An On-Line Model of Human Sentence Interpretation*

Daniel Jurafsky
571 Evans Hall
Computer Science Division
University of California at Berkeley
Berkeley, California 94720
jurafsky@magnolia.Berkeley.EDU

Abstract

This paper presents a model of the human sentence interpretation process, concentrating on modeling psycholinguistic data through the use of rich semantic and grammatical knowledge and expectations. The interpreter is an on-line model, in that it reads words left-to-right, maintaining a partial interpretation of the sentence at all times. It is strongly interactionist in using both bottom-up evidence and top-down suggestions to access a set of constructions to be used in building candidate interpretations. It uses a coherence-based selection mechanism to choose among these candidate interpretations, and allows temporary limited parallelism to handle local ambiguities. The interpreter is a unified one, with respect to both representation and process. A single kind of knowledge structure, the *grammatical construction*, is used to represent lexical, syntactic and semantic knowledge, and a single processing module is used to access and integrate these structures.

Criteria for a Theory

Although the sentence interpretation process has received a great amount of attention in the cognitive science community, earlier models address very limited subparts of the problem of interpreting an utterance, and do not extend well to other significant parts of the task. As a guideline for designing and judging any model of sentence interpretation, this section presents three criteria of adequacy. The first criterion of **Functional Adequacy** constrains the nature of the interpretation.

Functional Adequacy: *An interpreter must produce a representation which is rich and complete enough to function as an interpretation of the sentence in a larger model of language understanding.*

The functional adequacy criterion is the most important one for an interpreter which is intended to model human processing. It is the necessity of meeting this criterion which distinguishes an interpreter, which must meet semantic and functional constraints on its representation, from a parser, which need not.

The second criterion for an interpreter is that of **Representational Adequacy:**

Representational Adequacy: *An interpreter must include a declarative and linguistically motivated representation of linguistic knowledge.*

This criterion insures that the representational basis of the processing model meets independent linguistic criteria for linguistic knowledge, particularly the need to capture relevant linguistic generalizations and account for the creativity of the language faculty. This knowledge must include more than just phonological or syntactic information. An interpreter must bring to bear a large and rich collection of semantic, pragmatic, and world knowledge in order to be complete enough to meet the constraint of Functional Adequacy.

The final criterion concerns psychological validity:

Cognitive Adequacy: *An interpreter must meet standards of psycholinguistic and general cognitive validity.*

The principle of Cognitive Adequacy requires that the theory account in a principled manner for psycholinguistic results. Such results include the on-line nature of the language interpretation process, the nature and time course of lexical access, on gap filling, the use of thematic roles, on inference, anaphora, and work on grammatical interpretation phenomena such as attachment preferences and garden path effects.

As mentioned above, most models have focused on limited parts of sentence interpretation process, and do not meet all the adequacy criteria. Many processing models which emphasize Representational Adequacy, particularly those associated with linguistic theories, such as Ford *et al.* (1982) (LFG), Marcus (1980) (EST) or Pritchett (1988) (GB), include no semantic knowledge and do not meet the criterion of Functional Adequacy. Alternatively, some models such as Riesbeck & Schank (1978) have emphasized semantic knowledge but ignored syntactic knowledge. In general, much of the cognitive modeling community has emphasized either syntactic parsing or lexical access. Very few cognitive models of interpretation have been proposed, although these models (such as Hirst (1986) and Kurtzman (1985)) have proved extremely important.

Architectural Principles

The first architectural principle, the **On-Line Principle**, follows directly from the criterion of cognitive adequacy:

On-Line Principle: *Maintain a continually-updated partial interpretation of the sentence at all times in the processing.*

There is a great amount of psycholinguistic evidence for the on-line nature of natural language processing, including Swinney (1979), Tanenhaus *et al.* (1979), Tyler

---

*Thanks to Peter Norvig, Nigel Ward, and Robert Wilensky. Sponsored by the Defense Advanced Research Projects Agency (DoD), monitored by Space and Naval Warfare Systems Command under N00039-88-C-0292, and by the Office of Naval Research under contract N00014-89-J-3205.

& Marslen-Wilson (1982), Marslen-Wilson *et al.* (1988) and many others cited below.

The on-line principle has two implications for the interpreter. The first is that it must produce an interpretation incrementally, that is in a strictly left-to-right manner while the sentence is being processed. This rules out the traditional depth-first or backtracking control structure for parsers. Thus, for example, depth-first ATN's do not conform to the principle, even those like RUS (Bobrow & Webber 1980) which eliminate some backtracking through semantic constraints.

The second implication of the on-line principle is that the interpreter cannot maintain all possible interpretations of a sentence during the processing. It is required, fairly frequently, to choose a single interpretation with which to continue processing. Evidence from lexical access, (Swinney (1979) and other cited above), inference generation (Swinney & Osterhout 1990) gap filling (Kurtzman (1989) and Carlson & Tanenhaus (1987)) show that selection among candidate interpretations is done on-line. This rules out the use of parallel parsers which maintain every possible syntactic or semantic structure in parallel, such as the active chart parser of Winograd (1983), the breadth-first ATN parser (Woods 1970), or the table-driven parser of Tomita (1987).

Unfortunately it is not possible to follow the on-line principle by simply choosing an interpretation immediately whenever an ambiguity arises. This is due to the fundamental conflict in human language understanding between the need to produce an interpretation as soon as possible, and the need to produce the correct interpretation. Because evidence for the correct interpretation may be delayed, any on-line interpreter must choose a method for integrating this late evidence.

Our model uses limited local parallelism to represent these local ambiguities while waiting for further evidence. At any point, multiple possible candidate interpretations are entertained, but only for a short time, and the interpreter is forced to choose among them quickly. The use of temporary local parallelism in interpretation can be considered an extension of results which show temporary local parallelism in lexical processing (Swinney (1979), Tanenhaus *et al.* (1979), and Tyler & Marslen-Wilson (1982)). In this sense our model resembles Gibson (1990a) and (1990b) and Kurtzman (1985).

Alternative approaches for modeling local ambiguity fall into two classes. The first is the "wait-and-see" approach proposed by Marcus (1980) (used also by the shift-reduce parser of Shieber 1983). In this approach, the model waits to build structure until it can be certain it is building the correct interpretation, although the delay is strictly limited. The second class of models use global heuristics (such as Minimal Attachment) to resolve local ambiguity immediately. Example of these include Kimball (1973), Frazier & Fodor (1978), Wanner (1980), and Pritchett (1988).

The second principle, the **Expectation Principle**, calls for a knowledge-based approach to sentence processing.

Expectation Principle: *Make use of syntactic, semantic, and higher-level expectations to help integrate new information into the interpretation.*

The use of expectations to guide the integration of linguistic knowledge in processing is well-grounded in artificial intelligence. Language interpretation should make use of frames (Hirst 1986), thematic roles (Carlson & Tanenhaus 1987 and Stowe 1988), and other high-level semantic information (Riesbeck & Schank 1978) to build interpretations.

The final principle, the **Uniformity Principle**, makes more specific claims about the algorithm used to produce the interpretation.

Uniformity Principle: *A single processing module accounts for access, integration, and selection of structures at all levels of sentence processing.*

The uniformity principle proposes a single, integrated mechanism to replace the traditional informationally encapsulated lexical analyzer, syntactic tree-builder, morphological analyzer, and interpretation mechanisms. This has the advantages of parsimony and elegance.

The functions of access, integration, and selection apply uniformly across the lexical, syntactic, and semantic domains. For example, the access function accounts for the access of lexical items as well as syntactic rules (as we will see below, this is because both are represented as "grammatical constructions"). The integration function builds structures by combining component structures at each level (in building words, syntactic phrases, or semantic interpretations). The selection function resolves both lexical and higher-level ambiguities.

Introducing the Algorithm

Any model must implement the basic functions of the interpretation process: *access*, *integration*, and *selection*. This section provides a sketch of the algorithm used to implement each function.

The *access* function suggests possible constructions to use in an interpretation, using a number of knowledge sources. These include bottom-up information from the input or from previously accessed constructions, and top-down information and semantic expectations. Constructions can also be annotated with access clues. The *integration* function combines information from various constructions to build complete interpretations. The function is currently implemented using an extension of unification. The *selection* function chooses the best interpretation from the disjunction of candidates being considered at any time. Selection takes place quite soon after a set of parallel candidates is proposed, using a simple coherence metric to choose among interpretations.

In the remainder of the paper, I will further describe these three theories and the representation of linguistic knowledge in the interpreter. A prototype interpreter has been implemented, using a small grammar of about 60 constructions and interpreting a set of questions from an earlier consultant project.

Representation

The interpreter's linguistic knowledge consists of a collection of structures which represent information from various domains of linguistic knowledge: phonological, morphological, syntactic, semantic, and pragmatic. These uniform collections of constraints are called *grammatical constructions*, following the construction grammar of Fillmore *et al.* (1988). A construction is an abstraction over some complex pairing of meaning and form. Because both lexical items and syntactic rules are constructions, there is no distinction between the lexicon and the syntactic rule base.

The representation language and the grammar have a number of distinguishing features, two of which will be discussed here. First is the ability to define constituents of constructions semantically as well as syntactically. Second is the definition of two kinds of constructions, *strong constructions* and *weak constructions*, characterized by their positions on a grammatical abstraction hierarchy.

Semantic constraints on a constituent are part of the definition of a construction. If an instance of a construction violates either syntactic or semantic constraints on its constituents it is unacceptable. Following Stucky (1987), I call such unacceptable constructions *uninterpretable* rather than *ungrammatical*, using the term *interpretability* for this extension of the idea of grammaticality. This distinguishes constructions from Montague Grammar rule-pairs, whose semantic rules play no role in grammaticality.

As an example of a construction which requires semantic constraints, consider the **How-Scale** construction first defined in Jurafsky (1990) which occurs in examples like the following:

(1a) **How wide** *is the ocean?*
How accurate *is her prophecy?*
(1b) **How much** *does she love him?*
How often *does it rain in San Francisco?*
How quickly *did she finish her work?*
(1c) **How many** *angels can dance?*

The **How-Scale** construction has two constituents. The first constituent is the lexical item "how". The second may be an adjective, such as "wide" or "accurate" in (1a), an adverb such as "quickly" or "often" in (1b), or even a quantifier like "many". Specifying this constituent syntactically requires a very unnatural disjunction of adverbs, quantifiers, and adjectives. Furthermore, such a disjunctive category is insufficient to capture the constraints on this constituent. For example, not every adverb or adjective may serve as the second constituent in the construction. Note the uninterpretability of (2abc), which have respectively an adverb, an adjective, and a quantifier as their second constituent.

*(2a) *How abroad ... ?*
*(2b) *How infinite ...?*
*(2c) *How three ...?*

The commonality among the grammatical uses of the construction can only be expressed semantically: the semantics of the second constituent must be *scalar*. A *scale* is a semantic primitive in the representational system, and is used to define traditional scalar notions like *size* or *amount* or *weight*. Note that in (1) above all the elements which are allowable as second constituents for the **How-Scale** construction have semantic components which are scales. Terms like "wide", "strong", and "accurate" meet the traditional linguistic tests for scalar elements (such as co-occurrence with scalar adverbs like "very", "somewhat", "rather", and "slightly"). The elements in the ungrammatical examples (2) do not have any sort of scalar semantics. The second constituent of the **How-Scale** construction may be an adjective, an adverb, or a quantifier so long as it has the proper semantics.

A theory which could not use semantic information to constrain a constituent would be unable to represent the **How-Scale** construction completely. Even a theory such as HPSG (Pollard & Sag 1987) which allows semantic constraints for lexical constructions could not represent it because HPSG does not allow semantic constraints on *syntactic* constructions. Figure 1 presents a sketch of the representation of the **How-Scale** construction. The semantic representation language used is a frame-like one, where the operator 'a' creates an instance of a concept.

```
(lexicalconstr How-Scale
    [(a Identify $t
        (Unknown $x)
        (Background $s)
      SuchThat
      (a Scale $s
         (Location $z $x)))]
   ->
    ["how"]
    [(a Scale $s
        (On $z))])

 "Given two constituents, one the word "how"
  and one a scale $s, construct a question
  about the location of object $z on scale $s"
```

Figure 1

A second feature of the grammar is the novel use of an abstraction hierarchy to structure the grammar. The grammar includes two kinds of constructions: *strong constructions*, which express productive grammatical knowledge, and *weak constructions*, which express abstractions over strong constructions. The use of these weak constructions allows us to capture the traditional notion of *semi-productive rules* or *subregularities*. Psycholinguistic arguments for the presence of these kinds of constructions are summarized in Cutler (1983). Weak constructions provide a structure for the linguistic knowledge base, rather than acting as regular, productive constructions. This use of the abstraction hierarchy is different than other uses of abstraction in linguistic knowledge (Bobrow & Webber (1980), Jacobs (1985), Flickenger *et al.* (1985), Pollard & Sag (1987), Jurafsky (1988)) in making this epistemological distinction between terminal (strong) and non-terminal (weak) nodes in the hierarchy.

The Access Theory

The history of models of access is an extremely rich one, but has tended to be somewhat balkanized. Psycholinguists have studied lexical access extensively, so much so that a rather broad consensus has arisen on at least the general nature of the lexical access process, while very little psycholinguistic work has been done on syntactic access. Some work on syntactic access has been done by computational linguists who have studied the computational properties of various algorithms for syntactic rule-access in parsing, but with no attempt to model human behavior.

By conflating the lexicon, the syntactic rule-base, and the semantic interpretation rules into a single linguistic knowledge base, and by using a uniform processing module, we are able to propose a single access algorithm which accounts for psycholinguistic data and meets computational criteria.

Our access algorithm generalizes the lexical access algorithm indicated by results such as Swinney (1979) and Tanenhaus *et al.* (1979) which show that when an ambiguous input is read, every sense of the ambiguous word is activated. All of these candidate senses of the word are maintained for a short period of time, after which various contextual factors can help select a single sense. Our access function similarly activates multiple candidate constructions, using a number of knowledge sources for clues. Two sources of access suggestions are the traditional bottom-up and top-down ones. Constructions are suggested if their first constituent matches the left-hand side of a recently applied rule, or if their left-hand side matches the current position of some previously suggested construction.

Following Wilensky & Arens (1980), access can also be delayed until more than one constituent of a construction has been seen. Thus constructions like "The Big Apple", which occur rarely but begin with common constituents like "the", are not accessed whenever "the" appears in the input. Instead such constructions are augmented with information specifying how much of the construction should be seen before it is accessed.

The richest source for access suggestions is semantic expectations. When a verb has a particular semantic subcategorization for one of its arguments, the access mechanism suggests a construction which builds that semantics. For example when parsing the verb "know", which semantically subcategorizes for a proposition, the mechanism suggests the Proposition construction, whose syntax builds a finite clause with a "that" complementizer. This is consistent with the experimental results of Shapiro *et al.* (1987) (also suggested by Kurtzman (1989)) that a verb's semantic arguments rather than syntactic ones are used immediately in processing. In this sense, our model is a *strongly interactionist* one, in allowing top-down and contextual information to directly affect the access of constructions. Although many researchers have not found evidence of these contextual effects, Tabossi (1988) has found such evidence by using particularly strong contexts. In general, construction access in our approach resembles the broader problem of conceptual access in memory (see also Riesbeck (1986).

The access theories of most earlier analyzers are quite straightforward. For example in a traditional bottom-up parser, the access algorithm recursively selects any rule for which a handle is found in the input. Access with top-down parsers works the same way, although indexing a rule by its left-hand side instead of by the first constituent. Some parsers (Tomita 1987) allow the use of both top-down and bottom-up access methods. In all these cases, access is very simple, since the number of syntactic categories is quite small. In the system described here, a construction's constituents may include any set of semantic relations rather than being restricted to a small, finite set of syntactic symbols. Thus these simple access methods used for parsers are insufficient. Most semantic analyzers have also simplified the access problem. For example in ELI (Riesbeck & Schank 1978) and the Word Expert Parser (Small & Rieger 1982) all constructions are lexical, and thus there are no higher-level constructions to consider accessing.

The Integration Theory

Partial interpretations are built by combining the information from constructions with an operation called *integration*. In its simplest form, integration is an extension of the unification operation to a richer semantic domain. Integration also differs from unification in being asymmetric, because it gives privilege to constructions over proposed constituents. Any proposed constituent must meet the constraints established by a construction, but not vice versa. Integration also allows modifying the structure of the elements being integrated, such as by binding the value of one element to some open variable in another. Where unification merely binds two elements together, integration allows one to fill some semantic gap inside the other.

The remainder of this section will work through the representation and integration of the two constituents of the **Wh-Non-Subject-Question** construction (Jurafsky (1990)). This construction accounts for sentences which begin with certain wh-clauses, where these clauses do not function as the subject of the sentence. Examples include:

(3a) **How** *can I create disk space?*

(3b) **What** *did she write?*

(3c) **Which book** *did he buy?*

The construction has two constituents. The first, indicated in bold type in the examples above, is a wh-element. The second is an instance of the **Subject-Second-Clause** construction, which consists of an auxiliary, a subject, and a verb phrase. The representation for the construction appears in Figure 2 below:

```
(constr Wh-Non-Subject-Question
  [(a Question $q
      (Queried $var)
      (Background (Integrate $/pre $/v)))]
  ->
  [(a Identify $t
      (Unknown $var)
      (Background $pre))]
  [(a Subject-Second-Clause $v)]  )

"Given two constituents, the first a questioned
element $t and the second a subject-second-clause
$v, build a question about $t which includes as
background knowledge the information in $v."
```

Figure 2

Note in Figure 2 that the background knowledge for the question is formed by integrating the variables $pre and $v. These contain the information from the two constituents. Note also that each of these variables is preceded by a slash. A slash on a variable means that the variable may be the matrix structure in which to search for a gap. Thus to integrate an argument into a verb, the variable representing the verb is slashed, while the argument is not. In Figure 2, both variables are slashed, indicating that the semantic gap could be in the structures bound to either of these variables. The gap could be inside the **Subject-Second-Clause** or inside the **Identify** structure.

For example, in the sentence *"What did she write?"* the gap is located in the second constituent, the **Subject-Second-Clause**, because the verb "write" has an unfilled semantic slot for the object written. The integration algorithm will bind the semantics of "what" to the unfilled "written-object" slot of the verb "write". Note that gap-filling is a semantic, not a syntactic process.

For the sentence *"How can I create disk space?"*, the gap is in the first constituent, the word "how". The semantics of this "how" construction in Figure 3 below are concerned with specifying the means or plan by which some goal is accomplished. The gap in this construction is the goal $g.

```
(a Identify $t
     (Unknown $p)
     (Background $x)
 Such-That
 (a Means-For $x
    (Means $p)
    (Goal $g)))

"The means $p for achieving
some goal $g is unknown."
```

Figure 3

The second constituent, the **Subject-Second-Clause** "can I create disk space", produces the semantics shown in Figure 4.

```
(a Ability-State $x
   (Actor (a Speech-Speaker))
   (Action
       (a Creation-Action
           (Actor (a Speech-Speaker))
           (Theme (a Disk-Freespace))))

"the ability of the speaker in the
discourse to create disk free space"
```

Figure 4

In order to build the correct interpretation of the sentence, the integration algorithm realizes that the goal $g in Figure 3 is a semantic gap which can be filled by the Ability-State $x in Figure 4, and it binds the Ability-State to the variable $g. The final result of this integration looks like Figure 5.

```
(a Question $q
  (Queried $p)
  (Background
    (a Means-For
      (Means $p)
      (Goal
        (a Ability-State  $x
           (-Actor (a Speech-Speaker))
           (-Action
                (a Creation-Action
                     (Actor (a Speech-Speaker))
                     (Theme (a Disk-Freespace)]
"A question about the means for achieving the
goal of being able to create some disk space."
```

Figure 5

This example of integration highlights a number of linguistic features of this model. First, the gap in the sentence *"How can I create disk space"* is in the word "how" rather than in the **Subject-Second-Clause**. Other linguistic analyses require wh-phrases to fill a syntactic gap in the matrix clause, which requires them to include traces or empty categories corresponding to each possible syntactic modifier position in the **Subject-Second-Clause**. By placing the gap inside the semantics of "how", we eliminate these numerous empty categories. Because the gap is semantic rather than syntactic, there is no need for the grammar to contain constructions with syntactic gaps (such as the slash-constructions of GPSG). Using semantic gaps also allows long-distance dependencies (WH-movement, Right-Node Raising, Topicalization, etc) to be treated with the same mechanism that is used to link verbs and other predicates with their arguments.

Using semantic gaps to act as expectations for binding WH- constructions is also consistent with a number of experimental results from psycholinguistics, including Kurtzman (1989), and Carlson & Tanenhaus (1987). Although these results were originally interpreted as support for the on-line location of gaps, they are consistent with a semantic integration of the wh-element into the verb. Our model can be distinguished from these by considering syntactic gaps which are located before the verb: we predict that subject gaps, for example, should not cause processing difficulty because the interpreter integrates the wh-element directly into the following verb without proposing a gap. This is exactly the result found by Stowe (1986).

The integration operation would need more inferential power to make the metaphoric inferences of Martin (1990), or the abductive inferences of Charniak & Goldman (1988) or Hobbs *et al.* (1988). Making these inferences in the integration algorithm allows them to be made on-line, unlike these earlier inferencing mechanisms.

The Selection Theory

The selection theory is based on assigning each candidate interpretation a confidence measure based on a simple coherence metric with semantic and constructional expectations (Ng & Mooney 1990). As soon as the measure of one of the candidates exceeds a threshold that interpretation is selected. If no candidate is ahead after a certain time threshold, the top-ranked interpretation is chosen, forcing the parallelism to be limited. Exactly when this selection takes place depends on the nature of the candidate interpretations. The current model uses a confidence threshold which is similar to the memory-capacity model of Gibson (1990b) and (1990a). Because an interpretation may be selected before all possible evidence has come in, the analyzer may choose an incorrect interpretation, discarding the correct one, and producing the well-known *garden path* phenomenon.

The selection model is still incomplete. A more complete metric would need to include syntactic distance, as well as take into account the relative frequencies of constructions.

Conclusions

I have proposed an on-line, incremental, strongly-interactionist model of human sentence interpretation, with a unified linguistic knowledge base and a single algorithm to use this knowledge to build interpretations. The model focuses on the use of semantic knowledge in each part of the interpreter: the use of semantic information to improve construction *access*, to do more intelligent *integration* of constructions, and to *select* constructions based on their coherence with expectations. Similarly, the representation language focuses on the use of semantic constraints on constituents and a semantic account of long-distance dependencies.

References

Bobrow, Robert J. & Bonnie Webber. 1980. Knowledge representation for syntactic/semantic processing. In *Proceedings of the First National Conference on Artificial Intelligence*, pp. 316–323. Morgan Kaufmann.

Carlson, Greg N. & Michael K. Tanenhaus. 1987. Thematic roles and language comprehension. In Wendy Wilkins, editor, *Thematic relations*. San Diego: Academic Press.

Charniak, E. & R. Goldman. 1988. A logic for semantic interpretation. In *Proceedings of the 26th ACL*.

Cutler, Anne. 1983. Lexical complexity and sentence processing. In G. B. Flores d'Arcais & R. J. Jarvella, editors, *The Process of language*, pp. 43–79. New York: Wiley.

Fillmore, Charles J., Paul Kay, & M. C. O'Connor. 1988. Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, 64(3):501–538.

Flickenger, D., C. Pollard, & T. Wasow. 1985. Structure-sharing in lexical representation. In *Proceedings of the 23rd ACL*, pp. 262–267.

Ford, M., J. W. Bresnan, & R. M. Kaplan. 1982. A competence-based theory of syntactic closure. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*. Cambridge: MIT Press.

Frazier, L. & J. D. Fodor. 1978. The sausage machine: A new two-stage parsing model. *Cognition*, 6:291–295.

Gibson, Edward. 1990a. Memory capacity and sentence processing. In *Proceedings of the 28th ACL*.

Gibson, Edward. 1990b. Recency preference and garden-path effects. In *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, pp. 372–379.

Hirst, Graeme. 1986. *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge: Cambridge Univ Press.

Hobbs, Jerry R., Mark Stickel, Paul Martin, & Douglas Edwards. 1988. Interpretation as abduction. In *Proceedings of the 26th ACL*, pp. 95–103, Buffalo, NY.

Jacobs, Paul. 1985. A knowledge-based approach to language generation. Technical Report 86/254, University of California at Berkeley Computer Science Division, Berkeley, CA.

Jurafsky, Daniel. 1988. Issues in relating syntax and semantics. In *Proceedings of 12th Coling*, pp. 278–284, Budapest.

Jurafsky, Daniel. 1990. Representing and integrating linguistic knowledge. In *Proceedings of 13th Coling*, Helsinki.

Kimball, J. 1973. Seven principles of surface structure parsing in natural language. *Cognition*, 2(1):15–47.

Kurtzman, Howard S., 1985. *Studies in Syntactic Ambiguity Resolution*. PhD thesis, MIT, Cambridge, MA.

Kurtzman, Howard S., 1989. Locating wh- traces. Draft, to appear.

Marcus, Mitchell P. 1980. *A Theory of Syntactic Recognition for Natural Language*. Cambridge: MIT Press.

Marslen-Wilson, W. D., Colin M. Brown, & L. K. Tyler. 1988. Lexical representations in spoken language comprehension. *Language and Cognitive Processes*, 3(1):1–16.

Martin, James H. 1990. *A Computational Model of Metaphor Interpretation*: Academic Press.

Ng, Hwee Tou & Raymond J. Mooney. 1990. On the role of coherence in abductive explanation. In *Proceedings of the Eighth AAAI*, pp. 337–342. Morgan Kaufmann.

Pollard, Carl & Ivan A. Sag. 1987. *Information-Based Syntax and Semantics: Volume 1: Fundamentals*. Chicago: University of Chicago Press.

Pritchett, Bradley. 1988. Garden path phenomena and the grammatical basis of language processing. *Language*, 64(3):539–576.

Riesbeck, Christopher K. 1986. From conceptual analyzer to direct memory access parsing: An overview. In *Advances in Cognitive Science 1*, pp. 236–258. Chichester: Ellis Horwood.

Riesbeck, Christopher K. & Roger C. Schank. 1978. Comprehension by computer: Expectation-based analysis of sentences in context. In W. J. M. Levelt & G. B. Flores d'Arcais, editors, *Studies in the perception of language*, pp. 247–293. London: Wiley.

Shapiro, L. P., E. Zurif, & J. Grimshaw. 1987. Sentence processing and the mental representation of verbs. *Cognition*, 27:219–246.

Shieber, Stuart M. 1983. Sentence disambiguation by a shift-reduce parsing technique. *ACL Proceedings, 21st Annual Meeting*, pp. 113–118.

Small, S. L. & C. Rieger. 1982. Parsing and comprehending with word experts. In Wendy G. Lehnert & Martin H. Ringle, editors, *Strategies for Natural Language Processing*, pp. 89–147. New Jersey: Lawrence Erlbaum.

Stowe, L. A. 1986. Evidence for on-line gap location. *Language and Cognitive Processes*, 1(3):227–245.

Stowe, L. A. 1988. Thematic structures and sentence comprehension. In G. N. Carlson & M. K. Tanenhaus, editors, *Linguistic structure in language processing*. Dordrecht, Holland: Kluwer Academic.

Stucky, Susan U. 1987. Configurational variation in english. In Geoffrey J. Huck & Almerindo E. Ojeda, editors, *Syntax and Semantics 20: Discontinuous Constituency*, pp. 377–404: Academic Press.

Swinney, D. A. 1979. Lexical access during sentence comprehension: (re)consideration of context effects. *Journal of Verbal Learning and Verbal Behavior*, 18:645–659.

Swinney, David A. & Lee Osterhout. 1990. Inference generation during auditory language comprehension. In Arthur C. Graesser & Gordon H. Bower, editors, *Inferences and Text Comprehension*: Academic Press.

Tabossi, P. 1988. Sentential context and lexical access. In G. W. Cottrell & M. K. Tanenhaus S. L. Small, editor, *Lexical ambiguity resolution*. San Mateo: Morgan Kaufmann.

Tanenhaus, Michael K., James M. Leiman, & Mark S. Seidenberg. 1979. Evidence for multiple stages in the processing of ambiguous words in syntactic contexts. *Journal of Verbal Learning and Verbal Behavior*, 18:427–440.

Tomita, Masaru. 1987. An efficient augmented context-free parsing algorithm. *Computational Linguistics*, 13(1–2):31–46.

Tyler, L. K. & W. D. Marslen-Wilson. 1982. Speech comprehension processes. In E. C. T. Walker & M. Garrett J. Mehler, editor, *Perspectives on mental representation*. Hillsdale, NJ: Erlbaum.

Wanner, Eric. 1980. The ATN and the sausage machine: which one is baloney. *Cognition*, 8:209–226.

Wilensky, Robert & Yigal Arens. 1980. Phran: A knowledge-based natural language understander. In *Proceedings of the 18th ACL*, pp. 194–204, Philadelphia.

Winograd, Terry. 1983. *Language as a Cognitive Process: Volume 1 Syntax*: Addison-Wesley.

Woods, William A. 1970. Transition network grammars for natural language analysis. *Communications of the ACM*, 13(10):591–606.