Cognitive Science 29 (2005) 375–419
Copyright © 2005 Cognitive Science Society, Inc. All rights reserved.

# An Activation-Based Model of Sentence Processing as Skilled Memory Retrieval

Richard L. Lewis[a], Shravan Vasishth[b]

[a]*Departments of Psychology and Linguistics, University of Michigan*
[b]*Institute for Linguistics, University of Potsdam*

Received 19 July 2004; received in revised form 2 February 2005; accepted 19 February 2005

**Abstract**

We present a detailed process theory of the moment-by-moment working-memory retrievals and associated control structure that subserve sentence comprehension. The theory is derived from the application of independently motivated principles of memory and cognitive skill to the specialized task of sentence parsing. The resulting theory construes sentence processing as a series of skilled associative memory retrievals modulated by similarity-based interference and fluctuating activation. The cognitive principles are formalized in computational form in the Adaptive Control of Thought–Rational (ACT–R) architecture, and our process model is realized in ACT–R. We present the results of 6 sets of simulations: 5 simulation sets provide quantitative accounts of the effects of length and structural interference on both unambiguous and garden-path structures. A final simulation set provides a graded taxonomy of double center embeddings ranging from relatively easy to extremely difficult. The explanation of center-embedding difficulty is a novel one that derives from the model's complete reliance on discriminating retrieval cues in the absence of an explicit representation of serial order information. All fits were obtained with only 1 free scaling parameter fixed across the simulations; all other parameters were ACT–R defaults. The modeling results support the hypothesis that fluctuating activation and similarity-based interference are the key factors shaping working memory in sentence processing. We contrast the theory and empirical predictions with several related accounts of sentence-processing complexity.

*Keywords:* Sentence processing; Working memory; ACT-R; Cognitive modeling; Interference; Decay; Activation; Parsing; Syntax; Cognitive architectures

## 1. Introduction

In this article we present a detailed process theory of the moment-by-moment working-memory retrievals that subserve sentence comprehension. The theory is based on general, independently motivated principles of memory and cognitive skill and provides precise quanti-

Requests for reprints should be sent to Richard L. Lewis, Department of Psychology, University of Michigan, 525 East University, Ann Arbor, MI 48109–1109. E-mail: rickl@umich.edu

tative accounts of reading-time data. Our vehicle for bringing sentence processing into contact with cognitive theory is Adaptive Character of Thought–Rational (ACT–R; Anderson, this issue; Anderson & Lebiere, 1998). The sentence-processing theory is derived from the application of cognitive principles, as embodied in ACT–R, to the specialized task of sentence parsing. The resulting theory construes sentence processing as a series of skilled associative memory retrievals, modulated by similarity-based interference and the fluctuation of memory trace activation. It combines insights from cognitive architectures, memory theory, and psycholinguistic theory.

By focusing on working-memory retrieval, we are departing from a long tradition in psycholinguistics and computational psycholinguistics that takes pervasive local ambiguity as both the central functional problem and the central theoretical problem in human sentence processing (e.g., Altmann & Steedman, 1988; Crocker & Brants, 2000; Ferreira & Clifton, 1986; Frazier & Rayner, 1982; Jurafsky, 1996; MacDonald, Pearlmutter, & Seidenberg, 1994; Stevenson, 1994; Tabor, Juliano, & Tanenhaus, 1998; Tanenhaus, Spivey-Knowlton, Eberhard, & Sedivy, 1995; Trueswell, Tanenhaus, & Garnsey, 1994). This has been an extremely productive line of research that continues to provide insights into cross-linguistic processing regularities (Frazier, 1998), the nature and time course of information used to resolve ambiguities, and some major architectural issues such as serial versus parallel parsing (for reviews, see Clifton & Duffy, 2001; Mitchell, 1994).

However, an even older stream of research (Chomsky & Miller, 1963; Miller & Chomsky, 1963) focuses on another major functional requirement of sentence processing: the requirement to temporarily maintain partially interpreted linguistic material so that incoming material may be integrated with it. As McElree, Foraker, and Dyer (2003) pointed out, sentences with long-distance extractions provide a clear case of this requirement, as in (1), in which the noun *toy* is interpreted as the theme of *like*:

1. This weekend we bought a toy that Amparo hoped Melissa would like.

But long-distance linguistic dependencies show up routinely in other constructions as well, as illustrated by the following extended dependency between *the dog* and the verb *stopped*.

2. The dog running around the park without his collar yesterday finally stopped barking last night.

In fact, establishing any novel relation, whether long or short, requires some memory of the immediate past. This is the functional requirement for working memory. It is not unique to language, but is a necessary feature of any computational device that must process information over time (Elman, 1990) or compute sufficiently complex functions (Newell, 1990). This requirement gives rise to the following theoretical question:

*How are linguistic relations established in sentence processing—exactly what are the working-memory processes that bring prior linguistic material into contact with present material, and what are the constraints on those processes?*

Most sentence-processing theories make assertions *about* composed structures or attachments (e.g., explicitly stating why one is preferred over the other or why one is computationally more costly to maintain or build than another) without explaining the basic pro-

cesses and memory structures that give rise to them. (One outcome of this state of affairs is that almost no work has been done on the representation of serial order information in sentence processing, in contrast to speech production; Dell, Burger, & Svec, 1997.) This is even true for many explicit theories of working-memory resources in sentence processing, such as dependency locality theory (DLT; Gibson, 1998, 2000), which provides a characterization of the computational costs of building linguistic relations abstracted away from the processes that build them.

An important exception is the work of McElree et al. (2003) and Van Dyke and Lewis (2003), which explicitly attempts to pin down the retrieval processes in sentence comprehension. Using speed-accuracy trade-off paradigms that permit detailed time course analyses, McElree and colleagues built the case for an associative, parallel-retrieval process in parsing, and we draw extensively on this work in what follows. Van Dyke and Lewis (2003) used unambiguous and ambiguous structures to tease apart the effects of decay and interference and sketch a theory of retrieval in sentence processing that is consistent with the model presented here.

The remainder of this article is structured as follows. First, we briefly describe what it means, both theoretically and practically, to build a sentence-processing model within Adaptive Control of Thought–Rational (ACT–R) architecture. We then derive the sentence-processing theory from (a) ACT–R's architectural assumptions, (b) basic assumptions about the parsing algorithm from psycholinguistic work, and (c) representational assumptions from theoretical syntax. The next two major sections describe simulations that illustrate how the model accounts for a range of length, structural complexity, and garden-path reanalysis effects. We conclude with a summary of the theory and data coverage, provide an analysis of its relation to other accounts, and reflect on the role of ACT–R in the enterprise.

## 2. Toward computationally complete sentence-processing architectures

Our theoretical approach is composed of two related long-term goals. First, the sentence-processing theory should take the form of a functionally and computationally complete model of comprehension. We mean *complete* in two senses. The model should be *computationally* complete in that it specifies all the fixed mechanisms required to define any computational architecture: memories, primitive processes, and control structure (Lewis, 2000; Newell, 1973, 1990). The model should also be *functionally* complete for the particular task of real-time sentence comprehension, meaning it should specify mechanisms for lexical, syntactic, semantic, and referential processing, ambiguity resolution at all levels, and the reanalysis of ambiguous material that has been initially misinterpreted. These completeness criteria are only partially met in this model. In this article, we present a computationally complete architecture, but with an emphasis on specifying and testing the structure of working memory. Functionally, the emphasis of this model is on syntactic parsing.

The second long-term goal, alluded to previously, is to explain as much detailed psycholinguistic phenomena as possible with independent principles of cognitive processing. Cognitive architectures such as ACT–R provide the means to achieve both goals simulta-

Table 1
Summary of simulations and fits to reading-time data

| Parameter | Grodner and Gibson (2005) Experiment 1 | McElree et al. (2003) Experiment 2 | Grodner and Gibson (2005) Experiment 2 | Van Dyke and Lewis (2003) Experiment 4 | Grodner, Gibson, and Tunstall (2002) Experiment 1 | Double center embeddings |
|---|---|---|---|---|---|---|
| Decay rate | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 |
| Maximum associative strength | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| Activation noise | 0 | 0 | 0 | 0 | 0 | 0.3 |
| Latency factor | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 |
| Production firing time | 50 | 50 | 50 | 50 | 50 | 50 |
| $R^2$ | .76 | .91 | .86 | .77 | .77 | — |

neously: They are computationally complete, and they explicitly embody cognitive principles in a form that can be applied to a range of tasks.

Some researchers will reject outright our second long-term goal. Seeking to provide a general cognitive basis for linguistic processes might seem ill-advised if we take language to be a specialized faculty (Chomsky, 1980; Fodor, 1983). In fact, it must be doomed to fail for two reasons under Fodor's view: Not only is linguistic processing patently not a kind of general cognition, but because nothing secure is known (or can be known) about the nature of general cognition, it is surely not the place to look for detailed insights into other mental phenomena.

However, issues of modularity and specialization are largely orthogonal to the question of whether general cognitive principles may form the basis of an explanatory theory of sentence processing. The rejection of this independence implicitly assumes that the only way that general cognitive principles can explain both Task A and Task B is if Task A and Task B both execute on the *same* general purpose mechanisms. This need not be the case, and our guiding hypothesis is that language comprehension is a specialized process operating on specialized representations that is nevertheless subject to a range of general principles and constraints on cognitive processing (Bever, 1970; Lewis, 1996; Newell, 1990). The sharing of mechanisms and resources is a separate, though important, issue.

## 3. Overview of the essential elements of ACT–R theory

The theoretical basis of this model is the set of components that form the core of ACT–R's theory of cognition: the declarative memory system and procedural memory system. At this stage we are not taking full advantage of ACT–R's perceptual–motor system, although we consider it a feature of the architectural approach that we may naturally extend the theory to provide models of eye movements in eye-tracking paradigms, button presses in self-paced paradigms, and so on. Our present focus is on pinning down the cognitive structure of sentence processing. For more comprehensive and up-to-date overviews of ACT–R, see Anderson (2005, this issue) and Anderson et al. (2005).

Table 2 summarizes the principles in ACT–R that we are applying to the task of sentence processing. This is not intended to be a comprehensive overview (e.g., it omits the perceptual–motor systems), and it departs from standard descriptions in some ways. We believe though it is an accurate characterization and one that highlights the aspects important for our present purposes. In what follows we briefly explain each principle.

(A1) The declarative memory component in ACT–R serves functionally as both a long-term memory (encompassing semantic and episodic memory) and a short-term working memory, although there is not a structural distinction between the two. It is useful to think of each item or chunk in declarative memory as a feature bundle that can have relations to other chunks. Fig.1 illustrates this chunk structure showing the model's representation of syntactic structure (the linguistic assumptions are discussed in more detail later).

(A2) Rather than a fixed buffer that can hold seven (Miller, 1956) or four (Cowan, 2000) chunks, ACT–R has a small set of buffers each with a capacity of one. There are three important cognitive buffers (Anderson, 2005, this issue): a *control buffer,* a *problem state buffer,* and a *retrieval buffer.* The control (or goal) buffer serves to represent current control state informa-

Table 2
The subset of ACT–R's major cognitive processing principles relevant to this model

| | |
|---|---|
| A1 | *Declarative memory of chunks.* Declarative memory consists of items (*chunks*) identified by a single symbol. Each chunk is a set of feature–value pairs; the value of a feature may be a primitive symbol or the identifier of another chunk, in which case the feature–value pair represents a relation. |
| A2 | *Focused buffers holding single chunks.* There are an architecturally fixed set of *buffers,* each of which holds a single chunk in a distinguished state that makes it available for processing. Items outside of the buffers must be retrieved to be processed. |
| A3 | *Activation fluctuation as a function of usage and delay.* Chunks have numeric activation values that fluctuate over time; activation reflects usage history and time-based decay. The activation affects their probability and latency of retrieval. |
| A4 | *Associative retrieval subject to interference.* Chunks are retrieved by a content-addressed, associative retrieval process. Similarity-based retrieval interference arises as a function of retrieval cue overlap: The effectiveness of a cue is reduced as the number of items associated with the cue increases. |
| A5 | *Procedural memory of production rules with a least-commitment, run-time control structure.* All procedural knowledge is represented as production rules (Newell, 1973)—asymmetric associations specifying *conditions* and *actions*. Conditions are patterns to match against buffer contents, and actions are taken on buffer contents. All behavior arises from production rule firing; the order of behavior is not fixed in advance but emerges in response to the dynamically changing contents of the buffers. |

*Note.* ACT–R = Adaptive Control of Thought–Rational.

tion, and the problem state buffer represents this problem state. The retrieval buffer serves as the interface to declarative memory, holding the single chunk from the last retrieval. This structure has much in common with conceptions of working memory and short-term memory that posit an extremely limited focus of attention of one to three items, with retrieval processes required to bring items into focus for processing (McElree, 1993, 1998; McElree & Dosher, 1989; Wickelgren, Corbett, & Dosher, 1980). The state and retrieval buffers are the minimum functionally required to be able to establish novel relations between chunks. We further assume in the sentence-processing model the existence of a *lexical buffer,* which holds the results of lexical retrieval.

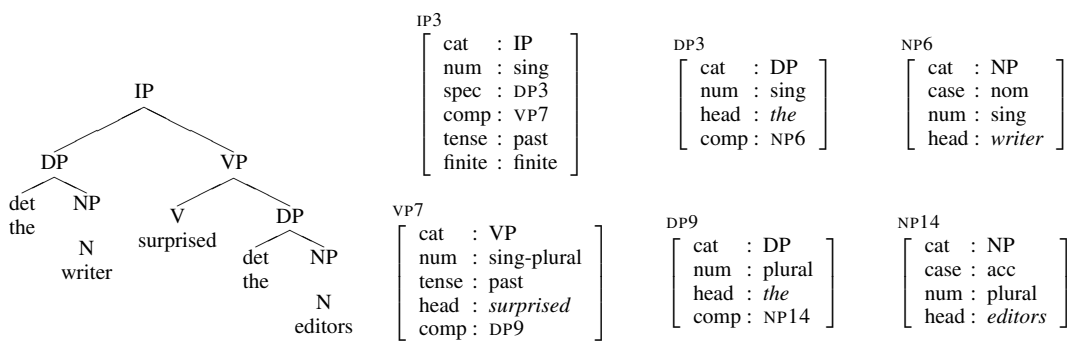


Fig. 1. An example of chunks in ACT–R's declarative memory, showing the chunk representation (right) of a syntactic structure (left).

How big is a chunk (Simon, 1974)? The claim that each buffer holds a single chunk has no empirical import if chunks can be arbitrarily defined to hold whatever information the modeler sees fit. The response to this concern is straightforward and is based on two observations. The first observation is that the answer is implicit in principle A1: A chunk is the representational element that enters into novel relations with other elements. The feature contents of two items and the novel relation between them cannot be represented in a single chunk. The second observation is that learning can of course change the representational vocabulary so that single symbols can come to denote more and more structure (Miller, 1956)—which is why we carefully restricted the first observation to *novel* relations. We take sentence comprehension to be principally a task of composing novel combinatorial representations, so the theoretical degrees of freedom in deciding what a single chunk contains are quite restricted.

(A3) All chunks have a fluctuating *activation* level, which is a function of usage history and decay. Equation 1 gives the equation for the base level activation of item $i$, $t_j$ is the time since the $j$th retrieval of the item, and the summation is over all $n$ retrievals.

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) \tag{1}$$

This equation is based on the rational analysis of Anderson and Schooler (1991) and is intended to track the log odds that an item will need to be retrieved, given its past usage history. The parameter $d$ is estimated to be 0.5 in nearly all ACT–R models (Anderson et al., 2005), and we adopt this value. A critical feature of this equation is that it does not yield a smoothly decaying activation from the initial encoding to this time; rather, the curve has a series of spikes corresponding to the retrieval events.

The total activation of a chunk is the sum of its base activation (given in Equation 1) and an associative activation boost received from retrieval cues in the goal buffer. The activation of chunk $i$ is defined as

$$A_i = B_i + \sum_j W_j S_{ji} \tag{2}$$

where $B_i$ is the base activation, $W_j$s are weights associated with elements of the goal chunk, and $S_{ji}$s are the strengths of association from elements $j$ to chunk $i$. The total activation of a chunk determines both retrieval latency and probability of retrieval. The weights $W_j$s are not generally free parameters in ACT–R models but are set to $G/j$, where $j$ is the number of goal features, and $G$ is the total amount of goal activation available, also set by default to 1.

Associative retrieval interference arises because the strength of association from a cue is reduced as a function of the number of items associated with the cue. This is captured by Equation 3, which reduces the maximum associative strength $S$ by the log of the "fan" of item $j$, that is, the number of items associated with $j$.

$$S_{ji} = S - \ln(fan_j) \tag{3}$$

The final equation we require maps activation level onto retrieval latency. The latency to retrieve chunk $i$ is given by

$$T_i = Fe^{-Ai} \tag{4}$$

$F$ is a scaling constant that varies across ACT–R models; in the sentence-processing model we fix $F$ to be 0.14 and use this for all the simulations.

Principles A1 to A4 in Table 2 and associated Equations 1 to 4 together form a simple theory of associative memory retrieval specified in enough detail to make quantitative predictions. What remains to have a computationally complete framework is an answer to the following question: How are the memory retrievals organized in the service of cognition?

Principle A5 gives ACT–R's answer. Cognition is controlled by production rules (Newell, 1973)—sets of condition–action pairs. In ACT–R, all conditions are constrained to match against the contents of buffers, and all actions are constrained to make changes to buffers. The form of a typical ACT–R production is given in (3).

(3) *IF* *control state is …*
*and chunk just retrieved has features …*
*and problem state has features …*
*THEN* *set new control state*
*and update problem state*
*and set retrieval cues*
*and request retrieval*

All cognitive behavior in ACT–R consists of the sequential selection and application of production rules. (See Anderson et al. (2005) for details of the ACT–R choice rule and utility learning.)

## 4. A theory of sentence processing based on ACT–R

Before describing and motivating the assumptions of the model in detail, it will be useful to have a high-level overview. Fig. 2 gives this overview, showing the critical buffer usage and production rule firings unfolding over time. The typical processing cycle is as follows; the numbers refer to the circled numbers in the figure.

a. A word is attended and a lexical entry is accessed from declarative memory (1) containing syntactic information, including argument structure. The lexical entry resides in the lexical buffer (2).
b. Based on this syntactic goal category (a kind of syntactic expectation) and the contents of the buffers, a production fires (3) that sets retrieval cues for a prior constituent to attach to.
c. The working-memory access takes some time (4), and eventually yields a single syntactic chunk that resides in the retrieval buffer (5).

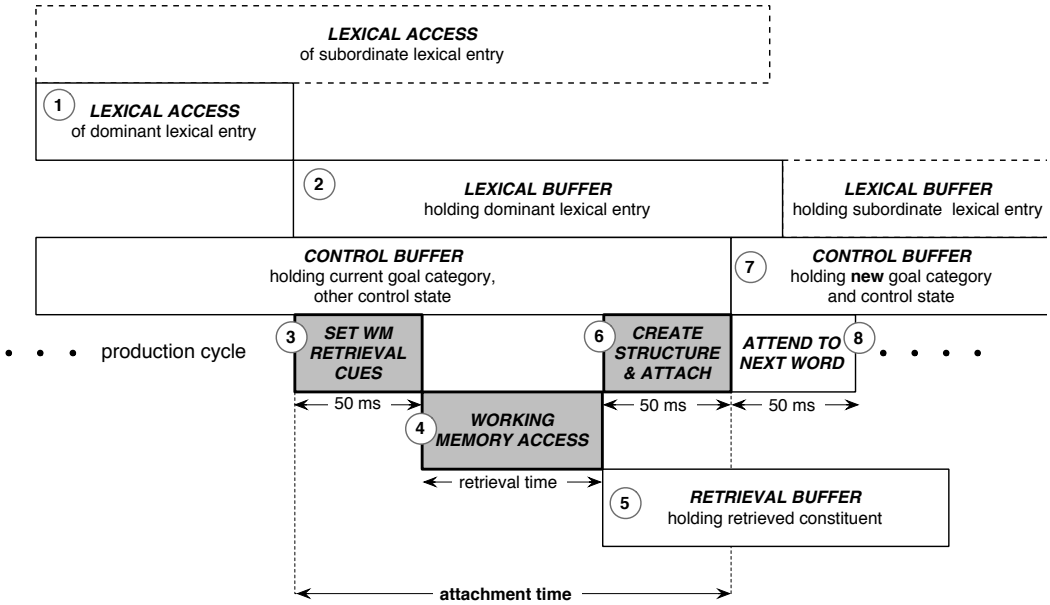


Fig. 2. Overview of the model, showing the critical focus buffers (control buffer, lexical buffer, and retrieval buffer) and processing dynamics (time flows left to right). The three key working-memory processes are shown in gray: (3) a production rule encoding grammatical knowledge sets cues for retrieval of a prior constituent; (4) a prior constituent is retrieved from working memory via parallel associative access; and (6) a second production rule creates the new structure and attaches it to the retrieved constituent.

d. Based on the retrieved constituent and lexical content, a production fires (6) that creates new syntactic structure and attaches it to the retrieved constituent. The control buffer is also updated with a new syntactic prediction (7).
e. Finally, other productions fire that guide attention to the next word.

The two production rules and retrieval processes in (3), (4), and (6) (in gray in the figure) are the critical processes of interest in this article; we refer to the time taken by all these processes jointly as the *attachment* time for a word. Apart from the new lexical buffer and parallel lexical access mechanisms, the structure of the architecture in Fig. 2 is standard ACT–R.

We now derive the details of the sentence-processing theory from a combination of ACT–R's assumptions and existing psycholinguistic evidence and theory. We first describe the major choice points in developing the model.

### *4.1. Major choice points in developing the sentence-processing model*

Practically speaking, building an ACT–R model means specifying the contents of procedural and declarative memory. For sentence processing, there are a few immediate major choices to be made:

- How should linguistic knowledge be distributed across the procedural memory and declarative memory?

- What kind of syntactic representation should be constructed, and how does this representation map onto ACT–R chunks?
- What kind of parsing algorithm should be used (head-driven, bottom-up, etc.)?
- How is local structural ambiguity handled? Are multiple structures generated and pursued in parallel? How are misanalyses recovered from?

We take up each of these questions in turn. We acknowledge that simply asking these questions presupposes much theoretically.

### *4.2. Distribution of linguistic knowledge across procedural and declarative memory*

We assume that the content of the lexicon resides in declarative memory. Although we are not concerned presently with modeling the details of lexical access, the immediate yield of this assumption is the prediction of context-modulated frequency effects (from Equation 1), and contextual priming effects (from Equation 3). The more contentious question concerns how grammatical knowledge is distributed across declarative and procedural memory.

Although it is impossible to construct a working model in ACT–R that consists *only* of declarative chunks, there is still a wide range of functionally viable distributions of grammatical knowledge across production rules and chunks. This range is reflected in the wide variety of parsers explored in computational linguistics and parsing theory. At one extreme are parsers with active procedures corresponding to grammar rules (e.g., traditional top-down recursive-descent parsers; Aho & Ullman, 1972). At the other extreme are parsers that encode all or most of the grammar in declarative form—the procedural component is simply an efficient interpreter of a database of declarative rules. Such parsers are widely favored in computational linguistics because they permit the easy inspection, modification, and extension of the grammar. Also closely related are lexicalist approaches to grammar and sentence processing, which seek to place as much grammatical structure as possible—perhaps all of it—into the lexicon, so that parsing is a matter of lexical retrieval and joining of retrieved structures (MacDonald et al., 1994; Schabes & Joshi, 1991; Srinivas & Joshi, 1999). Boland, Lewis, and Blodgett (2004) called this the "Full Lexical Representation Hypothesis." The potential explanatory gain, as MacDonald et al. and Jurafsky (1996) pointed out, is the unification of lexical and syntactic processing.

In contrast, the model we present here assumes that much grammatical knowledge is encoded procedurally in a large set of quite specific production rules that embody the skill of parsing. The model thus posits a structural distinction between the representation of lexical knowledge and the representation of abstract grammatical knowledge. Is such a move worth the potential loss of the unification of lexical and syntactic processing? We believe it is, for three reasons: two based on independent empirical evidence, and one based on ACT–R itself.

*First, empirical evidence suggests that not all syntactic knowledge is lexicalized.* Although early presentations of the lexicalist approach (e.g., MacDonald et al., 1994) emphasized the effects of traditional lexical features such as argument structure, it is now clear that a fully lexical parser must contain much more elaborated syntactic representations that go beyond argument structure. In particular, Frazier (1995, 1998) pointed out that a lexical parser must have the

ability to project beyond argument structure if it is to handle adjunct attachment and phenomena in head-final constructions (Bader & Lasser, 1994; Frazier, 1987b; Hirose & Inoue, 1997; Inoue & Fodor, 1995; Konieczny, Hemforth, Scheepers, & Strube, 1997). Some highly lexicalized grammar formalisms such as LTAG have the necessary properties (Schabes & Joshi, 1991; Srinivas & Joshi, 1999), such as adjunct positions encoded in lexicalized forms. However, recent empirical work consistently points to distinctions between argument and adjunct attachment that are unexpected under the lexicalized account (Boland & Blodgett, 2001; Boland et al., 2004).

In sum, these considerations weigh against loading all the syntactic information into the declarative lexicon. At first sight it may appear that this claim is inconsistent with well-known syntactic priming effects (Bock, 1986). However, syntactic priming can be explained within the ACT–R architecture in terms of the decay of production-relevant information (in addition to decay of declarative chunks); Lovett (1998) motivated this approach in the related and more general area of choice in human perceptual and response processes. In fact, recent work on syntactic priming is consistent with this approach: (Bock & Griffin in 2000) suggested that the persistence of priming over long periods favors a long-term adjustment within a sentence-production system rather than a transient memory account.

*Second, the goal of rapid processing mitigates against extra declarative retrievals in ACT–R*. Even if all syntactic knowledge is not lexicalized, it still might be possible to declaratively represent and access abstract structures, as in models based on *construction grammars* (e.g., Jurafsky, 1996; McKoon & Ratcliff, 2003). However, doing so would incur extra time cost in ACT–R. In general, ACT–R theory has always assumed that skill acquisition involves a shift from declarative to procedural processing. Because we are modeling a highly practiced behavior, when the choice arises it makes sense to assume information is proceduralized, rather than remaining in declarative form.

*Third, cognitive neuroscience evidence suggests that the lexicon and grammar do map onto distinct underlying declarative and procedural brain systems*. The natural ACT–R mapping of lexicon–grammar to declarative–procedural is consistent with a growing body of evidence from brain imaging and patient data (Ullman, 2004; Vannest et al., 2004;Vannest, Polk, & Lewis, in press). The mapping appears to show that combinatorial processing is realized by a frontal–basal–ganglia circuit, whereas noncombinatorial lexical processing is realized by a temporal circuit. The mapping of ACT–R's declarative–procedural system to these brain areas has already been proposed on independent grounds (Anderson et al., 2005). The existence of these separate brain systems along with this independent mapping considerably lessens the concern about loss of explanatory power in moving to a dual-memory-system approach.

### *4.3. Declarative representation of syntactic structure*

For the novel structures incrementally constructed during sentence processing, there is no choice in ACT–R for which memory system must be used. It must be the declarative system, under control of productions. Fig. 1 gives an example of the X-bar (Chomsky, 1986) structure that we assume, with a straightforward mapping onto chunks. Each chunk represents a maximal projection, with features corresponding to X-bar positions (specifier, comp, head)[1] and other syntactic features such as case and agreement.

It is important to understand that the entire syntactic tree built during the parse is not maintained in memory as a unified entity. Rather, syntactic nodes (e.g., Determiner Phrases) are maintained as chunks that are also values of features in a larger subtree. Given the architectural restrictions outlined previously, this means that accessing the information in subtrees generally requires working-memory retrievals.

### *4.4. Procedural representation of parsing skill*

In this section we describe a simple mapping of a well-known parsing algorithm onto production rules. The resulting production rules encode both the grammar and the procedural knowledge of how to apply it.

#### *4.4.1. Left-corner (LC) parsing*

There is much evidence that real-time sentence comprehension involves incremental structure building (see references cited previously): The human parsing mechanism strives to immediately predict syntactic structure as it successively encounters each word in a sentence. In syntactic terms, this means parsing is driven by a top-down (predictive) as well as bottom-up mechanism. Johnson-Laird (1983) was the first to note that this corresponds to the well-known LC parsing algorithm (Aho & Ullman, 1972).

LC parsing can be illustrated with a simplified example. Consider the rudimentary context-free grammar of English at the top of Fig. 3.[2] We use the LC rule, defined as follows: Given an input (terminal) and a goal category, if there exists a grammar rewrite rule of which the input is a left corner, replace the input with the left-hand side (LHS) of that rewrite rule; repeat this process with the LHS nonterminal symbol until no further replacements are possible. Given this grammar, and a sentence like *the dog ran,* the parse would proceed as in Fig. 3. When the parser makes the wrong commitment, it would need to restructure the predicted tree somehow. This can be accomplished by means such as backtracking over (ranked) alternative parse trees, or repair (Lewis, 1998a). As outlined below, the ACT–R model is a repair parser.

#### *4.4.2. The content of the production rules: How the model realizes LC parsing*

Note that the parsing algorithm in Fig. 3 implicitly assumes at least two memories: a control stack of predicted categories, and a memory of the partially built structures. In the ACT–R mapping it is critical that we do not implicitly or explicitly rely on such memories; everything must be realized within ACT–R's memory systems.

The realization of LC parsing in ACT–R is straightforward. Each word triggers the sequence of events described in the previous overview and illustrated in Fig. 2. There are thus two important kinds of production rules that embody parsing skill: productions that set working-memory retrieval cues, and productions that perform attachments to retrieved constituents. The form of the two types of rules is given in (4) (compare to the general ACT–R rule schema in [3]).

(4) a. *IF* *goal category is …*
*and lexical entry has features …*
*THEN* *set retrieval cues to …*
b. *IF* *lexical entry has features …*

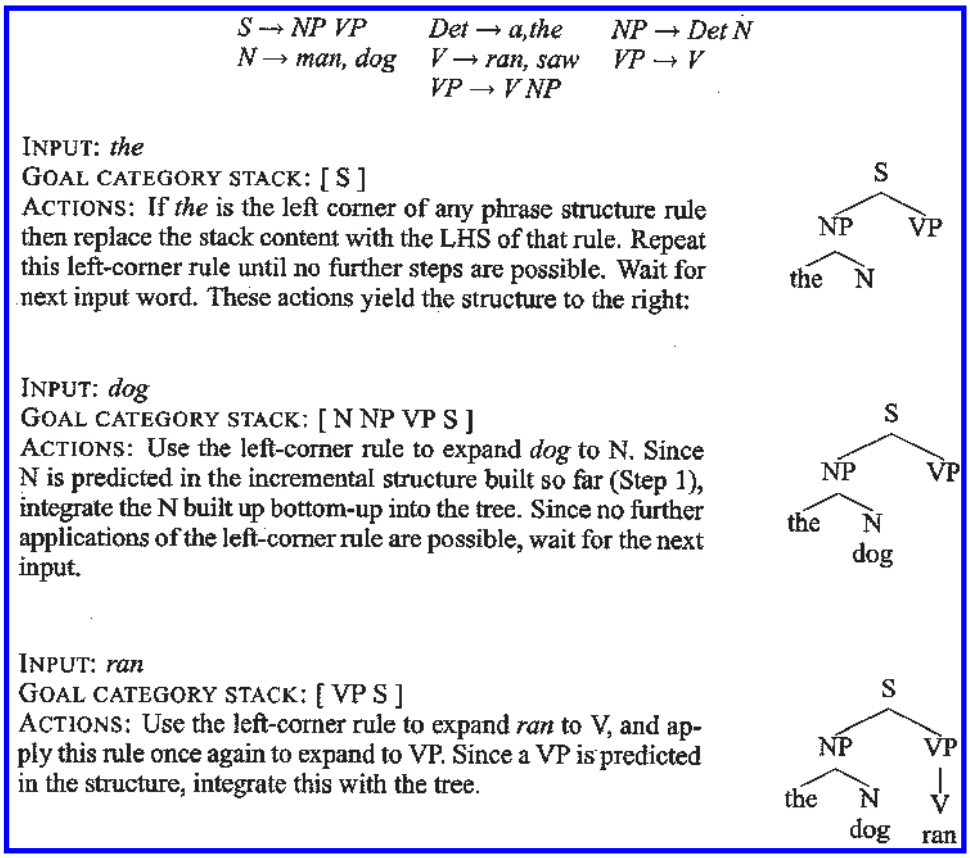


Fig. 3. A simple example of LC parsing, demonstrating the mix of bottom-up and top-down control with prediction. At top is the phrase structure grammar; the three panels and trees show the incremental parsing of *the dog ran*.

*and retrieved constituent has features …*
*THEN* *create new constituent*
*and attach it*

As a simple example, consider the case of projecting a determiner phrase from a determiner and attaching it to a predicted IP node (the first part of the LC example in Fig. 3). The two rules that handle this case are given as follows:

(5) a. Set-retrieval-cues-IP-goal-input-Det
*IF* goal category is IP
and lexical entry is a DET
*THEN* set retrieval cues to IP expectation

b. Attach-DP-as-subject-of-predicted-IP
*IF* lexical entry is category DET
and retrieved constituent is a
predicted category IP
*THEN* set goal category to be NP

and create new DP with det as head
and attach new DP as subject
of predicted IP

Note that the goal category "IP" is a primitive symbol; it is not an actual constituent. Access to the IP node in the parse requires a retrieval. A useful way to think of the syntactic goal category symbols is that they form a set of control states for the sentence processor.

These rules are a compiled form of the parsing and grammatical knowledge represented in the LC algorithm plus phrase structure rules such as those in Fig. 3. There are many other specific instances of these two rule classes, but for a given phrase structure rule set (or schema set), the set is finite. The direct mapping from specific context to specific actions that the rules embody is similar to the precompiled indexing schemes used in highly efficient LC parsers (van Noord, 1997).

#### *4.4.3. The elimination of the stack and serial order information*

In the ACT–R model, there is no separate stack or chart data structure. The memory consists exclusively of the chunks representing the syntactic structure built thus far, and the only access to that memory is via the retrieval mechanisms described previously. These chunks also double as a representation of the information in the control stack; a feature "next-goal" on each constituent chunk specifies the goal category that should be pursued once the constituent is complete.

Furthermore, there is no explicit serial order representation (Lewis, 2000; cf. McElree et al., 2003). This does not mean that word order plays no role—word-order constraints are deeply embedded in the structure of all the production rules, because they depend on a distinction between the word being processed now and what has come before (what must be retrieved). The critical problem is distinguishing the relative order of two items in the past. Instead of adopting a serial order mechanism, we are initially pursuing an extreme hypothesis about serial order representation in the human parser: There is none. Instead, the processor relies on the ability of retrieval cues to discriminate candidate attachment sites, and in cases where retrieval cues cannot discriminate, the processor relies (implicitly) on activation level. We explore the implications of this assumption later in the simulations of center embeddings.

#### *4.4.4. Retrieval cues for embedded structures and gapped structures*

One virtue of using a top-down goal category to guide behavior is that the processor can be sensitive to whether or not it is processing an embedded clausal structure or gapped structure (such as a relative clause). For gapped structures this is useful because it provides the triggering cue to attempt retrieval of the dislocated element. (We assume here a syntactic representation that uses empty categories, although the model could be reformulated as a direct association model.) Thus, in addition to goal symbols such as VP and IP there are also corresponding goal symbols VP-gapped and IP-gapped, and gap features on the constituent chunks themselves. Similarly, the model is sensitive to whether or not it is parsing an embedded clause by using goal symbols such as IP-embedded and VP-embedded and corresponding embedded features on the constituents themselves. This helps the processor functionally to discriminate between the predicted main and predicted embedded clauses during retrieval.

These assumptions have precedents in both the linguistic and psycholinguistic literature. The passing down of gap features (Gazdar, 1981; Gazdar, Klein, Pullum, & Sag, 1985) is similar to the use of the SLASH feature in head-driven phrase-structure grammar (Pollard & Sag, 1994), and the vocabulary of gapped categories is inspired by (combinatorial) categorial grammar (Bar-Hillel, 1953; Steedman, 1996). The distinction between embedded and main clauses is independently needed because these structures admit different syntactic phenomena (Green, 1976; Hooper & Thompson, 1973; Koster, 1978).

### *4.5. Responding to local ambiguity: Serial, probabilistic, repair parsing*

To fully specify a parser we must specify mechanisms for handling local lexical and structural ambiguity. In this model, these mechanisms follow from the ACT–R architecture and implicitly provide a theory of ambiguity resolution. Because our current focus is on working-memory retrieval and our initial set of results can be described independently of ambiguity resolution, the following brief summary will suffice:

- Lexical access proceeds via ordered access modulated by frequency and context, with competition effects. This is a direct result of Equations 1, 2, and 3.
- Structural ambiguity is resolved probabilistically via a combination of working-memory factors (such as recency) and ACT–R's rational production choice rule. This is a direct result of ACT–R's control structure and the declarative memory equations.
- A single structural interpretation is pursued, although multiple possibilities are locally generated in parallel. The single-path nature of the system is indirectly a consequence of associative retrieval interference, which mitigates against maintaining multiple similar structures.
- Limited recovery from misanalyses is initiated opportunistically by the reactivation of discarded structures, and completed by simple repair. This is indirectly a consequence of the architecture as well; it is the most natural outcome, given the serial nature of the parse. The discarded structures are simply chunks in memory like any other, but they have not participated in the parse and thus have not been retrieved since their initial creation.

### *4.6. An example parse*

We now briefly illustrate the behavior of the model on a simple example that brings all the mechanisms together. Consider the objective-relative (OR) clause structure in (6):

(6) The lawyer that the editor hired admired the writer.

Fig. 4 gives a partial trace of the model parsing this example, highlighting the processing of four words: the initial *the,* the complementizer *that,* and the embedded and main verbs. Each line prefixed by a timestamp indicates either the completion of a production rule firing or a memory retrieval. The two productions for *the* are the same ones previously given in (4).

The total syntactic attachment time indicated in the trace is the sum of the processing time after lexical access until the attachment is complete (see also Fig. 2). As described previously, each word is typically processed by (a) a lexical access (not shown in the trace), (b) a produc-

```
READING: "the" at  0.235

 0.286: Set-Retrieval-Cues-IP-Goal-Input-Det
 0.338: IP11 Retrieved
 0.388: Attach-DP-As-Subject-Of-Predicted-IP

 ==========================================
 Attachment time for THE:  0.152
 ==========================================

READING: "lawyer" at  0.623
READING: "that" at  1.016

 1.067: Set-Retrieval-Cues-Input-Comp
 1.103: DP12 Retrieved
 1.153: Attach-CP-As-Modifier-Of-Retrieved-DP

 ==========================================
 Attachment time for THAT:  0.136
 ==========================================

READING: "the" at  1.388
READING: "editor" at  1.947
READING: "hired" at  2.356

 2.408: Set-Retrieval-Cues-Goal-VP-Gapped-Input-V
 2.441: IP16 Retrieved
 2.491: Attach-VP-Transitive-DP-Object-Gap
 2.548: DP16 Retrieved
 2.598: Attach-Filler-In-Object-Gap

 ==========================================
 Attachment time for HIRED:  0.240
 ==========================================

READING: "admired" at  2.833

 2.885: Set-Retrieval-Cues-Goal-VP-Input-V-Finite
 2.963: IP11 Retrieved
 3.013: Attach-VP-Transitive-DP-No-Gap
 ==========================================
 Attachment time for ADMIRED:  0.178
 ==========================================
```

Fig. 4. A partial trace generated by the model in processing the sentence *The lawyer that the editor hired admired the writer.* Times are in seconds; see text for an explanation of "attachment time."

tion rule to set retrieval cues, (c) a working-memory retrieval, and (d) a production rule to perform the attachment.

When *the* is read at time 0.235, retrieval cues are set (via a production rule) for a predicted IP given the input determiner; the production rule execution takes 50 msec (the ACT–R default). The retrieval cues trigger a retrieval of the IP, which takes 53 msec. Finally, a second production rule attaches the determiner phrase as subject of the predicted IP, and this also takes another 50 msec. Adding 50, 52, and 50 gives the total attachment time for *the*: 152 msec.

The processing of *that* is similar. When it is read at time 1.067, retrieval cues are set for a DP, and the previously seen DP is retrieved and the CP attached to it as a modifier. The two productions take 50 msec each, and the retrieval 36 msec, resulting in a total attachment time of 136 msec for *that*.

For *admired* the attachment time 178 msec is the sum of 50 msec, 78 msec, and 50 msec (the two 50 msec for production rule execution times, and 78 msec for retrieving the IP node, which is expecting the VP). For *hired,* there is an additional production firing and retrieval associated with retrieving the relative pronoun and attaching a coindexed trace to fill the gap in the relative clause structure.

### *4.7. Summary of the major theoretical claims about working memory*

Before describing the quantitative simulations, let us take stock and summarize the major theoretical claims about working memory in sentence processing. Table 3 lists these claims. It is important to understand that these are simply instantiations of the basic ACT–R principles listed in Table 2, and formalized in part by Equations 1 to 3.

Table 3
Summary of the major theoretical claims about working memory in sentence processing

| | |
|---|---|
| SP1 | *Declarative memory for long-term lexical and novel linguistic structure.* Both the intermediate structures built during sentence processing and long-term lexical content are represented in declarative form by chunks, which are bundles of feature–value pairs. |
| SP2 | *Extremely limited working-memory focus.* Single-chunk buffers hold (a) the results of lexical access, (b) the constituent just retrieved from working memory, and (c) local control state, including the syntactic goal category. Active processing is restricted to the contents of these buffers. |
| SP3 | *Activation fluctuation as a function of usage and delay.* Chunks representing constituents during sentence processing and lexical entries in long-term memory have activation values that fluctuate over time; activation reflects usage history and time-based decay. The activation affects their probability and latency of retrieval. |
| SP4 | *Associative retrieval subject to interference.* All chunks, including chunks comprising working memory, are accessed by a content-addressed, associative retrieval process. Retrieval cues are a subset of the target chunk's features. Similarity-based retrieval interference arises as a function of cue overlap: The effectiveness of a cue is reduced as the number of items associated with the cue increases. |
| SP5 | *Efficient parsing skill in a procedural memory of production rules.* A large set of highly specific production rules constitute the skill of parsing and a compiled form of grammatical knowledge. The parsing algorithm is best described as a LC algorithm, with a mix of bottom-up and top-down control. Sentence processing consists of a series of memory retrievals guided by the production rules realizing this parsing algorithm. |

## 5. Quantitative effects of activation fluctuation and interference

In this section we describe a series of five sets of simulations that test the ability of the model to explain relatively complex patterns of reading times and provide quantitative accounts of those patterns. We are primarily interested here in the effects of decay and interference on syntactic processing, so we focus on experimental contrasts that factor out effects of lexical and other processing. These provide the most straightforward test of the model's working-memory assumptions.

### *5.1. How the fits were obtained*

In all the simulations reported in the following sections, the ACT–R model parses, word-by-word, examples of the sentences presented in the actual experiments. There are four important quantitative parameters that affect the predictions: the decay rate $b$ (Equation 1), the maximum associative strength $S$ (Equation 3), the latency factor $F$ (Equation 4), and the production execution latency. The decay rate $b$ has a standard ACT–R value of 0.5. The production execution latency has a standard ACT–R value of 50 msec, which is also consistent with the estimate in EPIC (Meyer & Kieras, 1997) and SOAR (Newell, 1990). The associative strength $S$ is estimated at 1.5 in several models of memory interference experiments (Anderson & Reder, 1999; Anderson et al., 2005), and we adopt that value here. We estimated the latency factor at 0.14; this just serves as a scaling factor. In sum, we are simply using ACT–R default parameter ranges and estimating a single parameter where the theory does not provide explicit guidance.

All of the simulations described here were fitted with the same parameter values; there is no parameter variation across simulations. Furthermore, the plotted "fits" are not standard linear regression fits with two parameters (a slope and intercept). Because ACT–R has its own scaling parameter, it is not appropriate to further scale the values with another regression parameter. Thus, we plot here absolute differences against the data, with a constant adjustment to factor out button-press times, and so forth.

Table 1 summarizes the simulation, the parameters, and the $R^2$ values. We report these $R^2$ values as a rough summary of how well the model is capturing the data patterns, with the caveat that correlations are based on a two-parameter linear model, and we are interested here in assessing the match of absolute time differences across the simulations. Rather than rely on a goodness of fit summarized by a single quantity, we prefer instead to guide the reader through analyses of the data that focus on individual contrasts of theoretical interest.

### *5.2. Simulation 1: Subject relatives (SRs) versus ORs*

There is considerable psycholinguistic evidence that English OR clauses are more difficult than SRs (e.g., Ford, 1983; Hakes, Evans, & Brannon, 1976; Holmes & O'Regan, 1981; Hudgins & Cullinan, 1978; King & Just, 1991; Larkin & Burns, 1977): They are processed more slowly and often result in poorer comprehension. Consider the pair in (7), from King and Just (1991), in which each relative clause modifies the subject of the main clause.

(7) a. [SR] The reporter who attacked the senator admitted the error.
    b. [OR] The reporter who the senator attacked admitted the error.

There have been many explanations proposed for this contrast. Most explanations can be classified into one of four types: (a) *distance* explanations (Gibson, 1998; Grodner & Gibson, in press; Just & Carpenter, 1992) attribute the contrast to the fact that in the OR structure there is a greater distance between the embedded verb *attacked* and the relative pronoun; (b) *double function* explanations (Bever, 1970; Sheldon, 1974) attribute the contrast to the fact that in the OR clause, one noun phrase (*reporter*) simultaneously plays two different underlying functions (object of the embedded clause and subject of the main clause); (c) *experience-based* explanations (MacDonald & Christiansen, 2002) attribute the contrast to the putative divergence of the OR structure from the canonical and much more frequent subject–verb–object English word order, or to the relatively rare occurrence of ORs (Korthals, 2001); and (d) *reanalysis* explanations attribute the contrast to the local ambiguity at *who* and the preference to initially analyze the structure as a SR (Frazier, 1987a, 1989).

There are empirical and conceptual difficulties with the latter three explanations; see Grodner and Gibson, in press, for a critique of the reanalysis and experience explanations.[3] One problem with the double-function account is that it does not hold up cross-linguistically. Studies in Japanese (Hakuta, 1981) and Hebrew (Schlesinger, 1975) have demonstrated that it is not double function, per se, but center embedding, or amount of interruption, that best accounts for difficulty.

We are interested in accounting for the pattern of reading times at embedded verbs and main verbs in both structures. King and Just (1991) collected word-by-word self-paced reading times for structures such as (7), but Grodner and Gibson (in press) pointed out that reading times for the main verb may be inflated in the OR case due to spillover from the embedded verb, so it is difficult to get a clean comparison of processing time on the main verb between the two structures. They corrected this (and other problems) by using the following structures:

(8) a. [SR] The reporter who sent the photographer to the editor hoped for a story.
    b. [OR] The reporter who the photographer sent to the editor hoped for a story.

Their observed reading times and the model generated times are shown in Fig. 5. The model captures the basic contrast between SRs and ORs; the highest reading times are on the embedded verb in the OR. There are two further interesting patterns present in the data. First, the effect of extraction type is greatest at the embedded verb. Table 4 shows this contrast. In fact, in the human data, there is no effect at all of extraction type on the main verb; there is a small effect present in the model. Second, the reading times on the main verb are shorter than the embedded verb in the OR condition, but longer in the SR condition.

This same interaction of extraction type and verb has been observed in two earlier reading-time studies. The pattern was observed in the King and Just (1991) experiment for all but the low working-memory-span subjects—despite the possibility of spillover inflating the reading times on the main verb as discussed previously. The pattern was also observed in a self-paced paradigm, using centrally presented words (Gordon, Hendrick, & Johnson, 2001). Neither the CC-READER model (Just & Carpenter, 1992) nor the locality-based integration metric (Grodner & Gibson, in press) captures this aspect of the data. The locality account predicts the highest reading times at the main verb in both constructions; the CC-READER model predicts higher times for the SR construction and equal times for the OR construction.

The contrasts between the SR embedded and main and OR main verbs in the ACT–R model are due only to differences in working-memory retrieval times, modulated by activation fluctu-

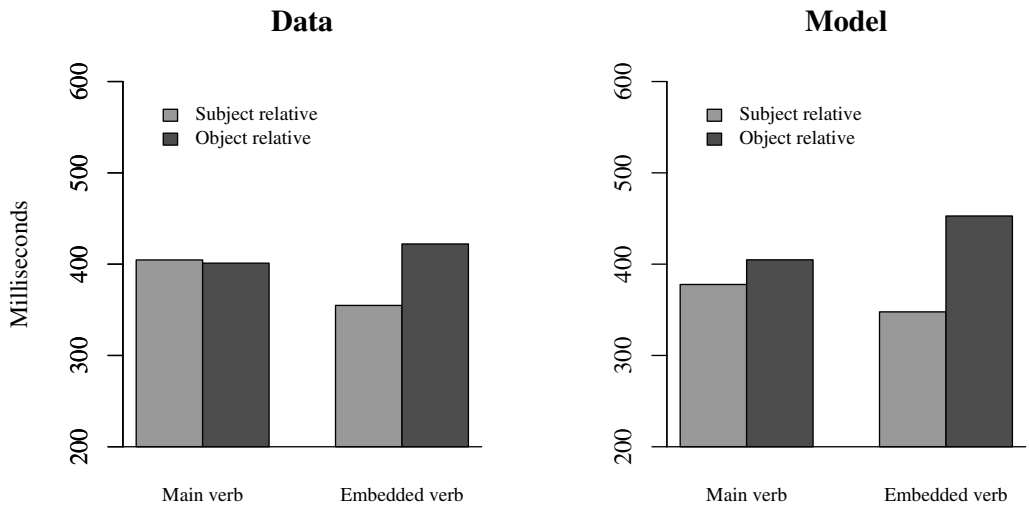


Fig. 5. Model-generated and observed self-paced reading times on embedded verbs and main verbs in SR and OR clauses. Data from Experiment 1, Grodner and Gibson (2005).

ation. The extra time at the object-gap filling is due in part to an extra retrieval and production cycle, as explained previously. The subject gap is posited and filled at the relative pronoun *who,* so an additional retrieval is not required at the verb.[4] Table 4 suggests that the model is overestimating the cost of the object-gap filling for this experiment. It is possible to adjust the latency factor to reduce the mismatch considerably, but we prefer for now to explore the implications of a consistent set of parameters across all of our simulations, until we adopt a principled basis for modeling individual differences.

### *5.3. Simulation 2: Estimating the time for working-memory retrieval*

McElree et al. (2003) applied McElree's (1993, 1998) theory of short-term memory to sentence processing. The major assumptions are consistent with the ACT–R model: There is a very limited focus of attention, and items outside the focus must be retrieved. McElree et al. (2003) tested this idea in sentence processing using a speed–accuracy–trade-off (SAT) paradigm designed to provide an assessment of the cost of retrieving an element outside the focus. Participants read sentences in a rapid serial visual presentation paradigm at 250 msec per word

Table 4
Contrasts in reading times (msec) for Subject Relatives and Object Realtives (data from Experiment 1, Grodner & Gibson, 2005)

| Contrast | Model | Data |
|---|---|---|
| Object Relative vs. Subject Relative | | |
| At main verb | 27 | –4 |
| At embedded verb | 105 | 67 |
| Interaction of verb and extraction | 78 | 71 |

and responded to a tone at various time lags after the onset of the final word of the sentence. The task was to perform a semantic acceptability judgment, which required the participants to successfully parse the sentence. Examples of their materials and conditions are given in (9):

(9) a. [None] The book ripped.
b. [OR] The book that the editor admired ripped.
c. [PP–OR] The book from the prestigious press that the editor admired ripped.
d. [OR–SR] The book that the editor who quit the journal admired ripped.
e. [OR–OR] The book that the editor who the receptionist married admired ripped.

The four conditions manipulate the amount and type of interpolated material between the main verb *ripped* and the subject *book* (None = no interpolation, PP-OR = prepositional phrase and OR, OR–SR = subject relative embedded within OR, OR–OR = double embedded OR). Their SAT analysis provides an estimate of the time to retrieve the displaced subject NP at the main verb.

The analysis revealed three distinct retrieval times: one fast time for the None condition, an intermediate time for the OR, PP–OR, and OR–SR conditions, and a third longer time for the OR–OR condition. Note that the OR–OR construction is the notoriously difficult classic double center embedding, and McElree et al. (2003) suggested that the inflated OR–OR times result from additional recovery times from misparsing this construction and do not provide a clean estimate of retrieval times. Fig. 6 shows the results, without the OR–OR condition. (We further discuss center embeddings later in the article.)

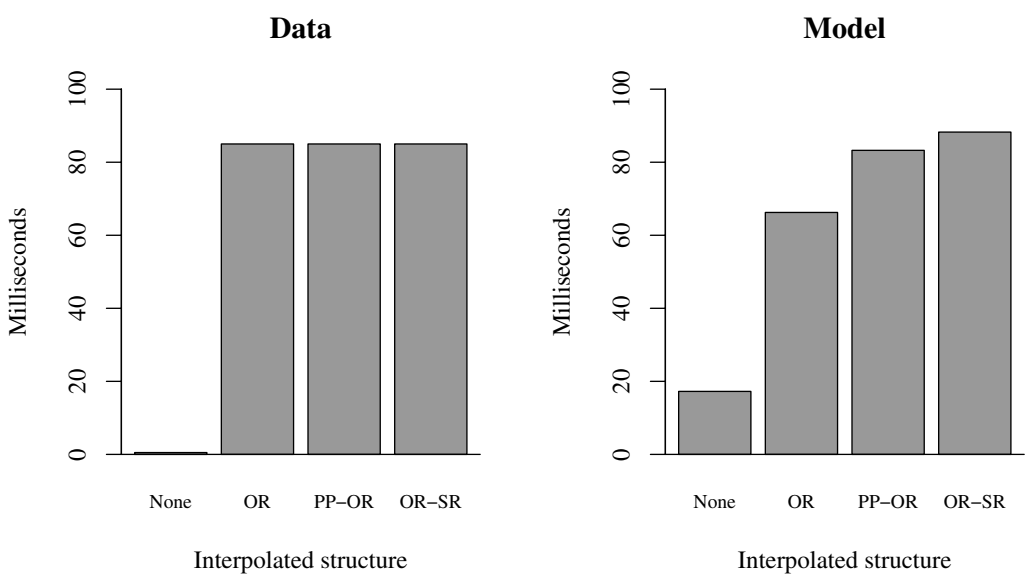


Fig. 6. Model-generated and observed processing times on the final verb as a function of interpolated structures. Data from Experiment 2, McElree et al. (2003). None = no interpolation, OR = interpolated object relative, OR-PP = interpolated prepositional phrase and object relative, OR-SR = interpolated subject relative embedded within object relative.

Table 5
Contrast in processing times (msec) at final verb (data from Experiment 2, McElree et al., 2003)

| Contrast | Model | Data Estimate |
|---|---|---|
| Distal vs. local attachment | 62 | 85 |
| PP–OR vs. OR | 17 | 0 |
| OR–SR vs. PP–OR | 5 | 0 |

McElree et al. (2003) argued that the 85-msec difference in processing dynamics between the None and OR/PP–OR/OR–SR conditions provides an estimate of the time required for the working-memory retrieval of the subject NP. The ACT–R model differs somewhat from the structural assumptions of the McElree et al. (2003) model, in that there is still a retrieval required in the None condition. Nevertheless, the attachment time in the None condition is much faster (by 62 msec) than the other three conditions. We characterize the contrast as "distal versus local" rather than "retrieval versus no retrieval."

Perhaps the most striking aspect of the McElree et al. (2003) data is the lack of difference between the PP–OR, OR–SR, and OR conditions. Table 5 summarizes this effect and shows that the ACT–R model captures this basic interaction, though still predicting some minor differences.

### 5.4. *Simulation 3: Effects of interpolated material on main verbs and embedded verbs*

Grodner and Gibson (in press) conducted a follow-up to their relative clause experiment that lets us further test the model's predictions concerning the effects of interpolated material on main verbs (as in McElree et al., 2003) and embedded verbs. Examples of their materials are given in (10)

(10) a. [main:None] The nurse *supervised* the administrator while …
b. [main:PP] The nurse from the clinic *supervised* the administrator …
c. [main:RC] The nurse who was from the clinic *supervised* the administrator while …
d. [embedded:None] The administrator who the nurse *supervised* scolded the medic while …
e. [embedded:PP] The administrator who the nurse from the clinic *supervised* scolded the medic while …
f. [embedded:RC] The administrator who the nurse who was from the clinic *supervised* scolded the medic while …

The conditions contrast the effect of modifying the subject of either the main verb or the embedded verb: The conditions None (no modification), PP modification, and RC modification are crossed with main verb and embedded verb (*supervised* in the previous example). These structures thus provide another test of both how the model processes relative clauses (the main vs. embedded contrast) and how the model handles nonlocal attachments across different structures (the modifier type contrasts).

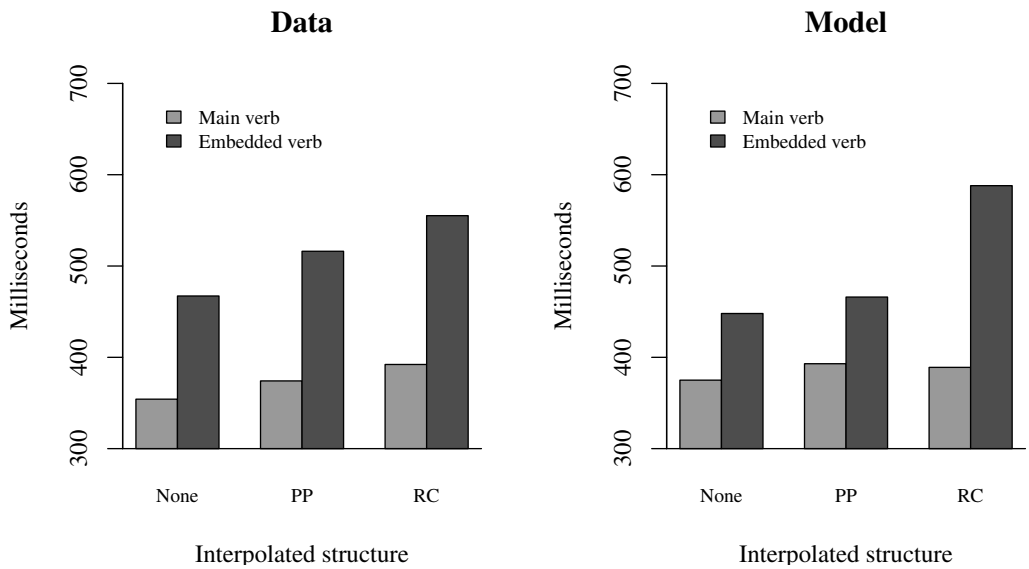


Fig. 7. Model-generated and observed self-paced reading times on main verbs and embedded verbs as a function of interpolated structure (different modifiers of the verb's subject). Data from Experiment 2, Grodner and Gibson (2005). None = no interpolated structure, PP = prepositional phrase, RC = subject relative clause.

Fig. 7 shows the experimental data and the model-generated reading times. There are a couple of important contrasts to focus on; Table 6 summarizes these. First is the effect of embedding for the OR: a general increase in reading times for the embedded verbs versus the main verbs, as discussed previously. Second, there is an effect of interpolated structure (modifier type), with the relative clause modification showing the largest effect. Furthermore, the main verb reading times are remarkably flat, which is consistent with the result of McElree et al. (2003) discussed previously. The ACT–R model also predicts an interaction of verb type and modification as Fig. 7 and Table 6 reveal. The interaction arises for two reasons: There is increased retrieval interference at the embedded verb (because the expectation for the main verb overlaps in retrieval cues), and the effect of decay is felt on both the retrieval of the expected IP

Table 6
Contrasts in reading times (msec) for main verbs and embedded verbs as a function of subject modification (data from Experiment 2, Grodner & Gibson, 2005)

| Contrast | Model | Data |
|---|---|---|
| Embedded vs. main | 139 | 115 |
| Distal vs. local | 49 | 48 |
| PP modifier vs. none | | |
| Main verb | 20 | 18 |
| Embedded verb | 49 | 18 |
| RC modifier vs. none | | |
| Main verb | 38 | 14 |
| Embedded verb | 88 | 140 |

node and the retrieval of the object-gap filler. This prediction is consistent with the human data, although the interaction was not statistically significant.

### 5.5. *Simulation 4: Effects of length and interference in garden paths and their unambiguous controls*

The experiments we have considered thus far provide good quantitative tests of the model's account of distal versus local attachments under various conditions, but do not yet begin to clearly distinguish the effects of decay and interference, both of which are fundamental parts of ACT–R's (and thus the model's) theory of memory retrieval. Van Dyke and Lewis (2003) recently used locally ambiguous garden-path sentences in addition to unambiguous structures in an attempt to tease apart the effects.

#### 5.5.1. *Materials and design*

Examples of the materials are given in (11). There were two manipulations: a structural interference ([int]) and distance ([short]) manipulation, crossed with an ambiguity ([(un)ambig]) manipulation.

(11) a. [unambig:short] The assistant forgot that the student *was standing* in the hallway.
b. [unambig:low-int] The assistant forgot that the student who was waiting for the exam *was standing* in the hallway.
c. [unambig:high-int] The assistant forgot that the student who knew that the exam was important *was standing* in the hallway.
d. [ambig:short The assistant forgot the student *was standing* in the hallway.
e. [ambig:low-int The assistant forgot the student who was waiting for the exam *was standing* in the hallway.
f. [ambig:high-int The assistant forgot the student who knew that the exam was important *was standing* in the hallway.

The critical region of interest is the same in all conditions: The reading time on the embedded verb *was standing*. Note that the embedded clause is a sentential complement, not a relative clause in this case. We are interested in the effect of material interpolated between the subject NP *the student* and the embedded verb.

Consider the unambiguous conditions first. In the short condition, nothing intervenes between the subject NP and the verb. In the low-interference condition, there is an intervening SR clause with a prepositional phrase. The high-interference condition is equally long, but instead of a prepositional phrase, it includes another intervening sentential complement, which provides additional structural interference. Van Dyke and Lewis (2003) argued that the contrast between the short- and low-interference conditions provides an estimate of a *distance* effect, whereas the contrast between the low-interference and high-interference conditions provides an estimate of an *interference* effect.

Now consider the ambiguous conditions. Van Dyke and Lewis (2003) reasoned that garden-path structures will show *additional* distance effects, but *no additional* interference effects. The argument is as follows. Distance effects should show up most clearly when memory traces are left to decay without intervening retrievals and processing. A garden-path sentence

with a clear preference for one structure over another provides just such a situation: The dispreferred structure is left to decay without additional processing during the ambiguous region (under a serial parsing account). However, the interference effect should be about the same, because the intervening structures in the ambiguous region are the same.

To clear: What is at issue here is the effect of distance and interference on the garden-path effect itself. The prediction is not that garden-path structures will show no interference effects; rather, relative to their unambiguous controls, there is no *additional* interference. In contrast, the prediction for distance is the opposite. Relative to their unambiguous controls, garden-path structures *will* show an additional distance effect.

The ambiguous materials in (11) use the classic subject–object garden path (e.g., Ferreira & Henderson, 1991) with lexical material carefully chosen to bias toward the simple object NP reading (*the student* is initially taken to be the object of *forgot,* rather than the subject of an upcoming embedded clause). Thus, the critical region *was standing* in the ambiguous conditions now serves to disambiguate toward the sentential complement structure, and it is here that we expect the garden-path effect to arise (Ferreira & Henderson, 1991).

#### *5.5.2. Empirical and modeling results*

The results of Van Dyke and Lewis (2003)'s self-paced reading study confirm these predictions: There was an effect of interference on the unambiguous constructions but almost no effect of interference on the garden-path effect. In contrast, there was no distance effect on the unambiguous constructions, but there was a significant distance effect on the garden-path effect—a clear cross-over interaction between effects of distance and interaction on reanalysis and attachment.

In the ACT–R model, both the noun phrase and sentential complement structures are generated at the locally ambiguous verb *forgot,* but lexical access delivers the NP complement argument structure first, so it is pursued in the parse (realizing a simple race model of ambiguity resolution), whereas the other structure continues to decay without further processing. At the critical disambiguating verb, this discarded structure must be retrieved.

Fig. 8 shows the human times and the model-generated times at the critical verb for the six conditions of the experiment (the times are residuals after regressing out length and word position effects). (The figure also shows times for another region in the unambiguous constructions; we return to this data in the discussion of syntactic load effects).

This is a fairly complex pattern of data, and there are several important contrasts. The contrasts are summarized in Table 7. First, there is a main effect of ambiguity—a garden-path effect of 61 msec. The model predicts this effect because of the additional retrieval time taken in the ambiguous case to reactive the discarded memory item. Second, there is also a main effect of interference (comparing the high- and low-interference conditions); the model predicts this because of additional retrieval interference from the intervening IP node in the high-interference structure.

Looking in more detail at the effects of interference, the model correctly predicts a greater effect of interference on attachment than on reanalysis. Looking in more detail at the effects of length, the model correctly predicts that length effects are greater than interference effects for reanalysis. This is because in the unambiguous case the pursued structure has one additional retrieval as a result of participating in the parse, and this reduces the effects of decay.

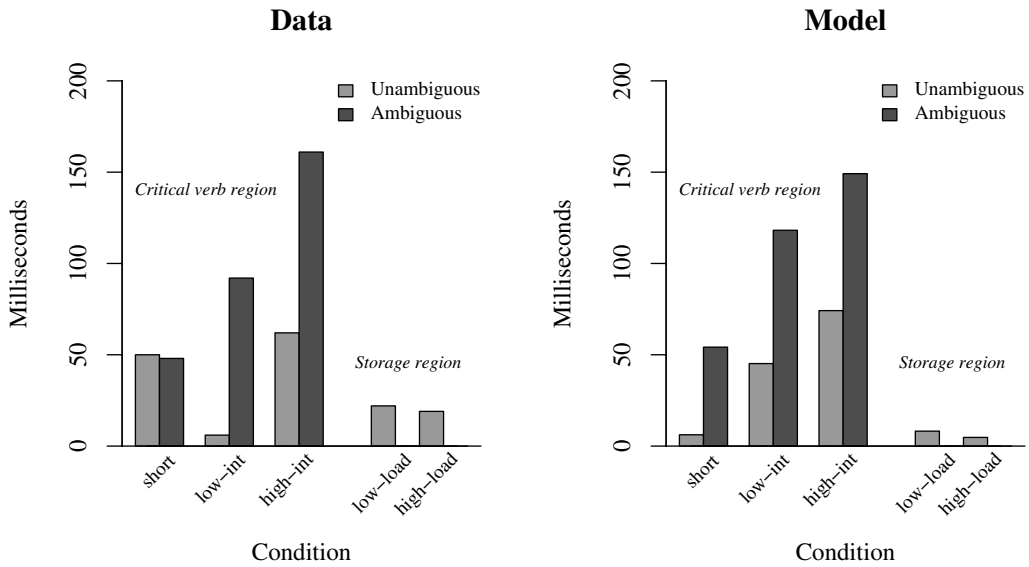


Fig. 8. Model-generated and observed self-paced residual reading times on the critical verb in unambiguous and ambiguous (garden-path) structures as a function of length and structural interference, and reading times in the storage region (Region 2 in the original experiment) as a function of syntactic storage load. Human data from Experiment 4, Van Dyke and Lewis (2003). Short = zero length ambiguous region, low-int(erference) = long ambiguous region with low structural interference, high-int = long ambiguous region with high structural interference. Low-load = low syntactic storage load, high-load = high syntactic storage load.

But we do see a discrepancy—see the first two data points in Fig. 8 and the third part of Table 7. The striking aspect of the data is the complete absence of a length effect on the unambiguous conditions. In fact, the effect numerically goes in the opposite direction: a slight speedup. As a result, the model also underpredicts the effect of length on reanalysis. In short, the actual human data is a more striking reflection of the qualitative form of the prediction than the model behavior is. We believe that the model data overestimates the length effect because it underesti-

Table 7
Contrasts in reading times (msec) at the critical verb for ambiguous and unambiguous structures as a function of length and interference (data from Experiment 4, Van Dyke & Lewis, 2003)

| Contrast | Model | Data |
|---|---|---|
| Aggregate garden-path effect | 65 | 61 |
| Interference effect | | |
| Aggregate | 30 | 62 |
| On garden-path effect | 2 | 13 |
| On attachment | 29 | 56 |
| Length effect | | |
| Aggregate | 52 | 0 |
| On garden-path effect | 25 | 88 |
| On attachment | 39 | –44 |

mates the effect that participating in the parse has on activation level. In particular, there may be additional retrievals associated with semantic processing that are not captured by this model.

### 5.6. Simulation 5: Effects of storage load

The ACT–R model posits no separate role for "storage" versus "processing" or "integration" costs, in contrast to the Just and Carpenter (1992) models and the storage-based theories of Gibson (1998, 2000). There is only retrieval cost as modulated by associative interference and activation fluctuation. Theoretically, what the ACT–R model claims is that no processing effort is expended in keeping prior constituents active: They simply decay, and the difficulty in reactivating them is a function of how much they have decayed and how effective and discriminating the retrieval cues are. Thus, there is not a general prediction of storage-load effects—instead, there is a more specific prediction of similarity-based retrieval interference.

Grodner, Gibson, and Tunstall (2002) reported a self-paced reading experiment that manipulated syntactic load to test the effects of syntactic storage costs on processing of ambiguous and unambiguous constructions. We focus here on their unambiguous constructions, examples of which are given in (12):

(12) a. [SC/low load] The witness thought that the evidence that was examined by the lawyer implicated his next-door neighbor.
b. [RC/high load] The witness who the evidence that was examined by the lawyer implicated seemed to be very nervous.

The region of interest here is *the evidence that was examined by the lawyer implicated*. In the low-load condition, this region is embedded in a sentential complement. In the high-load condition, this region is embedded in a relative clause, so that a prediction of the empty category must be maintained across the region. Under the storage model, the reading times should increase in this region.

The results, shown in Fig. 9 confirm the predictions of the storage account. We have divided up the regions of interest according to the regions in the original analysis. We collapsed together Regions 3 (*the evidence*) and 4 (*that was*) to take into account spillover effects from the reanalysis of SR to OR clause at the subject NP *the evidence* in (12-b).

The are several important things to note about the model-data comparison. This the first simulation we have presented that models reading times across regions of a sentence that include constructions other than verbs. Such comparisons are inherently problematic because they compare across syntactic and semantic types, but we note that the model is doing a reasonable job of accounting for the basic patterns with syntactic processing alone. This is broadly consistent with the success that Grodner and Gibson (in press) had in modeling reading times with their locality-integration model.

More to the point, however, is whether the model succeeds in capturing the differences between the low- and high-load conditions. Fig. 9 clearly reveals that the success is mixed. In the region *the evidence that was* (Regions 3 and 4 in the original analysis), the model predicts greater reading times in the high-load condition due in part to processing involved in reanalyzing from a subject to OR clause. At the embedded-verb region, the model predicts in-

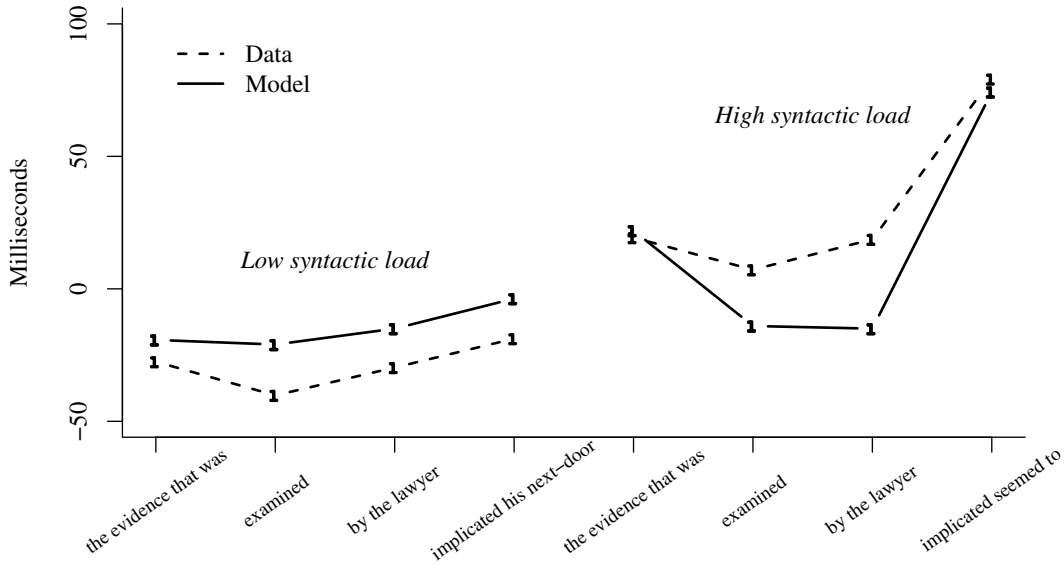


Fig. 9. Model-generated and observed residual self-paced reading times in regions of low and high syntactic storage load. Human data from Experiment 1, Grodner et al. (2002).

creased times associated with filling the object gap in the RC condition, for all the reasons explained earlier.

There is also, perhaps surprisingly, a small load effect at *examined;* this is a retrieval-interference effect due to the additional expected IP. Thus, we see that the model can in principle capture some storage effects as retrieval effects. The model does not presently account for the size of the effect, or the effect on the agent PP *by the lawyer,* although the latter could plausibly be spillover. Table 8 summarizes the contrasts.

One way to distinguish the retrieval interference and storage accounts is to find cases where the locality–storage theory predicts an increase in reading time during the maintenance of a stored prediction, but *before* the critical retrieval. The materials in the previously discussed

Table 8
Contrasts in reading times (msec) as a function of syntactic storage load (data from Grodner et al., 2002, and Van Dyke & Lewis, 2003)

| Contrast | Model | Data |
|---|---|---|
| Load effect (Grodner et al., 2002) | | |
| Aggregate (not including *implicated*) | 48 | 142 |
| *the evidence that was* | 41 | 46 |
| *examined* | 7 | 47 |
| *by the lawyer* | 0 | 49 |
| *implicated…* | 78 | 98 |
| Load effect (Van Dyke & Lewis, 2003) | | |
| Storage region | –3 | –3 |

Van Dyke and Lewis (2003) experiment include such a condition. Van Dyke and Lewis detailed how the storage account predicts increased reading time across the italicized regions of the long unambiguous materials (Region 2 in the original analysis):

(13) a. [unambig:long] The assistant forgot that the student who *was waiting for the exam* was standing in the hallway.
b. [unambig:int] The assistant forgot that the student who *knew that the exam was important* was standing in the hallway.

The reading times revealed no significant load differences, however (see the storage region points in Fig. 8 and the contrast in Table 8). The model actually predicts a tiny decrease in time over the high-load region.

In sum, the model predicts reasonably well the pattern across regions in the two load conditions of Grodner et al. (2002), but only partially accounts for the size of the load effect. However, data from Van Dyke and Lewis (2003) suggests that the explanation of load effects as retrieval-interference effects may in fact be the correct one. A more complete account of these patterns awaits further development of the model, as we describe in the General Discussion section.

## 6. Center embedding and the problem of serial order discrimination

We turn now to a set of contrasts concerning center embedding. Before we consider the center-embedded structures, it is important to confirm that the model does not predict dramatic increases in reading times for deep right embeddings.

### *6.1. Deep right embeddings*

We ran the model on the following structures: deep right-branching sentential complements, and deep right-branching prepositional phrase modifiers.

(14) a. The boy thought that the man said$_1$ that the editor thought$_2$ that the writer believed$_3$ that the medic thought$_4$ that the assistant admired the student.
b. The medic with$_1$ the assistant with$_2$ the dog with$_3$ the editor with$_4$ the duck …

We are interested in the attachment times for the verbs numbered 1 to 4 and the prepositions numbered 1 to 4. The question is whether there is a significant increase as depth of branching goes up. Table 9 gives the results, setting the attachment time of the first SC or PP at zero as a baseline; thus the other times are increases over this baseline. For the embedded sentential complements, there is a tiny increase of about 5 to 6 msec at each depth. This is a small effect of proactive retrieval interference. For the prepositional phrases, there is a larger but still modest increase (of 10 msec, decreasing to 6 msec) at successive attachments. This also represents proactive retrieval interference from additional potential attachment sites. The reason that the PP attachment generates longer increases is informative about the structure of the model: The sentential complements are arguments, and once attached, the verb no longer expects a complement. This keeps the "fan" of the retrieval cue at a constant across the embedding. The re-

Table 9
Memory-retrieval times in the model as a function of right-branching depth

| Level | Embedded Sentential Complements (at the verb) | Embedded Noun-Modifying Prepositional Phrases (at the preposition) |
|---|---|---|
| 1 | 0 | 0 |
| 2 | 6 | 10 |
| 3 | 11 | 18 |
| 4 | 16 | 24 |

*Note.* The first level is the baseline and set to zero.

trieval cue for the PP modifier attachment, however, is simply a constituent of category NP (or VP), and the number of potential attachment sites continues to increase throughout the sentence.

It seems unlikely that the small increases in the sentential complement embeddings could be detected empirically—even at greater depths, because the size of the effect continues to decrease as a result of the nonlinearity of Equation 3. The PP attachment times might be detectable, and further empirical work is warranted to see if the model's predictions are quantitatively correct. But for present purposes the results of the simulations support the qualitative claim that deep right branching does not produce dramatic effects in processing times.

### 6.2. *The "no serial order representation" hypothesis*

As part of a functional analysis of the demands on serial order representation in parsing, Lewis (2000) raised the following possibility: Perhaps there is no serial order representation or serial order mechanism in human sentence processing at all—parsing is based on nothing more than cue-based associative retrievals. This fits well with the findings of McElree (McElree & Dosher, 1989, 1993), which point to rapid, parallel access from working memory of *item* information, but a slow, serial access of relative *order* information.[5]

The ACT–R model embodies this hypothesis. In short, the model proposes that sentence processing relies heavily on discriminating retrieval cues and the distinction between the present and the past. For the most part, this works well—where it potentially fails is in cases that have the following structure, where $\beta$ is a word that triggers the retrieval of either $\alpha_1$ or $\alpha_2$, where $\alpha_1$ and $\alpha_2$ cannot be distinguished except on the basis of their relative serial positions:

(15) $\alpha_1 \ldots \alpha_2 \ldots \beta$

In a case such as (15) previously, memory retrieval will succeed for whichever of $\alpha_1$ or $\alpha_2$ has the highest activation value. All things being equal, this will be the most recent element due to decay, which is correct for nested structures (but not for cross-serial[6]).

Double self-embeddings are precisely cases that fit this general schema. Consider the extreme case of the classic double-embedded OR clause:

(16) The salmon that the man that the dog bit smoked tasted good.

These structures are notoriously difficult to comprehend (Blauberg & Braine, 1974; Blumenthal, 1966; Hakes & Cairns, 1970; Marks, 1968; Miller & Isard, 1964; Wang, 1970). Several theorists have proposed accounts of the processing difficulty in terms of the demand center embedding places on memory resources of some kind (e.g., Gibson, 1991; Kimball, 1973; Lewis, 1996; Miller & Isard, 1964; Stabler, 1994; Yngve, 1960). We identify here another basic problem with processing such structures: There are multiple attachment points that require distinguishing candidate constituents primarily or exclusively on the basis of their relative serial order. In particular, in (16), there are two active fillers and two predicted embedded finite clauses that must be properly distinguished by serial order to make the correct attachments at the verbs.

### 6.3. Simulation 6: Toward a graded taxonomy of center embeddings

To test how much various forms of center embedding depend on serial order discrimination, we ran a series of simulations of the model with activation noise.[7] One of ACT–R's standard assumptions is that there is always some noise added to the activation value of a chunk at each retrieval. This noise value is generated from a logistic distribution with mean 0 and variance given by Equation 5:

$$\sigma^2 = \frac{\pi^2}{3} s^2 \tag{5}$$

where $s$ is a parameter. This activation noise permits the use of ACT–R to model RT distributions and various kinds of memory errors; it plays a key role in the Anderson and Matessa (1997) model of list memory, for example, where $s$ ranged from 0.2 to 0.5. We set the noise parameter $s$ to 0.3.

One of the empirical puzzles about double center embedding pointed out by several theorists (Cowper, 1976; Gibson, 1991), and most fully addressed by Gibson (1998), is the existence of double clausal center embeddings that do not yield the same breakdown associated with (16). For example, subject sentences such as (17-a) are noticeably easier that the classic double ORs, as are sentential complements embedded within relatives (17-b; Gibson, 1998):

(17) a. That the food that the medic ordered tasted good pleased her.
b. The news that the assistant that the lawyer hired left the firm surprised everyone.

We were therefore interested in the model's performance on these structures relative to the classic cases. The full set of structures was as follows:

(18) a. [OR/OR] *object relative within object relative*
The editor that the writer that the dog chased scolded supervised the assistants.
b. [SR/OR] *subject relative within object relative*
The medic who the dog that bit the reporter chased admired the writer.
c. [RC/SC] *sentential complement within object relative clause*
The reporter who the claim that the editor admired the medic amused sent the gift.
d. [SC/SC] *sentential complement within sentential complement*

The claim that the news that the reporter admired the medic amused the editor upset everyone.

e. [SC/RC] *object relative within sentential complement*
The claim that the reporter who the editor admired sent the gift amused the writer.

f. [SS] *object relative within subject sentence*
That the reporter who the editor married liked the medic was surprising.

The right-branching structure (14-a) was included as a baseline.

For each structure, we ran the model 100 times and determined the number of times that the model arrived at the correct parse, as well as the number of times that a parse led to a memory-retrieval failure. Not all incorrect parses led to memory-retrieval failures, but in some cases an incorrect attachment earlier in the sentence eliminated the correct candidate for a retrieval later in the sentence. This happened, for example, when the main verb prediction was retrieved too early—perhaps at the second verb in a double center embedding—and was not available when the final (main) verb was actually read.

Fig. 10 plots the results. There are a number of interesting results. The most striking effect is the qualitative contrast between the two difficult double relative clause embeddings and the two easier double embeddings introduced previously.

A second interesting contrast is between the RC/SC and SC/RC embedding: the former led to some misparses. The RC/SC and SC/RC contrast was first pointed out by Gibson (1998); the locality theory predicts the contrast because the filler-gap relation is over a much longer distance in the RC/SC construction than in the SC/RC construction. We visit this contrast again in a moment.

A third interesting and more subtle contrast is between the OR/OR and SR/OR structures. The prediction that the SR/OR structure is easier is consistent with the results of McElree et al.

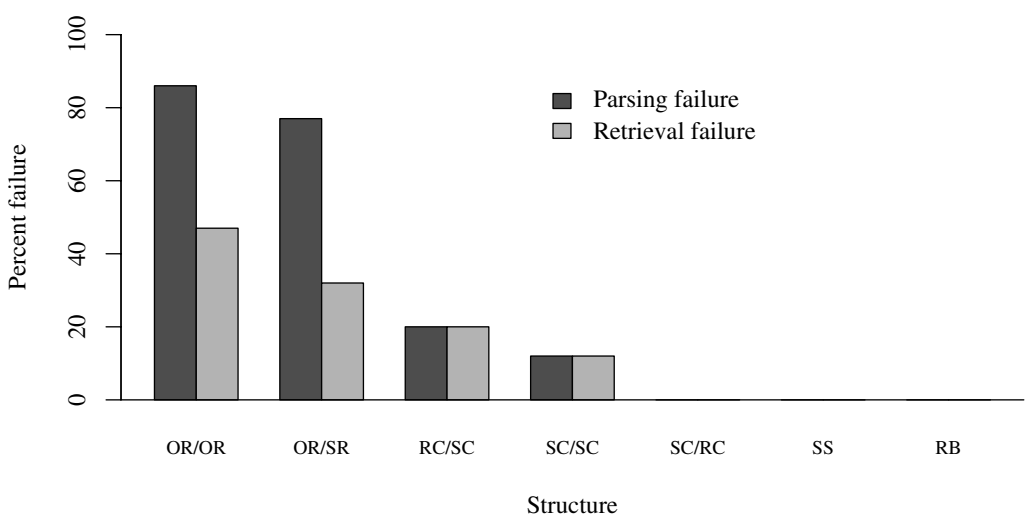


Fig. 10. The behavior of the model on double-center-embeddings, when activation noise is added. Parsing failure is the percentage (out of 100) of over-all failed parses. Retrieval failure is the percentage of parses that led to working-memory retrieval failures.

(2003) who found that the OR/OR structures produced slower dynamics than the SR–OR structures. Their description of how this might have happened in their SAT paradigm is remarkably consistent with our account:

> Misanalysis of any one of the (verb-argument) dependencies will leave a verb stranded without an argument or produce a semantic anomaly. Time course differences of the form seen here could arise from successful reanalysis following misanalysis on a proportion of times. (p. 81)

"Misanalyzing" dependencies is precisely what the model is doing here. A little over half of these misanalyses do result in memory-retrieval failures that result from leaving a verb "stranded." Finally, note the positioning of the SC/SC contrast, somewhere midway between the RC/SC and SC/RC. The increased errors on SC/SC relative to SC/RC are a result of the greater discriminability of the clauses in the SC/RC case—the relative clause carries an additional "gap" feature, which permits a bit more retrieval-time discrimination.

One way to view the ordering of the structures in Fig. 10 is that the failure rates provide a quantitative measure of the degree to which successful parsing of these structures depends on an accurate representation of serial order. This overall ordering is consistent with the acceptability ratings reported by Gibson and Thomas (1997) as well, with the exception that they found that the SC/SC structures patterned with RC/SC—the SC/SC structures were not more difficult.

However, it is important to note that the results of Fig. 10 report parsing failure rates only; they do not take into account reading-time differences. The model's predicted reading times across the three critical verbs for the SC/RC, RC/SC, and SC/SC structures are given in Table 10. Not surprisingly, the highest reading times occur in the SC/RC and RC/SC structures. By far the highest predicted reading time occurs in the RC/SC structure on the second verb. The increased distance of the filler-gap relation means that there is more time for the gap filler's activation to decay—and crucially, there are no intervening reactivations to mitigate this effect. This is consistent with the locality account.

Although we do not yet have a precise quantitative theory that predicts acceptability judgments, we believe a plausible one will take into account *both* failure rates and latencies. The latencies will tend to dominate in structures with low failure rates. Under such an account, it would not be surprising to see the SC/SC structure and SC/RC structures pattern together. The reason is that although the SC/SC structures generated somewhat more failures than SC/RC structures, they also generated somewhat lower reading times when the parses succeeded (see Table 10 and Fig. 10).

Although we believe this is the first explicit serial order discrimination account of difficult center embeddings in the literature, it is very much in the spirit of accounts proposed by a num-

Table 10
Predicted reading times for verbs on SC (sentential complements) and RC (relative clause) mixes

| Attachment time | RC/SC | SC/RC | SC/SC |
|---|---|---|---|
| Verb 1 | 163 | 278 | 163 |
| Verb 2 | 371 | 210 | 211 |
| Verb 3 | 238 | 237 | 211 |

ber of theorists over the years that explain the difficulty in terms of similarity and discrimination rather than memory overload (Bever, 1970; Chomsky, 1965).[7]

## 7. Summary and general discussion

### *7.1. Summary: Theory and modeling results*

We presented a theory and computational model of the working-memory processes that subserve sentence comprehension. The theory construes parsing as an efficient series of guided memory retrievals. The basic tenets of the theory are simple and follow directly from ACT–R: Sentence processing is controlled by production rules representing the skill of efficient parsing; the activation of memory elements fluctuates as a function of usage and decay; and memory retrieval is subject to associative interference. The computational model represents a novel integration and application of these theoretical concepts to sentence processing.

The model was successful in accounting quantitatively for reading times from five separate experiments, which test in various ways the model's assumptions about memory and processing. The model also led naturally to a novel account of the difficulty of processing center-embedded structures and yielded a graded taxonomy of double center embeddings ranging from easy to difficult.

These results provide support for the claim that the ACT–R account of memory retrieval, developed in domains outside real-time sentence processing, also provides a good account of what is happening in the moment-by-moment retrievals of parsing.

Table 1 summarizes the simulation results and underlying quantitative parameters. Only the single scaling factor was estimated to fit the data, and this factor was set once for all the simulations. All other parameters were set based on ACT–R defaults and prior work on memory interference.

Although the fit to the data is reasonably good for a single-parameter model, it does provide clear cues for how to improve the model: In general, the model appears to *overestimate length effects* and *underestimate interference effects*. Any structural or quantitative change to the model that moves in the direction of decreased emphasis on decay and increased emphasis on interference would likely yield better fits. This bodes well for extending the theory to semantic and discourse processing, which can only add extra retrievals and therefore reduce the effects of decay, but cannot reduce interference.

### *7.2. Relation to other models and theories*

Although the ACT–R-based theory is conceptually simple, it has interesting relations to a wide variety of other approaches in the literature and owes a large intellectual debt to much empirical and modeling work in psycholinguistics and cognitive architectures. We consider some of these debts, spending the most time relating the approach to Gibson's (1998, 2000) DLT, which has been one of the most successful attempts to date to account for data such as that presented here.

*7.2.1. Relation to DLT*

There are several views one might take on the relation of the ACT–R model to DLT. One view focuses on the fact that the ACT–R model is a decay model and takes the model to simply be an elaborate implementation of locality theory. A second, related, view takes the ACT–R account to be a cognitively plausible, lower level computational realization of locality principles. We believe both views are incorrect. The reason is that DLT and the ACT–R-based accounts diverge substantially, both theoretically and empirically. There are at least four key points of divergence:

1. *Fluctuating activation* is not the same theoretical concept as *locality*. A locality framework leads the theorist to ask the question: What linguistic units should be counted as contributing to locality cost? Indeed, attempting to answer this question has driven some of the work in the DLT framework. ACT–R's activation fluctuation, in contrast, does not require the postulation of a unit of locality cost. The only currency is time. One particularly interesting difference is that DLT posits a difference in integration cost as a function of the referential types of the intervening elements over which integration is made; this is not an explicit part of the ACT–R model.

Furthermore, what counts in the ACT–R model is not the distance to the dependent, but rather the history of retrievals. In fact, ACT–R's activation equation is not properly described as a *decay* equation; rather, it predicts both power law learning and forgetting (Anderson & Lebiere, 1998): The activation can in fact rise over time with additional retrievals.

Of course, we are not denying that there is a clear relation between decay and locality. All things being equal, longer distances (over whatever unit) usually mean more time, which sometimes (but not always) means activation decay. This association of length and decay has a long history in psycholinguistics: It is why Chomsky (1965) suggested that heaviness and length effects point to memory decay as the culprit, and why Gibson (1998) also suggested decay might be behind distance effects. We believe they are both right.

2. *Associative retrieval interference* is not the same theoretical concept as *storage cost*. DLT posits separate integration and storage cost components; this dual processing–storage framework is more consistent with the Just and Carpenter (1992) model than ACT–R. The ACT–R model expends no resource keeping memory items active (of course, it can strategically work to keep items active via rehearsal, but there is not time to do that in language processing). The ACT–R model posits no computational resource consumed except time.

3. *Inadequate serial order discrimination* and *high locality-based integration costs* are different theoretical explanations for the difficulty associated with the notoriously bad double center embeddings. As we outlined previously, the discrimination account does not supplant the distance and interference effects, which the model also captures in the reading times on those rare occasions that it parses them correctly. But we believe that the discrimination account provides a plausible explanation for the rather dramatic qualitative effect of the double embeddings, and complements in an important way a reading-time–locality-based explanation.

4. The ACT–R model specifies computational structure and quantitative parameters on an independent basis that must be added to DLT to generate precise quantitative predictions. In particular, these include assumptions about the locality cost function (linear, nonlinear, etc.) and the combination function for cost at points of multiple integrations (e.g., whether the dis-

tances are summed then put through the cost function, or put through the cost function separately and then summed, etc.)

The empirical predictions of DLT and the ACT–R model reflect the theoretical divergence; we noted some of these throughout the article. To summarize:

1. The fluctuating activation of the ACT–R model (which can yield different levels of activation across identical distances) was critical for accounting for the differential effect of distance and interference on garden-path reanalysis and attachment (Simulation 4). (2) The ACT–R model makes different predictions from DLT about "storage effects" (Simulations 4 and 5). (3) The ACT–R model predicts a cross-over interaction of extraction type and verb (main vs. embedded) in the reading times on relative clauses (Simulation 1); DLT does not. (4) The ACT–R model predicts a greater effect of interpolated material on embedded relative clauses than main verbs (due to greater interference and processing of the object-gap relation); DLT predicts this interaction but to a smaller degree than the ACT–R model (Simulation 3). (5) The ACT–R model predicts a relatively flat effect of interpolated material on main verb reading times (Simulations 1–3); this is a direct result of the parallel associative retrieval mechanisms of the model. (6) The ACT–R model predicts probabilistic parsing failures in addition to longer reading times; DLT has not been used to generate such probabilistic predictions (Simulation 6; see Gibson & Thomas, 1999, for use of locality theory to account for binary acceptability contrasts). (7) Although we did not show such effects in the simulations in this article, the ACT–R model is capable of predicting "reverse" length effects—that is, *faster* reading times with increased length. Such apparently puzzling effects have been confirmed now in German (Konieczny, 2000; Vasishth, Cramer, Scheepers, & Wagner, 2003; Vasishth, Scheepers, Uszkoreit, Wagner, & Kruijff, 2004), Hindi (Vasishth, 2003a, 2003b), and Japanese (Gibson, personal communication, July 2003). We are currently developing models to account for data from these languages.

Finally, although we have not yet attempted to model differences due to referential type of intervening constituents, we expect differences in the predictions of the ACT–R model and DLT, which explicitly assumes that integration cost varies as a function of referential type. One important challenge for the ACT–R model is to account for the greater ease of processing double center embeddings with pronouns in the subjects of the most embedded clause (Bever, 1970; Gibson, 1998; Gordon et al., 2001; Kac, 1981). ACT–R would clearly be sensitive to the difference because the pronoun is processed more quickly than a full NP, but it seems unlikely that this would be enough to capture the contrast in the difficult embeddings. However, one possibility is to simply assume that nominals have base activations that reflect their referential type (perhaps their position in an accessibility hierarchy; see Givón, 1978). Such an assumption does not derive from the ACT–R theory, itself, unlike other aspects of the theory presented here—it would simply be an assumption about how certain linguistic distinctions are manifest in processing. This is precisely the status of the assumption in DLT.

#### *7.2.2. Working-memory theory*

As we pointed out earlier, the ACT–R theory is most consistent with the memory theory of McElree (1998), which emphasizes an extremely limited working-memory focus and parallel,

associative retrieval. What this theory adds is activation fluctuation and similarity-based retrieval interference. One minor, but possibly empirically very important, structural difference is that the McElree account assumes that the most recent item is available in the focus. That is not always possible in the ACT–R model, depending on how that item was processed. In general, the McElree SAT phenomena themselves will provide important future empirical tests of the model.

*7.2.2.1. Activation-based models.* Of the prior computational sentence-processing theories, the ACT–R model bears the most resemblance to other hybrid symbolic activation-based models, including the competition model of Stevenson (1994), the inhibition model of Vosse and Kempen (2000), and CAPS/CC-Reader (Just & Carpenter, 1992). The Stevenson model has something like an activation decay component and predicts recency effects in a similar way to ACT–R. It maintains a stack-like data structure, however, so we do not believe it will provide the foundation for a plausible account of unambiguous structures. The Vosse and Kempen (2000) unification space model has been applied to both ambiguous and unambiguous structures, and its account of difficult center embeddings is the closest to the one we proposed here. It has been used to model qualitative garden-path and embedding phenomena and has neurobiological support (Hagoort, 2003). However, it was not intended to—and is unable to (Kempen, personal communication, September 2004)—model reading-time data. Although all of these models use the concept of activation of memory elements, the ACT–R account has the virtue that it is founded on a set of independently motivated and extensively tested architectural principles of memory retrieval and controlled processing; the success of the reported modeling simulations attests to the benefits of this foundation.

*7.2.2.2. Interference sentence-processing theories.* Some recent approaches to sentence processing have emphasized the role of similarity-based interference (Gordon et al., 2001; Gordon, Hendrick, & Levine, 2002; Lewis, 1993, 1996; Van Dyke & Lewis, 2003). Lewis (1993, 1996) emphasized *storage* interference, whereas Lewis (1998b) and Gordon et al. (2001, 2002) emphasized *retrieval* interference. The latter emphasis is consistent with our current account.

It is worth reviewing briefly here why similarity-interference arises in the model, and how similarity is computed. Similarity interference arises at retrieval, and it is a direct consequence of Equation 3, which reduces the strength of association between a cue and a target as a function of the number of items associated with that cue. Reduced strength of association means reduced activation boost, which produces higher latencies and higher failure rates. The strong claim about this kind of interference is that it is *retrieval* interference, and it is purely a function of *cue overlap*. In this model, the retrieval cues are simply a subset of the features of a target item, so similarity interference is based purely on representational overlap, not general thematic connections (e.g., *cow–milk,* though ACT–R as an architecture admits of such associations).

It is therefore the nature of the cues that dictate the nature of the interference. In this model, we have realized only syntactic cues, which are used primarily to reactivate predicted structure to unify with. However, the model can accommodate a richer set of cues—for example, there may also be semantic cues derived from specific lexical constraints (e.g., the semantic con-

straints that a verb places on its subject). Apart from the extensive literature on the immediate effects of semantic constraints on ambiguity resolution (e.g., Trueswell et al., 1994), Van Dyke (2002) provided direct evidence for interference effects as a function of semantic cue overlap. Adopting such a richer set of cues may be on the path to providing an account for Gordon's (Gordon et al., 2001; Gordon, Hendrick, & Johnson, 2004) effects of NP type—at least on reading times at the verb, when retrieval interference should arise. However, some of the similarity effects—particularly those showing up before the verb (Gordon et al., 2004)—may be more properly understood as *encoding* interference, something that is missing from all current processing models.

#### *7.2.3. Degrees of freedom in the modeling*

A legitimate concern that can be raised about applying computational cognitive architectures to empirical data is whether the theorist has too many degrees of freedom in fitting data. We will not attempt to provide a general response to this concern (for insightful responses, see Anderson et al., 2005; Meyer & Kieras, 1997; Newell, 1990). We will attempt to address the issue with respect to this model.

As with any ACT–R model, there are two kinds of degrees of freedom: quantitative parameters, and the contents of production rules and declarative memory. The quantitative parameters, such as the decay rate, associative strength, and so on, are the easiest to map onto the traditional concept of degrees of freedom in statistical modeling. With respect to these parameters, our response is straightforward: We believe we have come as close as possible to zero-parameter predictions by adjusting only the scaling factor, adopting all other values from the ACT–R defaults and literature, and using the same parameter settings across multiple simulations.

With respect to the degrees of freedom inherent in the content of the production rules and declarative memory, we take advantage of the fact that in sentence processing we are blessed with a brutally fast domain with a well-studied content domain theory (syntax). The absolute time scale of the effects means that there is simply no room for adding or subtracting a production here and there to get a better fit. The overarching principle for language processing is "Do things as fast as possible," and that principle entirely guided the development of the parser. All of the model's processing is done in the most efficient manner that is functionally possible, and all of it boils down to two or three production firings and one or two memory retrievals.

#### *7.2.4. The value of architectures and prospects for the future*

Is ACT–R just an implementation language for theories? Again, we will not provide a general answer to the question, but simply note that the answer should be very clear in this case. ACT–R is not simply an implementation language for the theory; a comparison of Tables 2 and 3 reveals that this sentence-processing theory is itself best understood as the application of ACT–R theory (and thus, a particular set of theoretical principles) to the task of parsing.

There are many interesting possible directions for additional functional and empirical extensions to the model. It should be possible to extend the model to lower levels of processing to explore the interactions of lexical and syntactic processing; indeed much of the structure is already in place to do so. It should also be possible to extend the model to higher levels of semantic and discourse processing; an integration with the semantic interpretation model of Budiu and Anderson (2004) provides one possible avenue. Given prior ACT–R work on modeling in-

dividual differences, it should also be possible to use this model as a basis for accounting for performance differences due to variation in working-memory capacity.

Another important direction is to develop detailed models of the different experimental paradigms used in reading research, including eye-tracking, self-paced reading, central presentation, and speed–accuracy–trade-off designs. Developing such models requires making explicit linking assumptions about how linguistic processing is coordinated with eye-movement control and manual button pressing. It is one of the key advantages of working within an architecture such as ACT–R that such models can be developed in a theoretically constrained and computationally explicit manner. This also has the advantage of addressing a possible concern about relying too heavily on the self-paced moving-window paradigm, which might increase working-memory demands relative to natural reading.

Just as important, we believe the model opens up new empirical and theoretical questions to ask about sentence processing: questions about the precise nature of length effects, interference effects, and the relation of linguistic processing to more general cognitive processing. These are all direct benefits of pursuing the architectural approach to cognitive theory pioneered by Newell (Newell, 1990; Newell & Simon, 1972) and Anderson (Anderson, 1983, Anderson, 2005, this issue).

## Notes

1. We make a simplifying assumption in the present model that adjuncts are attached via a "modifier" slot rather than through Chomsky-adjunction, though this is not a critical assumption for the model's empirical predictions.
2. This presentation borrows from Crocker (1996), and the rules depart from our X-bar assumptions in some ways.
3. The distance-based explanation does not appear to work for head-final languages because the verb invariably appears clause finally in the embedded clause, and yet in these languages too there is an SO preference. Here, Korthals' frequency-based explanation is plausible.
4. One might argue that there should be additional retrievals for VP-internal subject traces; doing so would just increase times for all the verbs across the board and would not change the basic pattern.
5. McElree et al. (2003) adopt an alternative position: that the sentence processing has special-purpose serial order mechanisms that operate on a fast, parallel basis. We believe that the data that motivated this position can be handled by the model without serial information; but this remains a tenable position.
6. There are a number of linguistic phenomena that may appear at first to be insurmountable challenges to parsing without serial order. These include explicit order terms such as *former*, *latter*, and *respectively*, and cross-serial dependencies such as those that routinely appear in Dutch subordinate clauses with perception or causation verbs (Bach, Brown, & Marslen-Wilson, 1986; Kaan & Vasic, 2004; Steedman, 2000). Cross-serial dependencies are particularly interesting because they are syntactic constructions that appear to violate the standard nested most-recent-first ordering most naturally sup-

ported by a parser with activation decay. A full treatment of these phenomena is beyond the scope of this paper, but it is worth summarizing briefly what our initial analyses have revealed. First, it is important to carefully analyze what other cues may be available to the system to achieve the discrimination required—even in the absence of serial order information. Perhaps surprisingly, in a LC parse of Dutch cross-serial dependencies, there is enough information available at the verb to discriminate the predicted verbal categories *without* appeal to explicit serial order information, and without additional processing cost relative to nested structures. Furthermore, even in cases where explicit serial order information does seem to be required (as we believe is the case for order terms), the locus of that order information may be in a different memory representation from that supporting the incremental parse. Our hypothesis is that order terms like *former* and *latter*, or *first*, *second* and *third*, are discourse anaphors whose semantics are grounded in explicit relations in a discourse model, perhaps held in long-term working memory. In this sense, they may not be fundamentally different from spatial relation terms like *above* or *below* (and indeed such terms are often used in text discourse, as we have occasionally done above).

7. The other simulations presented in this paper do not have activation noise switched on because the goal there was to first understand the model's behavior without involving any random variation. We have started to experiment with activation noise in these other simulations, and preliminary results suggest that the results do not change significantly.

## Acknowledgments

We thank Daniel Grodner for generously sharing a prepublication version of his work with Ted Gibson. We are also grateful to Jerry Ball and three anonymous reviewers whose detailed comments significantly improved this article. Please send comments to either of us at rickl@umich.edu or vasishth@acm.org

## References

Aho, A. V., & Ullman, J. D. (1972). *The theory of parsing, translation, and compiling: Vol.1. Parsing*. Englewood Cliffs, NJ: Prentice Hall.

Altmann, G., & Steedman, M. (1988). Interaction with context during human sentence processing. *Cognition, 30,* 191–238.

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Anderson, J. R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science, 29,* 313–341.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2005). An integrated theory of mind. *Psychological Review.*

Anderson, J. R., & Lebiere, C. (1998). *Atomic components of thought*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Anderson, J. R., & Matessa, M. P. (1997). A production system theory of serial memory. *Psychological Review, 104,* 728–748.

Anderson, J. R., & Reder, L. M. (1999). The fan effect: New results and new theories. *Journal of Experimental Psychology: General, 128,* 186–197.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science, 2,* 396–408.

Bach, E., Brown, C., & Marslen-Wilson, W. (1986). Crossed and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes, 1,* 249–262.

Bader, M., & Lasser, I. (1994). German verb-final clauses and sentence processing: Evidence for immediate attachment. In C. Clifton Jr., L. Frazier, & K. Rayner (Eds.), *Perspectives in sentence processing* (pp. 225–242). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Bar-Hillel. (1953). A quasi-arithmetical notation for syntactic description. *Language, 29,* 47–58.

Bever, T. G. (1970). The cognitive basis for linguistic structures. In J. R. Hayes (Ed.), *Cognition and the development of language* (pp. 279–362). New York: Wiley.

Blauberg, M. S., & Braine, M. D. S. (1974). Short-term memory limitations on decoding self-embedded sentences. *Journal of Experimental Psychology, 102,* 745–748.

Blumenthal, A. L. (1966). Observations with self-embedded sentences. *Psychonomic Science, 6,* 453–454.

Bock, J. K. (1986). Syntactic persistence in language production. *Cognitive Psychology, 18,* 355–387.

Bock, J. K., & Griffin, Z. M. (2000). The persistence of structural priming: Transient activation or implicit learning? *Journal of Experimental Psychology: General*, *129,* 177–192.

Boland, J. E., & Blodgett, A. (2001). Understanding the constraints on syntactic generation: Lexical bias and discourse congruency effects on eye movements. *Journal of Memory and Language, 45,* 391–411.

Boland, J. E., Lewis, R. L., & Blodgett, A. (2004). *Distinguishing generation and selection of modifier attachments: Implications for lexicalized parsing.* Manuscript submitted for publication.

Budiu, R., & Anderson, J. R. (2004). Interpretation-based processing: A unified theory of semantic sentence comprehension. *Cognitive Science, 28,* 1–44.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Chomsky, N. (1980). Rules and representations. *Behavioral and Brain Sciences, 3*(1), 1–15.

Chomsky, N. (1986). *Barriers*. Cambridge, MA: MIT Press.

Chomsky, N., & Miller, G. A. (1963). Introduction to the formal analysis of natural languages. In D. R. Luce, R. R. Bush, & E. Galanter (Eds.), *Handbook of mathematical psychology* (Vol. 2; pp. 269–322). New York: John Wiley.

Clifton, C., Jr., & Duffy, S. A. (2001). Sentence and text comprehension: Roles of linguistic structure. *Annual Review of Psychology, 52,* 167–196.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Brain and Behavioral Sciences, 24,* 87–114.

Cowper, E. A. (1976). *Constraints on sentence complexity: A model for syntactic processing*. Unpublished doctoral dissertation, Brown University, Providence, RI.

Crocker, M. W. (1996). *Mechanisms for sentence processing* (Research paper EUCCS/RP-70). Edinburgh, Scotland: University of Edinburgh, Centre for Cognitive Science.

Crocker, M. W., & Brants, T. (2000). Wide-coverage probabilistic sentence processing. *Journal of Psycholinguistic Research, 29,* 647–669.

Dell, G. S., Burger, L. K., & Svec, W. R. (1997). Language production and serial order: A functional analysis and a model. *Psychological Review, 104,* 123–147.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14,* 179–211.

Ferreira, F., & Clifton, C., Jr. (1986). The independence of syntactic processing. *Journal of Memory and Language, 25,* 348–368.

Ferreira, F., & Henderson, J. M. (1991). Recovery from misanalyses of garden-path sentences. *Journal of Memory and Language, 30,* 725–745.

Fodor, J. A. (1983). *Modularity of mind: An essay on faculty psychology.* Cambridge, MA: MIT Press.

Ford, M. (1983). A method for obtaining measures of local parsing complexity throughout sentences. *Journal of Verbal Learning and Verbal Behavior, 22,* 203–218.

Frazier, L. (1987a). Sentence processing: A tutorial review. In M. Coltheart (Ed.), *Attention and performance xii* (pp. 559–585). Norwood, NJ: Lawrence Erlbaum Associates, Inc.

Frazier, L. (1987b). Syntactic processing: Evidence from Dutch. *Natural Language and Linguistic Theory, 5,* 519–560.

Frazier, L. (1989). Filler driven parsing: A study of gap filling in Dutch. *Journal of Memory and Language*, *28*, 331–334.

Frazier, L. (1995). Constraint satisfaction as a theory of sentence processing. *Journal of Psycholinguistic Research, 6,* 437–468.

Frazier, L. (1998). Getting there … slowly. *Journal of Psycholinguistic Research, 27,* 123–146.

Frazier, L., & Rayner, K. (1982). Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology, 14,* 178–210.

Gazdar, G. (1981). Unbounded dependencies and coordinate structure. *Linguistic Inquiry, 12,* 155–184.

Gazdar, G., Klein, E., Pullum, G. K., & Sag, I. A. (1985). *Generalized phrase structure grammar.* Oxford: Basil Blackwell.

Gibson, E. A. (1991). *A computational theory of human linguistic processing: Memory limitations and processing breakdown* (Tech. Rep. CMU-CMT-91–125). Pittsburgh, PA: Carnegie Mellon University, Center for Machine Translation.

Gibson, E. A. (1998). Linguistic complexity: Locality of syntactic dependencies. *Cognition, 68,* 1–76.

Gibson, E. A. (2000). The dependency locality theory: A distance-based theory of linguistic complexity. In Y. Miyashita, A. Mirantz, & W. O'Neil (Eds.), *Image, language, brain* (pp. 95–126). Cambridge, MA: MIT Press.

Gibson, E. A., & Thomas, J. (1997). *The complexity of nested structures in English: Evidence for the syntactic prediction locality theory of linguistic complexity.* Unpublished manuscript, Massachusetts Institute of Technology.

Gibson, E. A., & Thomas, J. (1999). Memory limitations and structural forgetting: The perception of complex ungrammatical sentences as grammatical. *Language and Cognitive Processes, 14,* 225–248.

Givón, T. (1978). Definiteness and referentiality. In J. H. Greenberg, C. A. Ferguson, & E. A. Moravcsik (Eds.), *Universals of human language: Vol. 4. Syntax* (pp. 291–330). Stanford, CA: Stanford University Press.

Gordon, P. C., Hendrick, R., & Johnson, M. (2001). Memory interference during language processing. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 27,* 1411–1423.

Gordon, P. C., Hendrick, R., & Johnson, M. (2004). Effects of noun phrase type on sentence complexity. *Journal of Memory and Language, 51,* 97–114.

Gordon, P. C., Hendrick, R., & Levine, W. H. (2002). Memory load interference in syntactic processing. *Psychological Science, 13,* 425–430.

Green, G. M. (1976). Main clause phenomena in subordinate clauses. *Language, 52,* 382–397.

Grodner, D., Gibson, E., & Tunstall, S. (2002). Syntactic complexity in ambiguity resolution. *Journal of Memory and Language, 46,* 267–295.

Grodner, D. J., & Gibson, E. A. (in press). Consequences of the serial nature of linguistic input for sentential complexity. *Cognitive Science.*

Hagoort, P. (2003). How the brain solves the binding problem for language: A neurocomputational model of syntactic processing. *NeuroImage, 20,* S18–S29.

Hakes, D. T., & Cairns, H. S. (1970). Sentence comprehension and relative pronouns. *Perception & Psychophysics, 8*(1), 5–8.

Hakes, D. T., Evans, J. S., & Brannon, L. L. (1976). Understanding sentences with relative clauses. *Memory and Cognition, 4,* 283–290.

Hakuta, K. (1981). Grammatical description versus configurational arrangement in language acquisition: The case of relative clauses in Japanese. *Cognition, 9,* 197–236.

Hirose, Y., & Inoue, A. (1998). Ambiguities in parsing complex sentences in Japanese. In D. Hillert (Ed.), *Syntax and semantics* (Vol. 31, pp. 71–93). New York: Academic.

Holmes, V. M., & O'Regan, J. K. (1981). Eye fixation patterns during the reading of relative clause sentences. *Journal of Verbal Learning and Verbal Behavior, 20,* 417–430.

Hooper, J. B., & Thompson, S. A. (1973). On the applicability of root transformations. *Linguistic Inquiry, 4,* 465–497.

Hudgins, J. C., & Cullinan, W. L. (1978). Effects of sentence structure on sentence elicited imitation responses. *Journal of Speech and Hearing Research, 21,* 809–819.

Inoue, A., & Fodor, J. D. (1995). Information-paced parsing of Japanese. In R. Mazuka & N. Nagai (Eds.), *Japanese sentence processing* (pp. 9–63). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Johnson-Laird, P. N. (1983). *Mental models*. Cambridge, MA: Harvard University Press.

Jurafsky, D. (1996). A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science, 20,* 137–194.

Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review, 99,* 122–149.

Kaan, E., & Vasic, N. (2004). Cross-serial dependencies in Dutch: Testing the influence of NP type on processing load. *Memory & Cognition*, *32,* 175–184.

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

King, J., & Just, M. A. (1991). Individual differences in syntactic processing: The role of working memory. *Journal of Memory and Language, 30,* 580–602.

Konieczny, L. (2000). Locality and parsing complexity. *Journal of Psycholinguistic Research, 29,* 627–645.

Konieczny, L., Hemforth, B., Scheepers, C., & Strube, G. (1997). The role of lexical heads in parsing: Evidence from German. *Language and Cognitive Processes, 12,* 307–348.

Korthals, C. (2001). Self embedded relative clauses in a corpus of German newspaper texts. In K. Striegnitz (Ed.), *Proceedings of the Sixth ESSLLI Student Session* (pp. 179–190). Helsinki, Finland: University of Helsinki.

Koster, J. (1978). Why subject sentences don't exist. In S. J. Keyser (Ed.), *Recent transformational studies in European languages* (pp. 53–64). Cambridge, MA: MIT Press.

Larkin, W., & Burns, D. (1977). Sentence comprehension and memory for embedded structure. *Memory and Cognition, 5,* 17–22.

Lewis, R. L. (1993). *An architecturally-based theory of human sentence comprehension* (Tech. Rep. CMU-CS-93–226). Retrieved March 18, 2005, from http://www-personal.umich.edu/~rickl/Documents/thesis.pdf

Lewis, R. L. (1996). Interference in short-term memory: The magical number two (or three) in sentence processing. *Journal of Psycholinguistic Research, 25,* 93–115.

Lewis, R. L. (1998a). Leaping off the garden path: Reanalysis and limited repair parsing. In J. Fodor & F. Ferreira (Eds.), *Reanalysis in sentence processing* (pp. 247–285). Boston: Kluwer Academic.

Lewis, R. L. (1998b, March). *Working memory in sentence processing: Retroactive and proactive interference in parsing.* Paper presented at The 11th Annual CUNY Conference on Human Sentence Processing, New Brunswick, NJ.

Lewis, R. L. (2000). Specifying architectures for language processing: Process, control, and memory in parsing and interpretation. In M. W. Crocker, M. Pickering, & C. Clifton Jr. (Eds.), *Architectures and mechanisms for language processing* (pp. 56–89). Cambridge, England: Cambridge University Press.

Lovett, M. (1998). Choice. In J. R. Anderson & C. Lebiere (Eds.), *The atomic components of thought* (pp. 255–296). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

MacDonald, M. C., & Christiansen, M. (2002). Reassessing working memory: Comment on Just and Carpenter (1992) and Waters and Caplan (1996). *Psychological Review, 109,* 35–54.

MacDonald, M. C., Pearlmutter, N. J., & Seidenberg, M. S. (1994). The lexical nature of syntactic ambiguity resolution. *Psychological Review, 101,* 676–703.

Marks, L. E. (1968). Scaling of grammaticalness of self-embedded English sentences. *Journal of Verbal Learning and Verbal Behavior, 7,* 965–967.

McElree, B. (1993). The locus of lexical preference effects in sentence comprehension: A time-course analysis. *Journal of Memory and Language, 32,* 536–571.

McElree, B. (1998). Attended and non-attended states in working memory: Accessing categorized structures. *Journal of Memory and Language, 38,* 225–252.

McElree, B., & Dosher, B. A. (1989). Serial position and set size in short-term memory: Time course of recognition. *Journal of Experimental Psychology: General, 118,* 346–373.

McElree, B., & Dosher, B. A. (1993). Serial retrieval processes in the recovery of order information. *Journal of Experimental Psychology: General, 122,* 291–315.

McElree, B., Foraker, S., & Dyer, L. (2003). Memory structures that subserve sentence comprehension. *Journal of Memory and Language, 48,* 67–91.

McKoon, G., & Ratcliff, R. (2003). Meaning through syntax: Language comprehension and the reduced relative clause construction. *Psychological Review, 110,* 490–525.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review, 104,* 3–65.

Miller, G. A. (1956). The magical number seven plus or minus two: Some limits on our capacity for processing information. *Psychological Review, 63,* 81–97.

Miller, G. A., & Chomsky, N. (1963). Finitary models of language users. In D. R. Luce, R. R. Bush, & E. Galanter (Eds.), *Handbook of mathematical psychology* (Vol. 2, pp. 419–492). New York: Wiley.

Miller, G. A., & Isard, S. (1964). Free recall of self-embedded English sentences. *Information and Control, 7,* 292–303.

Mitchell, D. C. (1994). Sentence parsing. In M. A. Gernsbacher (Ed.), *Handbook of psycholinguistics* (pp. 375–409). San Diego: Academic.

Newell, A. (1973). Production systems: Models of control structures. In W. G. Chase (Ed.), *Visual information processing* (pp. 463–526). New York: Academic Press.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.

Pollard, C., & Sag, I. (1994). *Head-driven phrase structure grammar.* Chicago: University of Chicago Press.

Schabes, Y., & Joshi, A. K. (1991). Parsing in lexicalized tree adjoining grammar. In M. Tomita (Ed.), *Current issues in parsing technologies* (pp. 25–47). Boston: Kluwer Academic.

Schlesinger, I. M. (1975). Why a sentence in which a sentence in which a sentence is embedded is embedded is difficult. *International Journal of Psycholinguistics,* 53–66.

Sheldon, A. (1974). The role of parallel function in the acquisition of relative clauses in English. *Journal of Verbal Learning and Verbal Behavior, 13,* 272–281.

Simon, H. A. (1974, February 8). How big is a chunk? *Science, 183,* 482–488.

Srinivas, B., & Joshi, A. K. (1999). Supertagging: An approach to almost parsing. *Computational Linguistics, 25,* 237–265.

Stabler, E. P. (1994). The finite connectivity of linguistic structure. In C. Clifton Jr., L. Frazier, & K. Rayner (Eds.), *Perspectives in sentence processing* (pp. 303–336). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc. (An earlier version of this paper appeared in UCLA Occasional Papers No.11.)

Steedman, M. (1996). *Surface structure and interpretation*. Cambridge, MA: MIT Press.

Steedman, M. (2000). *The Syntactic Process.* Cambridge, MA: MIT Press.

Stevenson, S. (1994). Competition and recency in a hybrid network model of syntactic disambiguation. *Journal of Psycholinguistic Research, 23,* 295–322.

Tabor, W., Juliano, C., & Tanenhaus, M. K. (1998). Parsing in a dynamical system: An attractor-based account of the interaction of lexical and structural constraints in sentence processing. *Language and Cognitive Processes, 12,* 211–272.

Tanenhaus, M. K., Spivey-Knowlton, M. J., Eberhard, K. M., & Sedivy, J. C. (1995). Integration of visual and linguistic information in spoken language comprehension. *Science, 268,* 1632–1634.

Trueswell, J. C., Tanenhaus, M. K., & Garnsey, S. M. (1994). Semantic influences in parsing: Use of thematic role information in syntactic ambiguity resolution. *Journal of Memory and Language, 33,* 285–318.

Ullman, M. T. (2004). Contributions of memory circuits to language: The declarative/procedural model. *Cognition, 92,* 231–270.

Van Dyke, J. A. (2002). *Retrieval effects in sentence parsing and interpretation*. Unpublished doctoral dissertation, University of Pittsburgh, Pittsburgh, PA.

Van Dyke, J. A., & Lewis, R. L. (2003). Distinguishing effects of structure and decay on attachment and repair: A cue-based parsing account of recovery from misanalyzed ambiguities. *Journal of Memory and Language, 49,* 285–316.

van Noord, G. (1997). An efficient implementation of the head-corner parser. *Computational Linguistics, 23,* 425–456.

Vannest, J., Newman, A., Polk, T. A., Lewis, R. L., Newport, E. L., & Bavelier, D. (2004). *fMRI evidence that some, but not all, derived words are decomposed during lexical access.* Manuscript submitted for publication.

Vannest, J., Polk, T. A., & Lewis, R. L. (2003, in revision). Dual-route processing of complex words: New fMRI evidence from derivational suffixation. *Cognitive, Affective, and Behavioral Neuroscience.*

Vasishth, S. (2003a). Quantifying processing difficulty in human sentence parsing: The role of decay, activation, and similarity-based interference. In *Proceedings of the European Cognitive Science Conference* (*Osnabrück*). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Vasishth, S. (2003b). *Working memory in sentence comprehension: Processing Hindi center embeddings*. New York: Garland.

Vasishth, S., Cramer, I., Scheepers, C., & Wagner, J. (2003). Does increasing distance facilitate processing? In *Proceedings of the 16th Annual CUNY Sentence Processing Conference.* MIT, Cambridge, MA (Available from: ftp://ftp.coli.uni-sb.de/pub/people/vasishth/ wip.html)

Vasishth, S., Scheepers, C., Uszkoreit, H., Wagner, J., & Kruijff, G. J. M. (2004, March). *Constraint defeasibility and concurrent constraint satisfaction in human sentence processing.* Paper presented at the CUNY Sentence Processing Conference, Cambridge, MA.

Vosse, T., & Kempen, G. (2000). Syntactic structure assembly in human parsing: A computational model based on competitive inhibition and a lexicalist grammar. *Cognition, 75,* 105–143.

Wang, M. D. (1970). The role of syntactic complexity as a determiner of comprehensibility. *Journal of Verbal Learning and Verbal Behavior, 9,* 398–404.

Wickelgren, W. A., Corbett, A. T., & Dosher, B. A. (1980). Priming and retrieval from short-term memory: A speed-accuracy trade-off analysis. *Journal of Verbal Learning and Verbal Behavior, 19,* 387–404.

Yngve, V. H. (1960). A model and an hypothesis for language structure. *Proceedings of the American Philosophical Society, 104,* 444–466.