A Model of Language Processing as Hierarchic Sequential Prediction

Marten van Schijndel
The Ohio State University
vanschm@ling.ohio-state.edu

Andy Exley
University of Minnesota
exley@cs.umn.edu

William Schuler
The Ohio State University
schuler@ling.ohio-state.edu

Abstract

Computational models of memory are often expressed as hierarchic sequence models, but the hierarchies in these models are typically fairly shallow, reflecting the tendency for memories of superordinate sequence states to become increasingly conflated. This article describes a broad-coverage probabilistic sentence processing model that uses a variant of a left-corner parsing strategy to flatten sentence processing operations in parsing into a similarly shallow hierarchy of learned sequences. The main result of this paper is that a model with these kinds of constraints can process broad coverage newspaper text with the same accuracy as a state-of-the-art parser not defined in terms of sequential working memory operations.

Introduction

Recent models of working memory (Howard and Kahana, 2002; Botvinick, 2007) are defined in terms of hierarchic sequential and temporal cueing operations. Observed events (for example, visible grasping and manipulation actions) are organized into hypothesized sequences of more general states (actions in a process of making coffee), encoded in a changing context (a set of continuous-valued neural units expressing *features* of states) representing a weighted set of active hypotheses pursued in parallel. Sequences of these states may themselves belong to higher-level sequences (steps in making breakfast, for example), forming a multi-level hierarchy. Sequential transitions between states in each level of this hierarchy may be directly learned from experience, then recalled rapidly and reliably by cueing successive states on features of preceding states (observations of pouring coffee into a mug are likely to be followed by adding milk, say). But when these learned sequences terminate, the process must recall its place in some immediately superordinate sequence (the current step in making breakfast). Unlike content-based sequential cueing, the transition from a terminating subordinate state to a state in some immediately superordinate sequence may not have been directly learned, so the superordinate state must instead be recalled based on the similarity of a set of *temporal features* associated with this state to a set of temporal features in the current context. These temporal features change as the current context changes. As a result, this temporal cueing becomes less reliable and takes more time to converge as the

similarity of these temporal features decreases. This provides a tidy explanation of scale-invariant long-term recency effects in serial recall experiments (Bjork and Whitten, 1974; Crowder, 1982).[1]

Can sentence processing be modeled using the same kind of sequential and temporal cueing operations popular in the computational memory community? Words, phrases, and sentences form hierarchies just like observed events, actions, and processes. But unlike phrase structure trees and discourse structures, the hierarchic sequence models described in the computational memory literature are typically fairly shallow, reflecting the tendency for memories of superordinate sequence states to become increasingly conflated as the temporal features of the current context diverge from their temporal features.

This article describes a broad-coverage probabilistic sentence processing model that uses a variant of a left-corner parsing strategy (Aho and Ullman, 1972) to flatten sentence processing operations into a similarly shallow hierarchy of learned sequences. These sequences are then mapped to explicit states in a hierarchic probabilistic sequence model. Unlike similar broad-coverage sequence model parsers of Crocker and Brants (2000) and Henderson (2004), this model exploits a property of left-corner parsing that ensures that subordinate sequences are initiated or terminated no more than once in each hypothesis after each observed word. This provides a natural constraint on temporal cueing operations, yielding a shallower hierarchy of sequence states than those required by Crocker and Brants (2000) and Henderson (2004).

The main result of this paper is that a model with these kinds of constraints can process broad coverage newspaper text with the same accuracy as a state-of-the-art parser not defined in terms of sequential working memory operations. As argued by Crocker and Brants (2000), broad coverage models like this are valuable because they allow experimental evaluation of interactions among factors across a variety of phenomena under uniform modeling assumptions, providing a more rigorous test of claims about general linguistic behavior, including unanticipated consequences of modeling decisions. This is expected to facilitate broad-coverage experimental evaluations in which memory-based measures (for example, measures of subordinate sequence termination as a proportion of the set of active hypotheses) can be combined with frequency-based measures (for example, surprisal or entropy reduction; Hale, 2001, 2006) on a fair footing.

The remainder of this article is organized as follows: The first section describes an incremental processing model based on sequential and temporal cueing operations, The next section adapts the model for use with probabilistic context-free grammars, A broad-coverage evaluation of this model follows, and the article concludes with a discussion of issues related to this model.

Model

The model described in this article represents time in discrete steps t, corresponding to discrete observations of words x_t. At each time step, the model maintains several hypotheses q_t which are probabilistically weighted and considered in parallel, as an explicit decomposition of the high-dimensional hidden context of a recurrent neural network like that of Howard and Kahana (2002) or Botvinick (2007). Each hypothesis defines a hierarchy of sequence states q_t^d, ordered by depth d from superordinate ($d = 1$) to subordinate ($d > 1$). Each sequence state q_t^d defines a maximal *connected component* of predicted phrase structure a_t^d/b_t^d, consisting of an *active sign* of category a_t^d lacking an *awaited sign* of category b_t^d yet to come. Any connected sequence of signs descending over time in a predicted phrase structure tree (e.g. S, VP, and NP in Figure 1) can form a single connected component state (e.g. S/NP).

Over time, the model predicts syntactically structured signs, compares them against observed words, generalizes them into new connected component states, then merges them with

[1] That is, subjects show a preference for recalling recent items in list recall studies even when these items are separated by distractor tasks (e.g. performing arithmetic), contra predictions of working memory models that posit short-term buffers to explain recency effects.

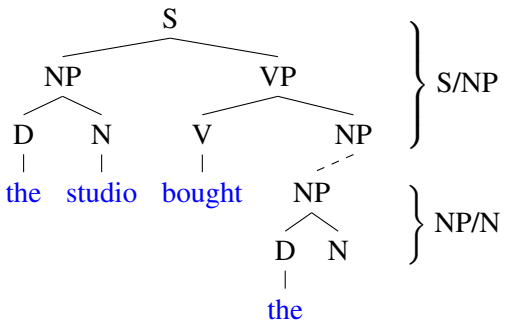

Figure 1. Two disjoint connected components of a phrase structure tree for the sentence *the studio bought the publisher's rights*, shown immediately prior to the word *publisher*.

subordinate and superordinate connected component states as the syntactic relations that connect them in the phrase structure tree are predicted. Any sequence of hierarchically-organized connected component states generated by this model corresponds to a traversal of a predicted phrase structure tree. For example, Figure 2 shows a hierarchic state sequence corresponding to a traversal of a predicted phrase structure tree for the sentence *The studio bought the publisher's rights*.

Note that transitions within a single hierarchy level may predict phrase structure relations upward along sequences of initial children (from the NP *the publisher* to the D *the publisher's* between time steps 5 and 6, for example), and downward along sequences of final children (from the VP *bought the publisher's rights* to the NP *the publisher's rights* between time steps 3 and 4). Although this predicted phrase structure may have an unbounded number of recursive initial or final children (for example, sequences of possessives extending the initial portion of a noun phrase or sequences of adjectives extending the final portion of a noun phrase, as shown in Figure 3), it requires only a bounded number of connected component states at any given time step. This flattening of the phrase structure into potentially cyclic sequences of connected component states is similar to the programming strategy of replacing head recursion and tail recursion with loops (where program instructions correspond to states, recursive function calls in the call stack correspond to subordinate states in the state hierarchy, and loops correspond to cyclic transitions over states like S/N and D/G). This is also similar to the left-corner parsing strategy commonly used in sentence processing models (Johnson-Laird, 1983; Abney and Johnson, 1991; Gibson, 1991; Henderson, 2004; Lewis and Vasishth, 2005), except that connected components in the state hierarchy defined here are paired into active signs with awaited signs somewhere on their final (right) edge, rather than as awaited signs with active signs somewhere on their initial (left) edge.[2]

Sequence Modeling with Connected Components

The general hierarchic sequence model described in this article is defined in terms of a set of syntactic relations — in particular, a set of context-free rules of the form $a \rightarrow a'\ b'$ (meaning sign a is composed of an initial child sign a' followed by final child sign b'), or $a \rightarrow x$ (meaning sign a is associated directly with an observation of word x). In addition to this set of syntactic relations, this model also assumes an ability to predict initial sub-signs a' of larger signs b, denoted $b \xrightarrow{+} a' \ldots$.

A simple nondeterministic process for incrementally predicting phase structures using a sequence of connected component states can be defined as a deductive system, given an input sequence consisting of an top-level connected component state $\top/\top$, corresponding to an existing discourse context, followed by a sequence of observed words $x_1, \ldots, x_n$, processed in time order.[3]

[2] In previous work, we therefore refer to this as *right-corner parsing* (Schuler et al., 2010).

[3] A deductive system consists of inferences or productions of the form: $\frac{P}{Q}R$, meaning premise P entails conclu-

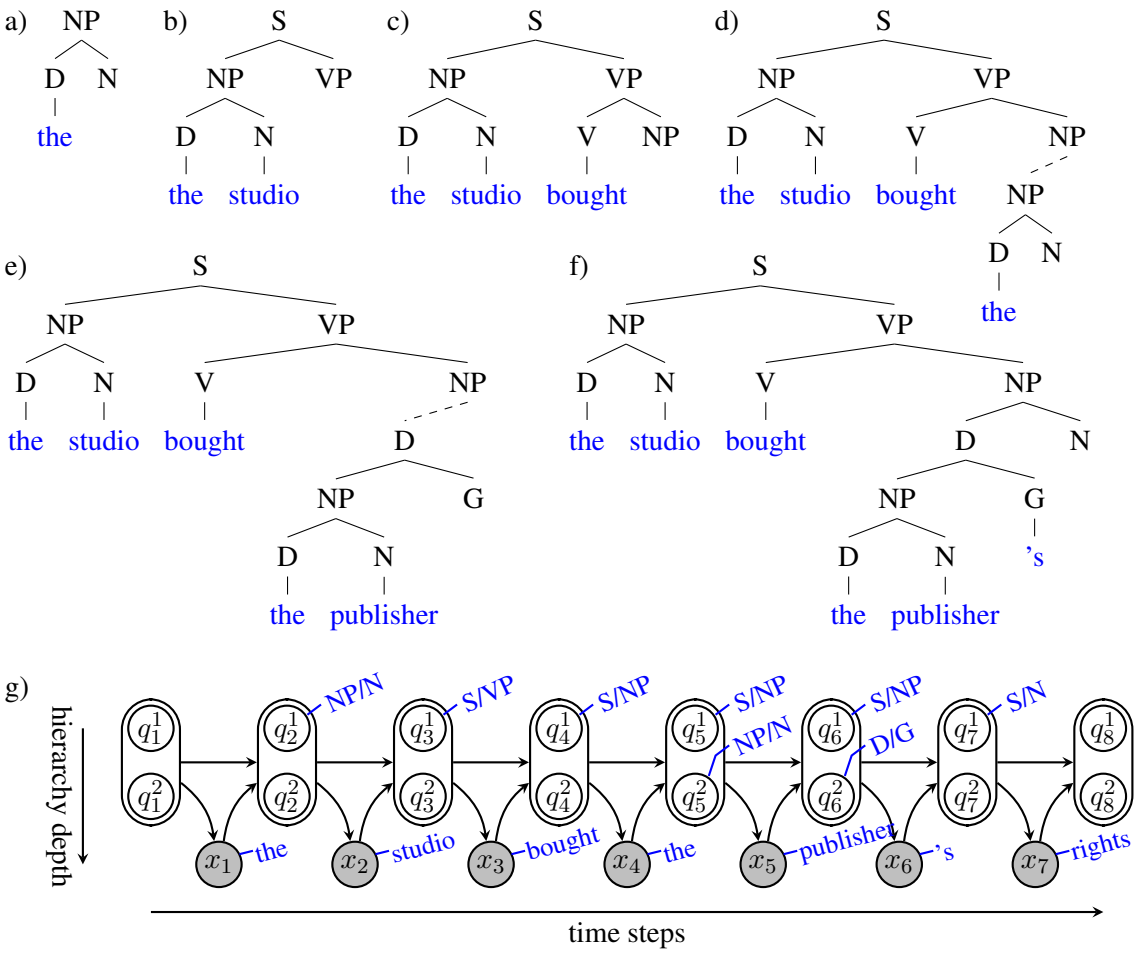

Figure 2. Incrementally constructed representations of the syntactic structure of the sentence *The studio bought the publisher's rights* (a–f), and the associated sequence of random variable values in a hierarchic sequential prediction model (g). Open circles represent hidden variables, shaded circles represent observed variables (x_t), and directed edges represent conditional dependencies. 'Pea-pod' ovals summarize dependencies over subsumed variables. Selected random variables are also annotated with example values, shown diagonally.

As each x_t is encountered, it is connected to the existing components or it introduces a new disjoint component using productions that treat each word as the first observation of a newly initiated connected component state, or as the last observation of a terminated connected component state, or as neither, or as both.

First, if an observation x_t can attach as the awaited sign of the most recent (most subordinate) connected component a/b, it is hypothesized to do so, turning this incomplete sign into a complete sign a (F–, below); or if the observation can serve as a lower descendant of this awaited sign, it is hypothesized to form the first complete sign a' in a newly initiated connected component (F+):

$$\frac{a/b \quad x_t}{a}\, b \rightarrow x_t \tag{F–}$$

$$\frac{a/b \quad x_t}{a/b \quad a'}\, b \xrightarrow{+} a' \ldots ; \quad a' \rightarrow x_t \tag{F+}$$

sion Q according to rule R.

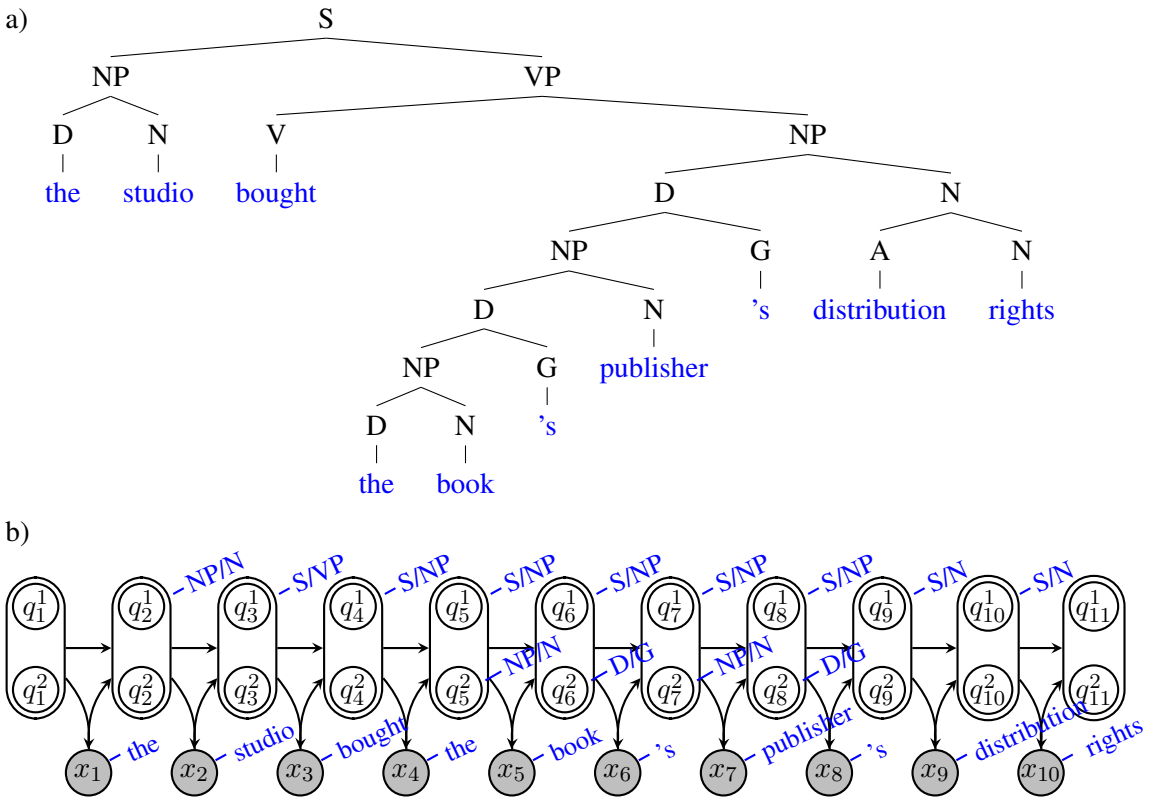

Figure 3. Phrase structure tree for the sentence *The studio bought the book's publisher's distribution rights* (a), with repeated initial and final signs for *book's* and *distribution*; and the associated sequence of random variable values in a hierarchic sequential prediction model (b), with corresponding repeated states D/G and S/N at time steps 8 and 10.

Then, if either of these complete signs (a or a' above, matched to a'' below) can attach as an initial child of the awaited sign of the immediately superordinate connected component state a/b, it is hypothesized to do so and terminate the subordinate connected component state, with x_t as the last observation of the terminated connected component (L+); or if the observation can serve as a lower descendant of this awaited sign, it is hypothesized to remain disjoint and form its own connected component (L–):

$$\frac{a/b \quad a''}{a/b''}\, b \rightarrow a''\ b'' \tag{L+}$$

$$\frac{a/b \quad a''}{a/b \quad a'/b''}\, b \xrightarrow{+} a' \ldots ;\ \ a' \rightarrow a''\ b'' \tag{L–}$$

These initiation (F) and termination (L) productions are similar to the push and pop operations respectively of a nondeterministic pushdown automaton or the shift and reduce operations of a shift-reduce parser (taking all non-observation consequents in this model to be store items), except that the model observes a memory-based processing constraint that it may use no more than one initiation (F) and one termination (L) production per time step. When applied to parsing, this model is therefore more constrained than the PDA-based models of Crocker and Brants (2000) and Henderson (2004), which allow unlimited numbers of initiations and terminations between each pair of observations. An example derivation of the sentence *The studio bought the publisher's rights,* using these productions is shown in Figure 4.

Note that the derivation shown in Figure 4 contains only two instances of a connected component more than one word long being composed with a disjoint superordinate connected compo-

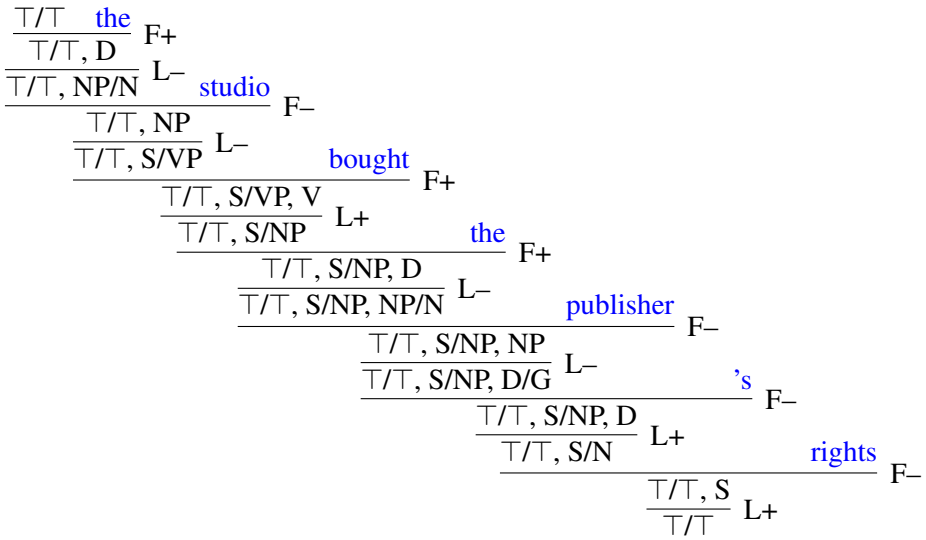

Figure 4. Derivation of the sentence *The studio bought the publisher's rights,* using F+, F–, L+, and L– productions.

nent, requiring the superordinate connected component to be recalled by temporal cueing. (This occurs after the genitive marker *'s*, merging *the publisher's* with *the studio bought*, and after the word *rights*, merging the sentence with the discourse context ⊤/⊤.) In general this only results from a non-initiation (F–) followed by a termination (L+). All other combinations of productions (non-termination, or initiation with immediate termination) can be implemented by sequential cueing on the content of the most subordinate state. This gives the system the capability to directly measure temporal cueing operations during parsing. When extended to a probabilistic model, this measure can be expressed as a proportion of a set of weighted hypotheses, and potentially combined with other probabilistic measures like surprisal and entropy reduction (Hale, 2001, 2006).

Also note that if there is no bound on the number of disjoint connected component states that can be hypothesized in a hierarchy, this system will be able generate all and only those syntactic structures allowed by the context-free rules it is defined over. This can be shown by observing that (i) every binary phrase structure tree over t observations must also contain t branches from a parent sign to a pair of children,[4] and (ii) for each observation x_t in a complete phrase structure there is a unique largest sign which is co-final with x_t. Inspection of the F and L operations shows that F– and F+ isolate this largest co-final sign (as a in the consequent of F– or a' in the consequent of F+), and L+ and L– connect this sign (as a'') to a parent and a sibling in some phrase structure (to b and b'' in L+ and to a' and b'' in L–). Since each such connection is unique, and since there are only t such connections in any complete phrase structure, the system must be able to predict any complete phrase structure for any sequence of t observations.

A Probabilistic Hierarchic Sequence Model

This process can be extended to calculate probabilities for phrase structure trees by introducing probability distributions ϕ and λ over the binary F and L decisions defined above, with constraints taken directly from these productions. F decisions (about whether to initiate a new subordinate sequence) are constrained such that:

$$\mathsf{P}_\phi(\text{`–'} \mid b\ x) \neq 0 \quad \text{only if} \quad b \rightarrow x \tag{1a}$$

$$\mathsf{P}_\phi(\text{`+'} \mid b\ x) \neq 0 \quad \text{only if} \quad b \xrightarrow{+} a' \ldots\ ;\ \ a' \rightarrow x \tag{1b}$$

[4] This includes a connection from the top-level sign of this structure to some discourse structure ⊤.

L decisions (about whether to terminate a subordinate sequence) are constrained such that:

$$\mathsf{P}_\lambda(\text{`+'} \mid b\ a'') \neq 0 \quad \text{only if} \quad b \rightarrow a''\ b'' \tag{2a}$$

$$\mathsf{P}_\lambda(\text{`–'} \mid b\ a'') \neq 0 \quad \text{only if} \quad b \overset{+}{\rightarrow} a' \ldots ;\ \ a' \rightarrow a''\ b'' \tag{2b}$$

Constraints for probability distributions α and β over the active and awaited signs a and b in hypothesized connected component states are also derived from F and L productions:

$$\mathsf{P}_\alpha(a' \mid b\ a'') \neq 0 \quad \text{only if} \quad b \overset{+}{\rightarrow} a' \ldots ;\ \ a' \rightarrow a''\ b'' \tag{3}$$

$$\mathsf{P}_\beta(b'' \mid a'\ a'') \neq 0 \quad \text{only if} \quad a' \rightarrow a''\ b'' \tag{4}$$

These constraints are more precisely defined in the next section.

Since F productions take observations x as input and produce complete signs a as output, and L productions take complete signs a as input and produce connected component states a/b as output, the process may only iterate by applying exactly one F production and one L production at each time step. Since F and L productions each have two ('+' and '–') options, a complete hierarchic transition model σ can be defined using only four cases: one for each combination of F and L productions. These cases are represented as addends in the definition below. Each addend is a product of factors for: (i) hypothesizing a combination of an initiation and a termination of a subordinate state sequence (using ϕ and λ probabilities), (ii) hypothesizing an active and awaited sign for the most subordinate connected component state in the resulting hierarchy (using α and β probabilities), and (iii) deterministically carrying forward empty or unmodified states from the previous time step. All models depend on the depth d of the most subordinate connected component state at the previous time step, and (in the case of the β model) on whether the first parameter is an active or awaited sign ('A' or 'B', respectively). In this definition, D is an arbitrary bound on the size of the state hierarchy (set to 4 in the evaluation described below), and $[\![...]\!]$ is a deterministic indicator function, evaluating to 1 if '...' is true, and 0 otherwise, used to represent a deterministic distribution.

The transition model is factored into three stages, below. First, the probability is split across the two possible outcomes for the F decision — whether to introduce a new subordinate sequence or not — based on the most subordinate connected component state q^d_{t-1} in the state hierarchy:

$$\begin{aligned}\mathsf{P}_\sigma(q_t^{1..D} \mid q_{t-1}^{1..D}\ x_{t-1}) \overset{\text{def}}{=}\ & \mathsf{P}_{\phi_d}(\text{`–'} \mid b^d_{t-1}\ x_{t-1}) \cdot \mathsf{P}_{\sigma'_d}(q_t^{1..D} \mid q_{t-1}^{1..D}\ a^d_{t-1}) \\ & + \mathsf{P}_{\phi_d}(\text{`+'} \mid b^d_{t-1}\ x_{t-1}) \cdot \mathsf{P}_{\sigma'_{d+1}}(q_t^{1..D} \mid q_{t-1}^{1..D}\ x_{t-1});\ \ d \overset{\text{def}}{=} \max\{d' \mid q^{d'}_{t-1} \neq \text{`–'}\}\end{aligned} \tag{5a}$$

For each F outcome, the transition model then splits the remaining probability across the two possible outcomes for the L decision — whether to terminate the current most subordinate sequence or not — traversing the predicted tree downward from b^{d-1}_{t-1} if so (using rule $b^{d-1}_{t-1} \rightarrow a''\ b^{d-1}_t$), and traversing the predicted tree upward from a'' if not (using rule $a^d_t \rightarrow a''\ b^d_t$):

$$\begin{aligned}\mathsf{P}_{\sigma'_d}(q_t^{1..D} \mid q_{t-1}^{1..D}\ a'') \overset{\text{def}}{=}\ & \mathsf{P}_{\lambda_d}(\text{`+'} \mid b^{d-1}_{t-1}\ a'') \cdot [\![a^{d-1}_t = a^{d-1}_{t-1}]\!] \cdot \mathsf{P}_{\beta_{\mathrm{B},d-1}}(b^{d-1}_t \mid b^{d-1}_{t-1}\ a'') \cdot \mathsf{P}_{\sigma''_{d-1}}(q_t^{1..D} \mid q_{t-1}^{1..D}) \\ & + \mathsf{P}_{\lambda_d}(\text{`–'} \mid b^{d-1}_{t-1}\ a'') \cdot \mathsf{P}_{\alpha_d}(a^d_t \mid b^{d-1}_{t-1}\ a'') \cdot \mathsf{P}_{\beta_{\mathrm{A},d}}(b^d_t \mid a^d_t\ a'') \cdot \mathsf{P}_{\sigma''_d}(q_t^{1..D} \mid q_{t-1}^{1..D})\end{aligned} \tag{5b}$$

For each combination of F and L outcomes, the transition model then ensures the rest of the hierarchy $q_t^{1..D}$ is copied over from the previous time step, or replaced with null values below the most subordinate connected component state in the hierarchy:

$$\mathsf{P}_{\sigma''_d}(q_t^{1..D} \mid q_{t-1}^{1..D}) \overset{\text{def}}{=} [\![q_t^{1..d-1} = q_{t-1}^{1..d-1}]\!] \cdot [\![q_t^{d+1..D} = \text{`–'}]\!] \tag{5c}$$

Note that the active sign of a superordinate state is deterministically carried forward whenever x_t is the last observation in a subordinate state sequence (when λ is positive). This is because the active sign of a superordinate state does not change when a subordinate state is terminated. These transition probabilities σ are then combined with observation probabilities ξ to define a most likely sequence of connected component hierarchies $\hat{q}_{1..T}^{1..D}$:

$$\hat{q}_{1..T}^{1..D} \stackrel{\text{def}}{=} \underset{q_{1..T}^{1..D}}{\operatorname{argmax}} \prod_{t=1}^{T} \mathsf{P}_\sigma(q_t^{1..D} \mid q_{t-1}^{1..D}\, x_{t-1}) \cdot \mathsf{P}_\xi(x_t \mid b_t^d); \quad d \stackrel{\text{def}}{=} \max\{d' \mid q_{t-1}^{d'} \neq \text{`–'}\} \tag{6}$$

This model predicts phrase structure while restricting access to superordinate states as a memory-based recency constraint. Since the model is implemented as a simple sum of products, it is essentially equivalent to a localist representation in a recurrent neural network, albeit one with a hidden context unit for every combination of disjoint connected components, represented in an explicit state hierarchy. In this respect, the hierarchic sequence model described in this article is similar to the hierarchic connectionist memory model of Botvinick (2007). However, in order to maintain a close connection with linguistic notions of phrase structure, the model is not trained using unsupervised learning techniques traditionally applied to connectionist models.

Application to Probabilistic Context Free Grammars

The constraints described in the previous section can be satisfied in a variety of ways. The model evaluated in this article is directly defined over a Probabilistic Context Free Grammar (PCFG), trained using latent variable induction (Petrov et al., 2006). PCFGs are widely used in parsing because they provide a simple branching stochastic process (Collins, 1997), because well-studied algorithms exist for inferring or refining PCFGs from data (Petrov et al., 2006), and because PCFGs have been shown to be useful as a basis for information-theoretic accounts of garden path effects and reading time delays (Jurafsky, 1996; Hale, 2001, 2003, 2006; Levy and Jaeger, 2007).

Side- and Depth-specific Rules

The general hierarchic sequence model described in the previous section can be defined for a given PCFG by first deriving side- and depth-specific rule probabilities and expected counts of initial sub-signs derived from rule probabilities in Chomsky Normal Form (CNF).[5] This derivation is expressed using context-free grammar notation:

- $\mathsf{P}_\gamma(a \rightarrow x)$ denotes the probability of a sign of category a expanding into an observation,
- $\mathsf{P}_{\gamma_{s,d}}(a' \rightarrow a''\ b'')$ denotes the probability that a sign of category a' at hierarchy depth d occurring on (initial or final) side s of its parent will expand into an initial sign of category a'' followed by a final sign of category b'',
- $\mathsf{E}_{\gamma_d^*}(b \stackrel{+}{\rightarrow} a'\ ...)$ denotes the expected number of times a sign of category a' at hierarchy depth d will occur as an initial sub-sign of another sign of category b, and
- $\mathsf{E}_{\gamma_d^*}(b \stackrel{*}{\rightarrow} a'\ ...)$ denotes the expected number of times a sign of category a' at hierarchy depth d will occur either as equivalent to or as an initial sub-sign of another sign of category b.

In lieu of a confusability model, the model instead imposes hard constraints on the number of distinct syntactically connected components allowed in each hypothesis. This has the effect of bounding the number of center embeddings allowed in any partial syntactic tree (in particular, initial children of final children in any CNF derivation). This is done by first computing a side- and depth-specific PCFG 'fit' model $\delta_{s,d}^{(i)}$, defining the probability that a subtree below a sign of category a, occurring on its parent's initial or final side $s \in \{A, B\}$, will fit within a bounded

[5] PCFGs not in CNF can be compiled into CNF by binarizing with unique symbols.

depth d of disjoint connected components. This fit model is computed according to the following recursive definition, where i is the recursive iteration:

$$\mathsf{P}_{\delta^{(0)}_{s,d}}(1 \mid a) \stackrel{\text{def}}{=} 0 \tag{7a}$$

$$\mathsf{P}_{\delta^{(i)}_{A,d}}(1 \mid a) \stackrel{\text{def}}{=} \sum_x \mathsf{P}_\gamma(a \to x) + \sum_{a',b'} \mathsf{P}_\gamma(a \to a'\ b') \cdot \mathsf{P}_{\delta^{(i-1)}_{A,d}}(1 \mid a') \cdot \mathsf{P}_{\delta^{(i-1)}_{B,d}}(1 \mid b') \tag{7b}$$

$$\mathsf{P}_{\delta^{(i)}_{B,d}}(1 \mid a) \stackrel{\text{def}}{=} \sum_x \mathsf{P}_\gamma(a \to x) + \sum_{a',b'} \mathsf{P}_\gamma(a \to a'\ b') \cdot \mathsf{P}_{\delta^{(i-1)}_{A,d+1}}(1 \mid a') \cdot \mathsf{P}_{\delta^{(i-1)}_{B,d}}(1 \mid b') \tag{7c}$$

Note that the only difference between the initial-sign ($\delta^{(i)}_{A,d}$) and final-sign ($\delta^{(i)}_{B,d}$) cases above is simply that the depth is incremented for initial children of final children. These initial-final zig-zags form the breaks between connected components in the hierarchy. In practice the recursive product is estimated to some constant I using value iteration (Bellman, 1957).

Now a side- and depth-specific PCFG model $\gamma_{s,d}$ can be defined by renormalizing over the probability mass isolated in $\delta^{(I)}_{s,d}$:

$$\mathsf{P}_{\gamma_{A,d}}(a \to a'\ b') \stackrel{\text{def}}{=} \frac{\mathsf{P}_\gamma(a \to a'\ b') \cdot \mathsf{P}_{\delta^{(I)}_{A,d}}(1 \mid a') \cdot \mathsf{P}_{\delta^{(I)}_{B,d}}(1 \mid b')}{\mathsf{P}_{\delta^{(I)}_{A,d}}(1 \mid a)} \tag{8a}$$

$$\mathsf{P}_{\gamma_{B,d}}(a \to a'\ b') \stackrel{\text{def}}{=} \frac{\mathsf{P}_\gamma(a \to a'\ b') \cdot \mathsf{P}_{\delta^{(I)}_{A,d+1}}(1 \mid a') \cdot \mathsf{P}_{\delta^{(I)}_{B,d}}(1 \mid b')}{\mathsf{P}_{\delta^{(I)}_{B,d}}(1 \mid a)} \tag{8b}$$

This renormalizing over $\delta^{(I)}_{s,d}$ ensures no probability mass is lost when the depth of the model is bounded. Again, the only difference between the initial- and final-sign cases is that the depth is incremented for initial children of final children.

Initial Sub-sign Expected Counts

The model also needs initial sub-sign expected counts, which are based on the expected number of times a constituent of category a'' occurs at the beginning of a constituent of category b after any number of expansions. This is also estimated recursively with j as the recursive iteration:

$$\mathsf{E}_{\gamma^*_d}(b \stackrel{1}{\to} a'\ ...) \stackrel{\text{def}}{=} \sum_{b'} \mathsf{P}_{\gamma_{B,d}}(b \to a'\ b') \tag{9a}$$

$$\mathsf{E}_{\gamma^*_d}(b \stackrel{j}{\to} a''\ ...) \stackrel{\text{def}}{=} \sum_{a',b''} \mathsf{E}_{\gamma^*_d}(b \stackrel{j-1}{\to} a'\ ...) \cdot \mathsf{P}_{\gamma_{A,d}}(a' \to a''\ b'') \tag{9b}$$

$$\mathsf{E}_{\gamma^*_d}(b \stackrel{+}{\to} a''\ ...) \stackrel{\text{def}}{=} \sum_{j=1}^{\infty} \mathsf{E}_{\gamma^*_d}(b \stackrel{j}{\to} a''\ ...) \tag{9c}$$

$$\mathsf{E}_{\gamma^*_d}(b \stackrel{*}{\to} a''\ ...) \stackrel{\text{def}}{=} [\![b{=}a'']\!] + \mathsf{E}_{\gamma^*_d}(b \stackrel{+}{\to} a''\ ...) \tag{9d}$$

Here again, the recursive products and infinite sum are estimated to some constant J using value iteration (Bellman, 1957).

Transition Operations for Language Comprehension

The model probabilities are then just straightforward probabilistic implementations of the constraints specified in Equations 1a–4 expressed in terms of the bounded PCFG probabilities and initial sub-sign expected counts defined above, normalized appropriately:

1. The initiation model ϕ probabilities are calculated from the expected counts of a sign of syntactic category a' occurring as an initial sub-sign of a larger sign of category b multiplied by the probability of that category generating an observation x:

$$\mathsf{P}_{\phi_d}(\text{`}-\text{'} \mid b\, a') \stackrel{\text{def}}{=} \frac{[\![b = a']\!] \cdot \sum_x \mathsf{P}_\gamma(a' \to x)}{\mathsf{E}_{\gamma_d^*}(b \xrightarrow{*} a' \ldots) \cdot \sum_x \mathsf{P}_\gamma(a' \to x)} \tag{10a}$$

$$\mathsf{P}_{\phi_d}(\text{`}+\text{'} \mid b\, a') \stackrel{\text{def}}{=} \frac{\mathsf{E}_{\gamma_d^*}(b \xrightarrow{+} a' \ldots) \cdot \sum_x \mathsf{P}_\gamma(a' \to x)}{\mathsf{E}_{\gamma_d^*}(b \xrightarrow{*} a' \ldots) \cdot \sum_x \mathsf{P}_\gamma(a' \to x)} \tag{10b}$$

2. The termination model λ probabilities are calculated from the expected counts of a sign of syntactic category a' occurring as an initial sub-sign of a larger sign of category b multiplied by the probability of that sign having an initial child of category a'':

$$\mathsf{P}_{\lambda_d}(\text{`}+\text{'} \mid b\, a'') \stackrel{\text{def}}{=} \frac{\sum_{a',b''} [\![b = a']\!] \cdot \mathsf{P}_{\gamma_{B,d}}(a' \to a''\ b'')}{\sum_{a',b''} \mathsf{E}_{\gamma_d^*}(b \xrightarrow{*} a' \ldots) \cdot \mathsf{P}_{\gamma_{A,d}}(a' \to a''\ b'')} \tag{11a}$$

$$\mathsf{P}_{\lambda_d}(\text{`}-\text{'} \mid b\, a'') \stackrel{\text{def}}{=} \frac{\sum_{a',b''} \mathsf{E}_{\gamma_d^*}(b \xrightarrow{+} a' \ldots) \cdot \mathsf{P}_{\gamma_{A,d}}(a' \to a''\ b'')}{\sum_{a',b''} \mathsf{E}_{\gamma_d^*}(b \xrightarrow{*} a' \ldots) \cdot \mathsf{P}_{\gamma_{A,d}}(a' \to a''\ b'')} \tag{11b}$$

3. The active sign model α probabilities are calculated from the expected counts of a sign of syntactic category a' occurring as an initial sub-sign of a larger sign of category b multiplied by the probability of that sign having an initial child of category a'':

$$\mathsf{P}_{\alpha_d}(a' \mid b\, a'') \stackrel{\text{def}}{=} \frac{\sum_{b''} \mathsf{E}_{\gamma_d^*}(b \xrightarrow{+} a' \ldots) \cdot \mathsf{P}_{\gamma_{A,d}}(a' \to a''\ b'')}{\sum_{a',b''} \mathsf{E}_{\gamma_d^*}(b \xrightarrow{+} a' \ldots) \cdot \mathsf{P}_{\gamma_{A,d}}(a' \to a''\ b'')} \tag{12}$$

4. The awaited sign model β probabilities are simply the probability that a sign of category a has a final child of category b' given that it has an initial child of category a':

$$\mathsf{P}_{\beta_{s,d}}(b' \mid a\, a') \stackrel{\text{def}}{=} \frac{\mathsf{P}_{\gamma_{s,d}}(a \to a'\ b')}{\sum_{b'} \mathsf{P}_{\gamma_{s,d}}(a \to a'\ b')} \tag{13}$$

Transition Model for Language Comprehension

These individual model probabilities are combined into a single hierarchic transition probability σ, as described in Equation 5a of the previous section. These transition probabilities σ are then combined with preterminal and terminal probabilities π and ξ, described below.

In the previous section, observations were generated directly from awaited signs. In practice, it is more efficient to make the assumption that certain syntactic categories are preterminals, which generate a single lexical observation as a child. Such an assumption is made primarily for efficiency since only a subset of syntactic categories must then be considered for prediction, which reduces the complexity of the ϕ, α, and β models that depend on preterminal signs.

Formally, the preterminal probabilities π define the normalized probabilities of generating a preterminal p as an initial sub-sign of a larger sign of category b:

$$\mathsf{P}_{\pi_d}(p \mid b) \stackrel{\text{def}}{=} \mathsf{E}_{\gamma_d^*}(b \xrightarrow{*} p \ldots) \cdot \sum_x \mathsf{P}_\gamma(p \to x) \tag{14}$$

Terminal probabilities ξ are then defined as the normalized probabilities of generating an observation x from a preterminal p:

$$\mathsf{P}_\xi(x \mid p) \stackrel{\text{def}}{=} \frac{\mathsf{P}_\gamma(p \to x)}{\sum_x \mathsf{P}_\gamma(p \to x)} \tag{15}$$

These transition probabilities σ, preterminal probabilities π, and terminal probabilities ξ are then combined to define a most likely sequence $\hat{q}_{1..T}^{1..D}$:

$$\hat{q}_{1..T}^{1..D} \stackrel{\text{def}}{=} \underset{q_{1..T}^{1..D}}{\operatorname{argmax}} \prod_{t=1}^{T} \mathsf{P}_{\sigma}(q_t^{1..D} \mid q_{t-1}^{1..D}\, p_{t-1}) \cdot \mathsf{P}_{\pi_{d'}}(p_t \mid b_t^d) \cdot \mathsf{P}_{\xi}(x_t \mid p_t); \;\; d \stackrel{\text{def}}{=} \max\{d' \mid q_{t-1}^{d'} \neq \text{`-'}\} \tag{16}$$

Evaluation

In order to determine whether the flattened hierarchy generated by the single initiation and single termination constraints described above could predict phrase structure trees as accurately as a model without such constraints, the hierarchic sequence model described in this article was evaluated on a standard parsing task (Collins, 1997). This task reproduces labeled bracketings on a standard test set of newspaper text from the Wall Street Journal section of the Penn Treebank (Marcus et al., 1993), which can be directly compared against published results of other models. Comparable results to these systems would indicate that the memory-based constraints of the hierarchic sequence model don't harm language processing performance.

Training

The model evaluated in this article uses the Petrov et al. (2006) split-merge-smooth algorithm to extract a latent variable PCFG from the Penn Treebank (Marcus et al., 1993). The corpus delimits syntactic constituents with parentheses and category labels. The split-merge-smooth algorithm attempts to find latent subcategorizations (splits) of each category label which conform to distinct distributions in a set of training data relative to the surrounding category labels. For example, the class of present tense ditransitive verbs (such as *gives*) may be discovered to have a different distribution than the class of present tense transitive verbs (such as *owns*), though both are typically labelled 'VBZ' (present tense verb) in the Treebank. Another iteration of the splitting algorithm may then find that certain categories appear more often as first arguments of ditransitive verbs (now that they have been uniquely identified) than as arguments of transitive verbs. Assigning each such distribution a unique subcategory label helps encode mild contextual information into each label. To avoid overfitting to the training data, splits which are not sufficiently statistically informative are then merged back into a larger category. The PCFG relations used in the previous section are then calculated over these refined grammar categories.

Results

The accuracy of the hierarchic sequence model as a parser was compared to that of the Petrov and Klein (2007) and Roark (2001) parsers using varying beam widths (numbers of competing hypotheses).[6] The Petrov and Klein (2007) parser is a state-of-the-art chart parser based on the same latent variable PCFG (Petrov et al., 2006) used to define the hierarchic sequence model evaluated in this article. As a chart parser, it does not calculate prefix probabilities like the model described in this article and therefore cannot be used to calculate complexity measures like surprisal or entropy reduction. The Roark (2001) parser is an incremental parser widely used in cognitive modeling evaluations, and can be used to calculate prefix probabilities necessary for calculating complexity measures like surprisal, but it is not as accurate as that of Petrov and Klein. Neither the Petrov and Klein nor Roark parsers are defined in terms analogous to sequential or temporal cued recall operations.

[6] The Petrov and Klein (2007) parser was run using the Viterbi decoding option on the latent variable grammar.

System	Precision	Recall	F-score
Roark 2001 (CNF)	86.6	86.5	86.5
Current Model (CNF, beam width 500)	86.6	87.3	87.0
Current Model (CNF, beam width 2000)	87.8	87.8	87.8
Current Model (CNF, beam width 5000)	87.8	87.8	87.8
Petrov Klein (CNF)	88.1	87.8	88.0
Petrov Klein (not CNF)	88.3	88.6	88.5

Table 1

Accuracy comparison with state-of-the-art syntactic parsers. Numbers in parentheses are the number of parallel activated hypotheses. The left-corner parser used here restricts trees to Chomsky Normal Form (CNF), in which trees are binary branching at all nonterminals except preterminals. This makes the model less able to reproduce unary branches in the Penn Treebank.

The evaluated hierarchic sequence model restricts the number of disjoint connected components in any hypothesis to at most four. This limit was empirically determined to be sufficient to achieve greater than 99.9% coverage on the Wall Street Journal Corpus (Schuler et al., 2010).

All parsers were trained on Sections 02-21 of the Wall Street Journal portion of the Penn Treebank (Marcus et al., 1993) and tested on Section 23. No tuning was done as part of the conversion to a sequence model. With the exception of the Roark (2001) parser,[7] all parsers used 5 iterations of the Petrov et al. (2006) split-merge-smooth algorithm.[8] These results are shown in Table 1. Note that the Petrov and Klein (2007) parser allows unary branching within the phrase structure, which is not directly supported by the set of production rules described in the model section of this article.To obtain a fair comparison, it was also run with strict binarization (restricting the grammar to Chomsky Normal Form). The hierarchic sequence model described in this article achieves comparable accuracy to the Petrov and Klein (2007) parser assuming a strictly binary-branching phrase structure, and superior accuracy to the Roark (2001) parser.

Conclusion and Discussion

The results of this evaluation indicate that the flattened hierarchic sequence model described in this article can obtain similar accuracy to state-of-the-art methods. This shows that the seemingly austere constraints of shallow hierarchic sequential prediction do not harm performance. The ready application of such general prediction to syntactic parsing suggests that human language processing might be performed using a shallow hierarchic sequential process similar to those described in existing computational models of memory (Howard and Kahana, 2002; Botvinick, 2007).

Semantic disambiguation and reference grounding would presumably permit better results by providing a better approximation, though this is outside the scope of the current article. For example, this model may be extended to account for unbounded semantic dependencies such as filler-gap phenomena in a similar manner (Schuler, 2011).

Some combinations of operations in this model correspond to combinators in a Combinatory Categorial Grammar (CCG) (Steedman, 2000) in a maximally incremental parse. In particular:

- F+ followed by L– performs the CCG operation of forward type raising of x_t,

[7] The top-down nature of the Roark (2001) parser is not amenable to efficient use of the subcategorizations output by the split-merge-smooth algorithm.

[8] This is the recommended number of split-merge iterations to obtain high accuracy while avoiding overfitting (Petrov and Klein, 2007).

• F+ followed by L+ performs forward type raising of x_t followed by the CCG operation of forward function composition of a/b on this raised category, and

• F– performs the CCG operation of forward function application of a/b on x_t.

The model described in this article can therefore be taken as an exploration of the origins of combinators as a consequence of sequential and temporal cueing operations.[9]

Sturt and Lombardo (2005) warn about the need for constituents to be interconnected in processing, in order to pass along appropriate information to predict reading time delays. But the definition of connectivity they use includes underspecified or non-immediate relations such as the initial sub-sign relations described in this article, and is therefore more permissive than the immediate graph-theoretic notion of connectivity used to define the limits of connected components. In the model described in this article, all disjoint connected components within the same hypothesis are syntactically connected by Sturt and Lombardo's (2005) more permissive definition since the awaited signs of superordinate connected components are related by initial sub-sign relations ($\overset{+}{\rightarrow}$) to the active signs of subordinate connected components. Indeed, probabilistic operations can be defined to be dependent on other connected components in this model as well (although recall of superordinate connected components is dispreferred due to recency effects).

PCFGs provide a simple branching probabilistic model that can be used to generate complex syntactic structures and can be trained to predict these structures for novel sentences with state-of-the-art accuracy. This article shows that this kind of model can be incrementally processed in a straightforward way that is compatible with current assumptions about working memory. Carefully trained but computationally simple models like this may provide a framework for evaluating the contribution of recency and other memory-based effects on processing complexity (for example, costs associated with F– or L+ operations) on top of frequency effects currently measured by probabilistic metrics like surprisal, entropy reduction, and uniform information density.

References

Abney, S. P. and Johnson, M. (1991). Memory requirements and local ambiguities of parsing strategies. *J. Psycholinguistic Research*, 20(3):233–250.

Aho, A. V. and Ullman, J. D. (1972). *The Theory of Parsing, Translation and Compiling; Volume. I: Parsing*. Prentice-Hall, Englewood Cliffs, New Jersey.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, NJ.

Bjork, R. A. and Whitten, W. B. (1974). Recency sensitive retrieval processes in long-term free recall. *Cognitive Psychology*, 6.

Botvinick, M. (2007). Multilevel structure in behavior and in the brain: a computational model of fuster's hierarchy. *Philosophical Transactions of the Royal Society, Series B: Biological Sciences*, 362:1615–1626.

Collins, M. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL'97)*.

[9]However, nothing in this account predicts any of the more specialized combinators (like backward cross composition, involved in right-node raising and filler-gap constructions with non-peripheral gaps), or the constrained set of elementary categories (S and NP signifying truth values and entities) normally associated with CCG. Moreover, although recall of superordinate connected components is dispreferred due to recency effects, nothing in this account rules out conditional dependencies in productions that cross multiple connected components in the hierarchy, which would violate the CCG principle of adjacency for combinators, providing the original motivation for some of the more sophisticated CCG combinators such as backward cross composition. A more thorough exploration of some of these issues would be an interesting topic for future research.

Crocker, M. and Brants, T. (2000). Wide-coverage probabilistic sentence processing. *Journal of Psycholinguistic Research*, 29(6):647–669.

Crowder, R. G. (1982). The demise of short-term memory. *Acta Psychologica*, 50(3):291–323.

Gibson, E. (1991). *A computational theory of human linguistic processing: Memory limitations and processing breakdown*. PhD thesis, Carnegie Mellon.

Hale, J. (2001). A probabilistic earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 159–166, Pittsburgh, PA.

Hale, J. (2003). *Grammar, Uncertainty and Sentence Processing*. PhD thesis, Cognitive Science, The Johns Hopkins University.

Hale, J. (2006). Uncertainty about the rest of the sentence. *Cognitive Science*, 30(4):609–642.

Henderson, J. (2004). Lookahead in deterministic left-corner parsing. In *Proc. Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 26–33, Barcelona, Spain.

Howard, M. W. and Kahana, M. J. (2002). A distributed representation of temporal context. *Journal of Mathematical Psychology*, 45:269–299.

Johnson-Laird, P. N. (1983). *Mental models: towards a cognitive science of language, inference, and consciousness*. Harvard University Press, Cambridge, MA, USA.

Jurafsky, D. (1996). A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science: A Multidisciplinary Journal*, 20(2):137–194.

Levy, R. and Jaeger, F. T. (2007). Speakers optimize information density through syntactic reduction. In Schölkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA.

Lewis, R. L. and Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29(3):375–419.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL'06)*.

Petrov, S. and Klein, D. (2007). Improved inference for unlexicalized parsing. In *Proceedings of NAACL HLT 2007*, pages 404–411, Rochester, New York. Association for Computational Linguistics.

Roark, B. (2001). Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276.

Schuler, W. (2011). Effects of filler-gap dependencies on working memory requirements for parsing. In *Proceedings of COGSCI*, Boston, Massachusetts.

Schuler, W., AbdelRahman, S., Miller, T., and Schwartz, L. (2010). Broad-coverage incremental parsing using human-like memory constraints. *Computational Linguistics*, 36(1):1–30.

Steedman, M. (2000). *The syntactic process*. MIT Press/Bradford Books, Cambridge, MA.

Sturt, P. and Lombardo, V. (2005). Processing coordinate structures: Incrementality and connectedness. *Cognitive Science*, 29:291–305.