired for the first one, namely ?x decided to fight ?y. The third phrase; however, requires one to generalize the initial notion since it now appears as ?x accepted a challenge presented by ?y. However, can a generalization be found to encompass the fourth phrase? Notice that although a very general concept which encompasses all of the given examples *could* be found (?x has something to do with ?y), however, the effectiveness of such a generalized notion is totally diminished. Therefore, a shared concept should be sought at the appropriate level of generality.

11.4 DEVIATIONAL USES OF LANGUAGE

So far, the notion of lexical presupposition has not been developed according to its agreed functional definition. It is agreed that lexical presupposition presents *felicity* conditions for phrase application. When these conditions are violated, phrases sound awkward, ironic, or simply incorrect. Consider the sentences below:

(62) We refused to let our baby stay up all night, so he **threw the book** at us. He yelled and screamed for hours.
(63) My pals asked me how I got straight A's. I managed to **explain it away** by telling them it was a bureaucratic mistake.

In each one of these sentences, a lexical presupposition is being violated. Our baby, as we all know, is not really an authority, as required of the actor of the phrase throw the book. Therefore, Sentence (62) sounds ironic. A presuppositional condition is violated also in sentence (63). The entire presupposition states: (a) a planning failure by the actor, (b) a threatening act by a social authority, and (c) an explanation act taken to block that punishment. Now, getting A's is not a planning failure, rather it is a fortuitous success, which makes the situation humorous. Consider the next pair of sentences:

(64) I made an appointment with my advisor. I **met** him on time.
(65) I made an appointment with my advisor. **I ran into** him on time.

Both `run into` and `meet` make the same statement: two characters got into a physical proximity. However, since run into presupposes an unplanned, surprising element which does not exist in the situation, sentence (65) sounds incorrect. In contrast to previous research in which presupposition was used for deriving secondary inferences which are mostly redundant, we suggest using presuppositions for disambiguation, detection of

irony [Dyer86a], and even for generation of irony by a computer (by applying phrases in situations where a presuppositional condition has been slightly mutated).

12. Conclusions

We have shown how the Dynamic Hierarchical Phrasal Lexicon (DHPL) supports language analysis, and language acquisition. We accounted for a dynamic language behavior by promoting four aspects of lexical representation:

Phrases: The lexicon contains entire phrases, accounting uniformly for an entire *range* including productive as well as non-productive phrase.

Hierarchy: The lexicon organizes in a hierarchy, phrases ranging from specific "lexical entries" at the bottom, to general "grammar rules" at the top.

Lexical Presupposition: Contextual conditions are incorporated into the lexicon through lexical presuppositions. Presuppositions account for disambiguation in parsing, and for phrase selection in generation.

Integration of Syntax and Semantics: Phrases specify a *relation* (in the logical sense) between syntax and semantics. Thus, the question whether any lexical feature is syntax or whether it is semantics, becomes insignificant. For example, consider thematic roles for a phrase such as promise (Section 6.4). Are they syntactic or are they semantic? They can be viewed as either.

Using this representation we have shown three results in language processing:

Coping with Lexical Gaps: The hierarchical structure of the lexicon enables parsing of text even when certain lexical elements are unknown. A partial meaning for the text, which serves as an initial hypothesis, is formed by applying *general* knowledge when *specific* knowledge is missing.

Using Lexical Clues: In learning meanings of phrases we have used "linguistic clues". For instance, the word at in the judge threw the book at Al, supports the learning process of that idiom. What is the justification for drawing inferences from apparently vague senses of words? In making the lexicon amenable as a linguistic database, from which inference rules can be drawn, we have systematically organized words in a hierarchy, representing words such as at, to, around and away. Thus, the use of linguistic clues per se is not inappropriate; however, all linguistic clues used in a reasoning system, must be drawn from a well-organized lexicon.

Knowledge Propagation through Generalization and Specialization: Hierarchy is a precondition for learning by generalization. Through the hierarchical scheme, there are two ways of propagating knowledge: First, *bottom-up*-from instantiated episodes up towards specific phrases, and even higher to generalized word senses. Second, *top-down*–generalized word senses are propagated down for prediction of new specific phrases. In both cases, effective learning depends on the existence of a well refined hierarchy. Any linguistic system must accommodate not only for spanning a static language, but also for augmenting the original linguistic system itself. In DHPL we have shown how, for a variety of linguistic features, the lexicon itself can be augmented through linguistic experiences. Thus we have accomplished a dynamic linguistic behavior.

Acknowledgement

The authors are indebted to Erik Mueller and Mike Gasser for help in developing the ideas in this paper. We also thank numerous second language speakers who inadvertently contributed interesting errors.

References

Anderson, John R. 1984 *The Architecture of the Mind.* Harvard University Press: Cambridge, Mass

Becker, Joseph D. 1975 The Phrasal Lexicon. In *Proceedings Interdisciplinary Workshop on Theoretical Issues in Natural Language Processing*. Cambridge, Massachusets June 70-73.

Bresnan, J. 1982 Control and Complementation. In J. Bresnan, *The Mental Representation of Grammatical Relations*. Cambridge, MA: The MIT Press. (a)

Bresnan, J.; R. Kaplan; J. Bresnan. 1982 Lexical-Functional Grammar. In *The Mental Representation of Grammatical Relations* MIT Press, Cambridge MA (b)

Carbonell, J. G.; P. J. Hayes. 1984 Coping with Extragrammaticality. *Proceedings Coling84*. Stanford, California 437-443.

Charniak, E. Passing Markers: A Theory of Contextual Influence in Language Comprehension. *Cognitive Science* 7 3 1983

Dyer, M.; M. Flowers; J. Reeves. 1986 A Computer Model of Irony Recognition in Narrative Understanding. *Advances in Computing and the Humanities* 1 1 (a)

Dyer, M. G. 1983 *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension.* MIT Press, Cambridge, MA

Dyer, M. G.; U. Zernik. 1986 Encoding and Acquiring Figurative Phrases in the Phrasal Lexicon. *Proceedings 24th Annual Meeting of the Association for Computational Linguistics*, New York NY (b)

Fauconnier, Gilles. 1985 *Mental Spaces: Aspects of Meaning Construction in Natural Language.* MIT Press, Cambridge MA

Fillmore, C. J. 1978 On the Organization of Semantics Information in the Lexicon. *Proceedings Chicago Linguistic* Society

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

Gasser, M. 1986 Memory Organization in the Bilingual/Second Language Learner: A Computational Approach. *Proceedings Eastern States Conference on Linguistics* (ESCOL). Chicago, IL (a)

Gasser, M.; M. G. Dyer. 1986 Speak of the Devil: Representing Deictic and Speech Act Knowledge in an Integrated Lexical Memory. *Proceedings 8th Conference of the Cognitive Science Society*. Amherst, MA, August 1986 (b)

Gazdar, Gerold. 1979 A Solution to the Projection Problem. In Choon-Kyu Oh, David A. Dinneen, *Syntax and Semantics (Volume 11: Presupposition)*. New-York, Academic Press 57-87

Gazdar, G.; E. Klein; G. Pullum; I. Sag. 1985 *Generalized Phrase Structure Grammar.* Harvard University Press, Cambridge, MA

Granger, R. H. 1977 FOUL-UP: A Program That Figures Out Meanings of Words from Context. *Proceedings Fifth IJCAI.* Cambridge, Massachusets, August 172-178

Grice, H. P. 1975 Logic and Conversation. In P. Cole, J. Morgan, *Syntax and Semantics (Volume 3: Speech Acts).* NY Academic Press

Jacobs, P. S. 1985 A Knowledge-Based Approach to Language Production. UC Berkeley, Computer Science Division, UCB/CSD 86/254, Berkeley, CA, August Ph.D. Dissertation

Karttunen, L.; S. Peter. 1979 Conventional Implicature. In C. K. Oh, D. Dinneen, *Syntax and Semantics (Volume 11, Presupposition).* NY Academic Press

Kay, Martin. 1979 Functional Grammar. *Proceedings 5th Annual Meeting of the Berkeley Linguistic Society*, Berkeley, California 142-158

Keenan, L. Edward. 1971 Two Kinds of Presupposition in Natural Language. In Charles Fillmore, D. T. Langendoen, *Studies in Linguistic Semantics*. New York, Holt, Reinhart and Winston, 44-52

Kiparsky, P.; C. Kiparsky. 1971 Fact. In D. Steinberg, L. Jakobovits, *Semantics, an Interdisciplinary Reader*. Cambridge, England, Cambridge University Press

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

Lakoff, George; Mark Johnson. 1980 *Metaphors We Live By*. The University of Chicago Press, Chicago and London

Langley, Pat. 1982 Language Acquisition Through Error Recovery. Cognition and Brain Theory 5 3 211-255

McClelland, J. L.; D. E. Rumelhart. 1986 Parallel Distributed Processing. MIT Press, Cambridge, MA

Mitchell, T. M. 1982 Generalization as Search. *Artificial Intelligence* 18 203-226

Mueller, Erik T. 1987 GATE Reference Manual (Second Edition) UCLA, Computer Science Department UCLA-AI-87-6 Los Angeles, CA

Pinker, S. 1984 *Language Learnability and Language Development*. Harvard University Press, Cambridge, MA

Rees, Jonathan; Norman Adams; James Meehan. 1984 *The T Manual*. Computer Science Department, Yale University, New Haven CT

Schank, R.; R. Abelson. 1977 *Scripts, Plans, Goals, and Understanding*. Lawrence Erlbaum Associates, Hillsdale, New Jersey

Selfridge, Malory. 1982 Why Do Children Misunderstand Reversible Passives? The CHILD Program Learns to Understand Passive Sentences. *Proceedings AAAI-82*. Pittsburgh, Pennsylvania, August 251-257

Waltz, D. L.; J. B. Pollack. 1985 Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation. *Cognitive Science* 9 1

Wilensky, R.; Y. Arens; D. Chin. 1984 Talking to UNIX in English: an Overview of UC. *Communications of the ACM* 27 6 June 574-593

Wilensky, R. 1981 A Knowledge-Based Approach to Natural Language Processing: A Progress Report. *Proceedings Seventh International Joint Conference on Artificial Intelligence*, Vancouver, Canada

Wilks, Y. 1975 Preference Semantics. In E. Keenan, *The Formal Semantics of Natural Language*. Cambridge, Britain

Zernik, U.; M. G. Dyer. 1985 Failure-Driven Aquisition of Figurative Phrases by Second Language Speakers. *Proceedings of the 7th Annual Conference of the Cognitive Science Society*. Irvine, CA (a)

Zernik, U.; M. G. Dyer. 1985 Towards a Self-Extending Phrasal Lexicon. *Proceedings 23rd Annual Meeting of the Association for Computational Linguistics*. Chicago, IL, July (b)

Zernik, U.; M. G. Dyer. 1986 Disambiguation and Acquisition through the Phrasal Lexicon. *Proceedings 11th International Conference on Computational Linguistics*. Bonn, Germany (a)

Zernik, U.; M. G. Dyer. 1986 Language Acquisition: Learning Phrases in Context. In T. Mitchell, J. Carbonell, R. Michalsky, *Machine Learning: A Guide to Current Research*. Boston, MA, Kluwer (b)

Zernik, U. 1987 How Do Machine-Learning Paradigms Fare in Language Acquisition? *Proceedings Fourth International Workshop on Machine Learning*. Irvine, CA, June (a)

Zernik, U. 1987 Acquiring Idioms from Examples in Context: Learning by Explanation. *Proceedings 13th Annual Meeting of the Berkeley Linguistic Society*. Berkeley, California, February (b)

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

Zernik, U. 1987 "Learning Idioms with and without Explanation" *10th International Joint Conference on Artificial Intelligence*, Milan (d)

Zernik, U. 1987 "Language Acquisition: Learning a Hierarchy Of Phrases," *10th Insternational Joint Conf. On Artificial Intelligence*, Milan (e)