Connectionist-Inspired Incremental PCFG Parsing

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

Probabilistic context-free grammars (PCFGs) are a popular cognitive model of syntax (Jurafsky, 1996). These can be formulated to be sensitive to human working memory constraints by application of a right-corner transform (Schuler, 2009). One side-effect of the transform is that it guarantees at most a single expansion (push) and at most a single reduction (pop) during a syntactic parse. The primary finding of this paper is that this property of right-corner parsing can be exploited to obtain a dramatic reduction in the number of random variables in a probabilistic sequence model parser. This yields a simpler structure that more closely resembles existing simple recurrent network models of sentence comprehension.

1 Introduction

There may be a benefit to using insights from human cognitive modelling in parsing. Evidence for incremental processing can be seen in garden pathing (Bever, 1970), close shadowing (Marslen-Wilson, 1975), and eyetracking studies (Tanenhaus et al., 1995; Allopenna et al., 1998; Altmann and Kamide, 1999), which show humans begin attempting to process a sentence immediately upon receiving linguistic input. In the cognitive science community, this incremental interaction has often been modelled using recurrent neural networks (Elman, 1991; Mayberry and Miikkulainen, 2003), which utilize a hidden context with a severely bounded representational capacity (a fixed number of continuous units or dimensions), similar to models of activation-based memory in the prefrontal cortex (Botvinick, 2007), with the interesting possibility that the distributed behavior of neural columns (Horton and Adams, 2005) may directly implement continuous dimensions of recurrent hidden units. This paper presents a refinement of a factored probabilistic sequence model of comprehension (Schuler, 2009) in the direction of a recurrent neural network model and presents some observed efficiencies due to this refinement.

This paper will adopt an incremental probabilistic context-free grammar (PCFG) parser (Schuler, 2009) that uses a right-corner variant of the left-corner parsing strategy (Aho and Ullman, 1972) coupled with strict memory bounds, as a model of human-like parsing. Syntax can readily be approximated using simple PCFGs (Hale, 2001; Levy, 2008; Demberg and Keller, 2008), which can be easily tuned (Petrov and Klein, 2007). This paper will show that this representation can be streamlined to exploit the fact that a right-corner parse guarantees at most one expansion and at most one reduction can take place after each word is seen (see Section 2.2). The primary finding of this paper is that this property of right-corner parsing can be exploited to obtain a dramatic reduction in the number of random variables in a probabilistic sequence model parser (Schuler, 2009) yielding a simpler structure that more closely resembles connectionist models such as TRACE (McClelland and Elman, 1986), Shortlist (Norris, 1994; Norris and McQueen, 2008), or recurrent models (Elman, 1991; Mayberry and Miikkulainen, 2003) which posit functional units only for cognitively-motivated entities.

The rest of this paper is structured as follows: Section 2 gives the formal background of the right-corner parser transform and probabilistic sequence

model parsing. The simplification of this model is described in Section 3. A discussion of the interplay between cognitive theory and computational modelling in the resulting model may be found in Section 4. Finally, Section 5 demonstrates that such factoring also yields large benefits in the speed of probabilistic sequence model parsing.

2 Background

2.1 Notation

Throughout this paper, PCFG rules are defined over syntactic categories subscripted with abstract tree addresses ($c_{\eta\iota}$). These addresses describe a node's location as a path from a given ancestor node. A 0 on this path represents a leftward branch and a 1 a rightward branch. Positions within a tree are represented by subscripted η and ι so that $c_{\eta 0}$ is the left child of c_η and $c_{\eta 1}$ is the right child of c_η. The set of syntactic categories in the grammar is denoted by C. Finally, $[\![\phi]\!]$ denotes an *indicator* probability which is 1 if ϕ and 0 otherwise.

2.2 Right-Corner Parsing

Parsers such as that of Schuler (2009) model hierarchically deferred processes in working memory using a coarse analogy to a pushdown store indexed by an embedding depth d (to a maximum depth D). To make efficient use of this store, a CFG G must be transformed using a right-corner transform into another CFG G' with no right recursion. Given an optionally arc-eager attachment strategy, this allows the parser to clear completed parse constituents from the set of incomplete constituents in working memory much earlier than with a conventional syntactic structure. The right-corner transform operates deterministically over a CFG following three mapping rules:

$$\frac{c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1} \in G}{c_\eta / c_{\eta 1} \rightarrow c_{\eta 0} \in G'} \tag{1}$$

$$\frac{c_{\eta\iota} \rightarrow c_{\eta\iota 0}\ c_{\eta\iota 1} \in G,\ \ c_\eta \in C}{c_\eta / c_{\eta\iota 1} \rightarrow c_\eta / c_{\eta\iota}\ c_{\eta\iota 0} \in G'} \tag{2}$$

$$\frac{c_{\eta\iota} \rightarrow x_{\eta\iota} \in G,\ \ c_\eta \in C}{c_\eta \rightarrow c_\eta / c_{\eta\iota}\ c_{\eta\iota} \in G'} \tag{3}$$

A bottom-up incremental parsing strategy combined with the way the right-corner transform pulls each subtree into a left-expanding hierarchy ensures at most a single expansion (push) will occur at any given observation. That is, each new observation will be the leftmost leaf of a right-expanding subtree. Additionally, by reducing multiply right-branching subtrees to single rightward branches, the transform also ensures that at most a single reduction (pop) will take place at any given observation.

Schuler et al. (2010) show near complete coverage of the Wall Street Journal portion of the Penn Treebank (Marcus et al., 1993) can be achieved with a right-corner incremental parsing strategy using no more than four incomplete contituents (deferred processes), in line with recent estimates of human working memory capacity (Cowan, 2001).

Section 3 will show that, in addition to being desirable for bounded working memory restrictions, the single expansion/reduction guarantee reduces the search space between words to only two decision points — whether to expand and whether to reduce. This allows rapid processing of each candidate parse within a sequence modelling framework.

2.3 Model Formulation

This transform is then extended to PCFGs and integrated into a sequence model parser. Training on an annotated corpus yields the probability of any given syntactic state executing an expansion (creating a syntactic subtree) or a reduction (completing a syntactic subtree) to transition from every sufficiently probable (in this sense *active*) hypothesis in the working memory store.

The probability of the most likely sequence of store states $\hat{q}_{1..T}^{1..D}$ can then be defined as the product of nonterminal θ_Q, preterminal $\theta_{P,d}$, and terminal θ_X factors:

$$\hat{q}_{1..T}^{1..D} \stackrel{\text{def}}{=} \underset{q_{1..T}^{1..D}}{\operatorname{argmax}} \prod_{t=1}^{T} \mathsf{P}_{\theta_Q}(q_t^{1..D} \mid q_{t-1}^{1..D}\ p_{t-1}) \cdot \mathsf{P}_{\theta_{P,d'}}(p_t \mid b_t^{d'}) \cdot \mathsf{P}_{\theta_X}(x_t \mid p_t) \tag{4}$$

where all incomplete constituents q_t^d are factored into active a_t^d and awaited b_t^d components:

$$q_t^d \stackrel{\text{def}}{=} a_t^d / b_t^d \tag{5}$$

and d' determines the deepest non-empty incomplete constituent of $q_t^{1..D}$:

$$d' \stackrel{\text{def}}{=} max\{d \mid q_t^d \neq \text{`-'}\} \quad (6)$$

The preterminal model $\theta_{P,d}$ denotes the expectation of a subtree containing a given preterminal, expressed in terms of side- and depth-specific grammar rules $\mathsf{P}_{\theta_{Gs,d}}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1})$ and expected counts of left progeny categories $\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota}\ ...)$ (see Appendix A):

$$\mathsf{P}_{\theta_{P,d}}(c_{\eta\iota} \mid c_\eta) \stackrel{\text{def}}{=} \mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota}\ ...) \cdot \sum_{x_{\eta\iota}} \mathsf{P}_{\theta_{GL,d}}(c_{\eta\iota} \rightarrow x_{\eta\iota}) \quad (7)$$

and the terminal model θ_X is simply:

$$\mathsf{P}_{\theta_X}(x_\eta \mid c_\eta) \stackrel{\text{def}}{=} \frac{\mathsf{P}_{\theta_G}(c_\eta \rightarrow x_\eta)}{\sum_{x_\eta} \mathsf{P}_{\theta_G}(c_\eta \rightarrow x_\eta)} \quad (8)$$

The Schuler (2009) nonterminal model θ_Q is computed from a depth-specific store element model $\theta_{Q,d}$ and a large final state model $\theta_{F,d}$:

$$\mathsf{P}_{\theta_Q}(q_t^{1..D} \mid q_{t-1}^{1..D}\ p_{t-1}) \stackrel{\text{def}}{=} \sum_{f_t^{1..D}} \prod_{d=1}^{D} \mathsf{P}_{\theta_{F,d}}(f_t^d \mid f_t^{d+1}\ q_{t-1}^d\ q_{t-1}^{d-1}) \cdot \mathsf{P}_{\theta_{Q,d}}(q_t^d \mid f_t^{d+1}\ f_t^d\ q_{t-1}^d\ q_t^{d-1}) \quad (9)$$

After each time step t and depth d, θ_Q generates a set of final states to generate a new incomplete constituent q_t^d. These final states f_t^d are factored into categories $c_{f_t^d}$ and boolean variables (0 or 1) encoding whether a reduction has take place at depth d and time step t. The depth-specific final state model $\theta_{F,d}$ gives the probability of generating a final state f_t^d from the preceding q_t^d and q_t^{d-1} which is the probability of executing a reduction or consolidation of those incomplete constituents:

$$\mathsf{P}_{\theta_{F,d}}(f_t^d \mid f_t^{d+1}\ q_{t-1}^d\ q_{t-1}^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_t^{d+1} = \text{`-'} : [\![ f_t^d = 0 ]\!] \\ \text{if } f_t^{d+1} \neq \text{`-'} : \mathsf{P}_{\theta_{F,d,R}}(f_t^d \mid q_{t-1}^d\ q_{t-1}^{d-1}) \end{cases} \quad (10)$$

With these depth-specific f_t^d in hand, the model can calculate the probabilities of each possible q_t^d for each d and t based largely on the probability of transitions ($\theta_{Q,d,T}$) and expansions ($\theta_{Q,d,E}$) from the incomplete constituents at the previous time step:

$$\mathsf{P}_{\theta_{Q,d}}(q_t^d \mid f_t^{d+1}\ f_t^d\ q_{t-1}^d\ q_{t-1}^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_t^{d+1} = \text{`-'}, f_t^d = \text{`-'} : [\![q_t^d = q_{t-1}^d]\!] \\ \text{if } f_t^{d+1} \neq \text{`-'}, f_t^d = \text{`-'} : \mathsf{P}_{\theta_{Q,d,T}}(q_t^d \mid f_t^{d+1}\ f_t^d\ q_{t-1}^d\ q_t^{d-1}) \\ \text{if } f_t^{d+1} \neq \text{`-'}, f_t^d \neq \text{`-'} : \mathsf{P}_{\theta_{Q,d,E}}(q_t^d \mid q_t^{d-1}) \end{cases} \quad (11)$$

This model is shown graphically in Figure 1.

The probability distributions over reductions ($\theta_{F,d,R}$), transitions ($\theta_{Q,d,T}$) and expansions ($\theta_{Q,d,E}$) are then defined, also in terms of side- and depth-specific grammar rules $\mathsf{P}_{\theta_{Gs,d}}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1})$ and expected counts of left progeny categories $\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota}\ ...)$ (see Appendix A):

$$\mathsf{P}_{\theta_{Q,d,T}}(q_t^d \mid f_t^{d+1} f_t^d\ q_{t-1}^d q_t^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_t^d \neq \text{`-'} : \mathsf{P}_{\theta_{Q,d,A}}(q_t^d \mid q_t^{d-1} f_t^d) \\ \text{if } f_t^d = \text{`-'} : \mathsf{P}_{\theta_{Q,d,B}}(q_t^d \mid q_{t-1}^d f_t^{d+1}) \end{cases} \quad (12)$$

$$\mathsf{P}_{\theta_{F,d,R}}(f_t^d \mid f_t^{d+1} q_{t-1}^d q_{t-1}^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } c_{f_t^{d+1}} \neq x_t : [\![f_t^d = \text{`-'}]\!] \\ \text{if } c_{f_t^{d+1}} = x_t : \mathsf{P}_{\theta_{F,d,R}}(f_t^d \mid q_{t-1}^d q_{t-1}^{d-1}) \end{cases} \quad (13)$$

$$\mathsf{P}_{\theta_{Q,d,E}}(c_{\eta\iota} / c'_{\eta\iota} \mid \_ / c_\eta) \stackrel{\text{def}}{=} \mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota}\ ...) \cdot [\![x_{\eta\iota} = c'_{\eta\iota} = c_{\eta\iota}]\!] \quad (14)$$

The subcomponent models are obtained by applying the transform rules to all possible trees proportionately to their probabilities and marginalizing over all constituents that are not used in the models:

- for active transitions (from Transform Rule 1):

$$\mathsf{P}_{\theta_{Q,d,A}}(c_{\eta\iota} / c_{\eta\iota 1} \mid \_ / c_\eta\ c_{\eta\iota 0}) \stackrel{\text{def}}{=} \frac{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota}\ ...) \cdot \mathsf{P}_{\theta_{GL,d}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0}\ c_{\eta\iota 1})}{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{+}{\rightarrow} c_{\eta\iota 0}\ ...)} \quad (15)$$

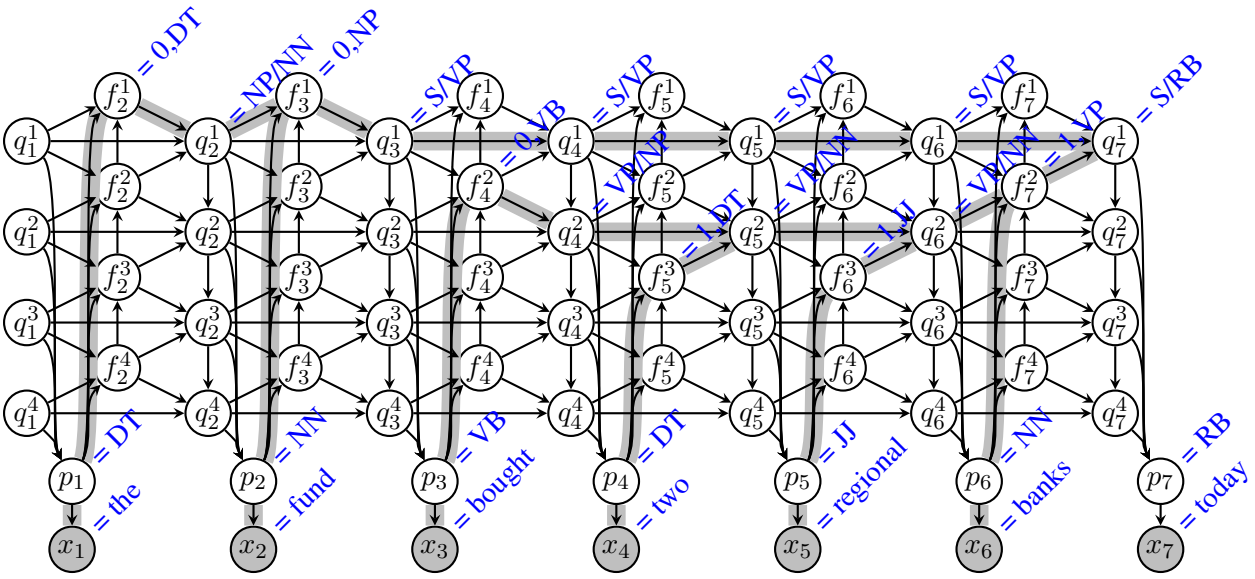

Figure 1: Schuler (2009) Sequence Model

- for awaited transitions (Transform Rule 2):

$$\mathsf{P}_{\theta_{Q,d,B}}(c_\eta / c_{\eta\iota 1} \mid c'_\eta / c_{\eta\iota}\ c_{\eta\iota 0}) \stackrel{\text{def}}{=} [\![c_\eta = c'_\eta]\!] \cdot \frac{\mathsf{P}_{\theta_{GR,d}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0}\ c_{\eta\iota 1})}{\mathsf{E}_{\theta_{G^*,d}}(c_{\eta\iota} \stackrel{0}{\rightarrow} c_{\eta\iota 0}\ \ldots)} \quad (16)$$

- for reductions (from Transform Rule 3):

$$\mathsf{P}_{\theta_{F,d,R}}(c_{\eta\iota}, \mathbf{1} \mid \_ / c_\eta\ c'_{\eta\iota} / \_) \stackrel{\text{def}}{=} [\![c_{\eta\iota} = c'_{\eta\iota}]\!] \cdot \frac{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{0}{\rightarrow} c_{\eta\iota}\ \ldots)}{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota}\ \ldots)} \quad (17)$$

$$\mathsf{P}_{\theta_{F,d,R}}(c_{\eta\iota}, \mathbf{0} \mid \_ / c_\eta\ c'_{\eta\iota} / \_) \stackrel{\text{def}}{=} [\![c_{\eta\iota} = c'_{\eta\iota}]\!] \cdot \frac{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{+}{\rightarrow} c_{\eta\iota}\ \ldots)}{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota}\ \ldots)} \quad (18)$$

3 Simplified Model

As seen in the previous section, the right-corner parser of Schuler (2009) makes the center embedding depth explicit and each memory store element is modelled as a combination of an active and an awaited component. Each input can therefore either increase (during an expansion) or decrease (during a reduction) the store of incomplete constituents or it can alter the active or awaited component of the deepest incomplete constituent (the *affectable* element). Alterations of the awaited component of the affectable element can be thought of as the expansion and immediate reduction of a syntactic constituent. The grammar models transitions in the active component implicitly, so these are conceptualized as consisting of neither an expansion nor a reduction.

Removing some of the variables in this model results in one that looks much more like a neural network (McClelland and Elman, 1986; Elman, 1991; Norris, 1994; Norris and McQueen, 2008) in that all remaining variables have cognitive correllates — in particular, they correspond to incomplete constituents in working memory — while still maintaining the ability to explicitly represent phrase structure. This section will demonstrate how it is possible to exploit this to obtain a large reduction in the number of modelled random variables.

In the Schuler (2009) sequence model, eight random variables are used to model the hidden states at each time step (see Figure 1). Half of these variables are *joint* consisting of two further (active and awaited) constituent variables, while the other half are merely over distributions of intermediate *final* states. Although the entire store is carried from time step to time step, only one memory element is affectable at any one time, and this element may be

reduced zero or one times (using an intermediate final state), and expanded zero or one times (using an incomplete constituent state), yielding four possible combinations. This means the model only actually needs one of its intermediate final states.

The transition model θ_Q can therefore be simplified with terms $\theta_{F,d}$ for the probability of expanding the incomplete constituent at d, and terms $\theta_{A,d}$ and $\theta_{B,d}$ for reducing the resulting constituent (defining the active and awaited components of a new incomplete constituent), along with terms for copying incomplete constituents above this affectable element, and for emptying the elements below it:

$$\begin{aligned}
&\mathsf{P}_{\theta_Q}(q_t^{1..D} \mid q_{t-1}^{1..D}\ p_{t-1}) \\
&\stackrel{\text{def}}{=} \mathsf{P}_{\theta_{F,d'}}(\text{`+'} \mid b_{t-1}^{d'}\ p_{t-1}) \cdot \mathsf{P}_{\theta_{A,d'}}(\text{`–'} \mid b_{t-1}^{d'-1}\ a_{t-1}^{d'}) \\
&\quad \cdot [\![a_t^{d'-1} = a_{t-1}^{d'-1}]\!] \cdot \mathsf{P}_{\theta_{B,d'-1}}(b_t^{d'-1} \mid b_{t-1}^{d'-1}\ a_{t-1}^{d'}) \\
&\quad \cdot [\![q_t^{1..d'-2} = q_{t-1}^{1..d'-2}]\!] \cdot [\![q_t^{d'..D} = \text{`–'}]\!] \\
&+ \mathsf{P}_{\theta_{F,d'}}(\text{`+'} \mid b_{t-1}^{d'}\ p_{t-1}) \cdot \mathsf{P}_{\theta_{A,d'}}(a_t^{d'} \mid b_{t-1}^{d'-1}\ a_{t-1}^{d'}) \\
&\quad \cdot \mathsf{P}_{\theta_{B,d'}}(b_t^{d'} \mid a_t^{d'}\ a_{t-1}^{d'+1}) \\
&\quad \cdot [\![q_t^{1..d'-1} = q_{t-1}^{1..d'-1}]\!] \cdot [\![q_t^{d'+1..D} = \text{`–'}]\!] \\
&+ \mathsf{P}_{\theta_{F,d'}}(\text{`–'} \mid b_{t-1}^{d'}\ p_{t-1}) \cdot \mathsf{P}_{\theta_{A,d'}}(\text{`–'} \mid b_{t-1}^{d'}\ p_{t-1}) \\
&\quad \cdot [\![a_t^{d'} = a_{t-1}^{d'}]\!] \cdot \mathsf{P}_{\theta_{B,d'}}(b_t^{d'} \mid b_{t-1}^{d'}\ p_{t-1}) \\
&\quad \cdot [\![q_t^{1..d'-1} = q_{t-1}^{1..d'-1}]\!] \cdot [\![q_t^{d'+1..D} = \text{`–'}]\!] \\
&+ \mathsf{P}_{\theta_{F,d'}}(\text{`–'} \mid b_{t-1}^{d'}\ p_{t-1}) \cdot \mathsf{P}_{\theta_{A,d'}}(a_t^{d'+1} \mid b_{t-1}^{d'}\ p_{t-1}) \\
&\quad \cdot \mathsf{P}_{\theta_{B,d'}}(b_t^{d'+1} \mid a_t^{d'+1}\ p_{t-1}) \\
&\quad \cdot [\![q_t^{1..d'} = q_{t-1}^{1..d'}]\!] \cdot [\![q_t^{d'+2..D} = \text{`–'}]\!] \qquad (19)
\end{aligned}$$

The first element of the sum in Equation 19 computes the probability of a reduction with no expansion (decreasing d'). The second corresponds to the probability of a store undergoing neither an expansion nor a reduction (a transition to a new active constituent at the same embedding depth). In the third is the probability of an expansion and a reduction (a transition among awaited constituents at the same embedding depth). Finally, the last term yields the probability of an expansion without a reduction (increasing d').

From Equation 19 it may be seen that the unaffected store elements of each time step are maintained sans change as guaranteed by the single-reduction feature of the right-corner parser. This results in a large representational economy by making the majority of store state decisions deterministic. This representational economy will later translate into computational efficiencies (see section 5). In this sense, cognitive modelling contributes to a practical speed increase.

Since the bulk of the state remains the same, the recognizer can access the affectable variable and operate solely over the transition possibilities from that variable to calculate the distribution over store states for the next time step to explore. Reflecting this change, the hidden states now model a single final-state variable (f) for results of the expansion decision, and the affectable variable resulting from the reduction decision (both its active (a) and awaited (b) categories), as well as the preterminal state (p) defined in the previous section. These models are again expressed in terms of side- and depth-specific grammar rules $\mathsf{P}_{\theta_{Gs,d}}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1})$ and expected counts of left progeny categories $\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota}\ \ldots)$ (see Appendix A).

Expansion probabilities are modelled as a binary decision depending on whether or not the awaited component of the affectable variable c_η is likely to expand immediately into an anticipated preterminal $c_{\eta\iota}$ (resulting in a non-empty final state: '+') or if intervening embeddings are necessary given the affectable active component (yielding no final state: '–'):

$$\mathsf{P}_{\theta_{F,d}}(f \mid c_\eta\ c_{\eta\iota}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f = \text{`+'} : \dfrac{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{0}{\rightarrow} c_{\eta\iota}\ \ldots)}{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota}\ \ldots)} \\ \text{if } f = \text{`–'} : \dfrac{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{+}{\rightarrow} c_{\eta\iota}\ \ldots)}{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota}\ \ldots)} \end{cases} \qquad (20)$$

The active component category $c_{\eta\iota}$ is defined as depending on the category of the awaited component above it c_η and its left-hand child $c_{\eta\iota 0}$:

$$\begin{aligned}
\mathsf{P}_{\theta_{A,d}}(c_{\eta\iota} \mid c_\eta\ c_{\eta\iota 0}) &\stackrel{\text{def}}{=} \frac{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{1}{\rightarrow} c_{\eta\iota 0}\ \ldots)}{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{+}{\rightarrow} c_{\eta\iota 0}\ \ldots)} \cdot [\![c_{\eta\iota} = \text{`–'}]\!] \\
&+ \frac{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{+}{\rightarrow} c_{\eta\iota}\ \ldots) \cdot \mathsf{P}_{\theta_{GL,d}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0}\ \ldots)}{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{+}{\rightarrow} c_{\eta\iota 0}\ \ldots)} \qquad (21)
\end{aligned}$$

The awaited component category $c_{\eta 1}$ is defined as

depending on the category of its parent c_η and the preceding sibling $c_{\eta 0}$:

$$\mathsf{P}_{\theta_{B,d}}(c_{\eta 1} \mid c_\eta\, c_{\eta 0}) \stackrel{\text{def}}{=} \frac{\mathsf{P}_{\theta_{GR,d}}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1})}{\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{1}{\rightarrow} c_{\eta 0}\ \ldots)} \quad (22)$$

Both of these make sense given the manner in which the right-corner parser shifts dependencies to the left down the tree in order to obtain incremental information about upcoming constituents.

3.1 Graphical Representation

In order to be represented graphically, the working memory store θ_Q is factored into a single expansion term θ_F and a product of depth-specific reduction terms $\theta_{Q,d}$:

$$\mathsf{P}_{\theta_Q}(q_t^{1..D} \mid q_{t-1}^{1..D}\, p_{t-1}) \stackrel{\text{def}}{=} \sum_{f_t} \mathsf{P}_{\theta_F}(f_t \mid q_{t-1}^{1..D}) \cdot \prod_{d=1}^{D} \mathsf{P}_{\theta_{Q,d}}(q_t^d \mid q_{t-1}^{1..D}\, p_{t-1}\, f_t\, q_t^{d+1}) \quad (23)$$

and the depth-specific reduction model $\theta_{Q,d}$ is factored into individual decisions over each random variable:

$$\mathsf{P}_{\theta_{Q,d}}(q_t^d \mid q_{t-1}^{1..D}\, p_{t-1}\, f_t\, q_t^{d+1}) \stackrel{\text{def}}{=}$$
$$\begin{cases} \text{if } q_t^{d+1} = \text{`–'}, f_t \neq \text{`–'}, d = d'-1: \\ \quad [\![a_t^d = a_{t-1}^d]\!] \cdot \mathsf{P}_{\theta_{B,d}}(b_t^d \mid b_{t-1}^d\, a_{t-1}^{d+1}) \\ \text{if } q_t^{d+1} = \text{`–'}, f_t \neq \text{`–'}, d = d': \\ \quad \mathsf{P}_{\theta_{A,d}}(a_t^d \mid b_{t-1}^{d-1}\, a_{t-1}^d) \cdot \mathsf{P}_{\theta_{B,d}}(b_t^d \mid a_t^d\, a_{t-1}^d) \\ \text{if } q_t^{d+1} = \text{`–'}, f_t = \text{`–'}, d = d': \\ \quad [\![a_t^d = a_{t-1}^d]\!] \cdot \mathsf{P}_{\theta_{B,d}}(b_t^d \mid b_{t-1}^d\, p_{t-1}) \\ \text{if } q_t^{d+1} = \text{`–'}, f_t = \text{`–'}, d = d'+1: \\ \quad \mathsf{P}_{\theta_{A,d}}(a_t^d \mid b_{t-1}^{d-1}\, p_{t-1}) \cdot \mathsf{P}_{\theta_{B,d}}(b_t^d \mid a_t^d\, p_{t-1}) \\ \text{if } q_t^{d+1} \neq \text{`–'}: [\![q_t^d = q_{t-1}^d]\!] \\ \text{otherwise}: [\![q_t^d = \text{`–'}]\!] \end{cases} \quad (24)$$

This dependency structure is represented graphically in Figure 2.

The first conditional in Equation 24 checks whether the input causes a reduction but no expansion (completing a subtree parse). In this case, d' is reduced from the previous t, and the relevant q_{t-1}^d is copied to q_t^d except the awaited constituent is altered to reflect the completion of its preceding awaited subtree. In the second case, the parser makes an active transition as it completes a left subtree and begins exploring the right subtree. The third case is similar to the first except it transitions between two like depths (awaited transition), and depends on the preterminal just seen to contrive a new subtree to explore. In the fourth case, d' is incremented as another incomplete constituent opens up in working memory. The final two cases simply update the unaffected store states to reflect their previous states at time $t-1$.

4 Discussion

This factoring of redundant hidden states out of the Schuler (2009) probabilistic sequence model shows that cognitive modelling can more closely approximate a simple recurrent network model of language processing (Elman, 1991). Probabilistic sequence model parsers have previously been modelled with random variables over incomplete constituents (Schuler, 2009). In the current implementation, each variable can be thought of as a bank of artificial neurons. These artificial neurons inhibit one another through the process of normalization. Conversely, they activate artificial neurons at subsequent time steps by contributing probability mass through the transformed grammar. This point was made by Norris and McQueen (2008) with respect to lexical access; this model extends it to parsing.

Recurrent networks can parse simple sentences but run into problems when running over more complex datasets. This limitation comes from the unsupervised methods typically used to train them, which have difficulty scaling to sufficiently large training sets for more complex constructions. The approach described in this paper uses a hidden context similar to that of a recurrent network to inform the progression of the parse, except that the context is in terms of random variables with distributions over a set of explicit syntactic categories. By framing the variable domains in a linguistically-motivated fashion, the problem of acquisition can be divested from the problem of processing. This paper then uses the semi-supervised grammar training of Petrov et al. (2006) in order to develop a simple, accurate model for broad-coverage parsing independent of scale.

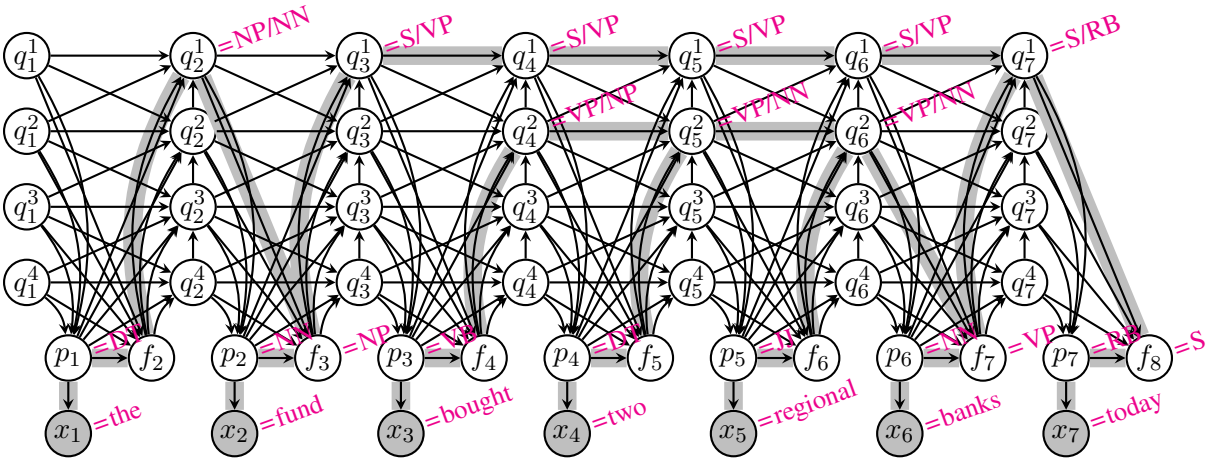

Figure 2: Parse using Simplified Model

Like Schuler (2009), the incremental parser discussed here operates in $O(n)$ time where n is the length of the input. Further, by its incremental nature, this parser is able to run continuously on a stream of input, which allows any other processes dependent on the input (such as discourse integration) to run in parallel regardless of the length of the input.

5 Computational Benefit

Due to the decreased number of decisions required by this simplified model, it is substantially faster than previous similar models. To test this speed increase, the simplified model was compared with that of Schuler (2009). Both parsers used a grammar that had undergone 5 iterations of the Petrov et al. (2006) split-merge-smooth algorithm as found to be optimal by Petrov and Klein (2007), and both used a beam-width of 500 elements. Sections 02-21 of the Wall Street Journal Treebank were used in training the grammar induction for both parsers according to Petrov et al. (2006), and Section 23 was used for evaluation. No tuning was done as part of the transform to a sequence model. Speed results can be seen in Table 1. While the speed is not state-of-the-art in the field of parsing at large, it does break new ground for factored sequence model parsers.

To test the accuracy of this parser, it was compared using varying beam-widths to the Petrov and Klein (2007) and Roark (2001) parsers. With the exception of the Roark (2001) parser, all parsers used 5 iterations of the Petrov et al. (2006) split-merge-smooth algorithm, and the training and testing datasets remained the same. These results may be seen in Table 2. Note that the Petrov and Klein (2007) parser allows unary branching within the phrase structure, which is not well-defined under the right-corner transform. To obtain a fair comparison, it was also run with strict binarization. The current approach achieves comparable accuracy to the Petrov and Klein (2007) parser assuming a strictly binary-branching phrase structure.

System	Sec/Sent
Schuler 2009	74
Current Model	12
Speed Increase	5.16x

Table 1: Speed comparison with an unfactored probabilistic sequence model using a beam-width of 500 elements

System	P	R	F
Roark 2001	86.6	86.5	86.5
Current Model (500)	86.6	87.3	87.0
Current Model (2000)	87.8	87.8	87.8
Current Model (5000)	87.8	87.8	87.8
Petrov Klein (Binary)	88.1	87.8	88.0
Petrov Klein (+Unary)	88.3	88.6	88.5

Table 2: Accuracy comparison with state-of-the-art models. Numbers in parentheses are number of parallel activated hypotheses

6 Conclusion

The primary goal of this paper was to demonstrate that a cognitively-motivated factoring of an existing probabilistic sequence model parser (Schuler, 2009) is not only more attractive from a modelling perspective but also more efficient. Such factoring yields a much slimmer model where every variable has cognitive correlates to working memory elements. This also renders several transition probabilities deterministic and the ensuing representational economy leads to a 5-fold increase in parsing speed. The results shown here suggest cognitive modelling can lead to computational benefits.

References

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

P. D. Allopenna, J. S. Magnuson, and M. K. Tanenhaus. 1998. Tracking the time course of spoken word recognition using eye movements: evidence for continuous mapping models. *Journal of Memory and Language*, 38:419–439.

G. T. M. Altmann and Y. Kamide. 1999. Incremental interpretation at verbs: restricting the domain of subsequent reference. *Cognition*, 73:247–264.

Richard Bellman. 1957. *Dynamic Programming*. Princeton University Press, Princeton, NJ.

Thomas G. Bever. 1970. The cognitive basis for linguistic structure. In J.R̃. Hayes, editor, *Cognition and the Development of Language*, pages 279–362. Wiley, New York.

Matthew Botvinick. 2007. Multilevel structure in behavior and in the brain: a computational model of fusters hierarchy. *Philosophical Transactions of the Royal Society, Series B: Biological Sciences*, 362:1615–1626.

Nelson Cowan. 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24:87–185.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210.

Jeffrey L. Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7:195–225.

John Hale. 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.

Jonathan C Horton and Daniel L Adams. 2005. The cortical column: a structure without a function. *Philosophical Transactions of the Royal Society of London - Series B: Biological Sciences*, 360(1456):837–862.

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

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

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

William D. Marslen-Wilson. 1975. Sentence perception as an interactive parallel process. *Science*, 189(4198):226–228.

Marshall R. Mayberry, III and Risto Miikkulainen. 2003. Incremental nonmonotonic parsing through semantic self-organization. In *Proceedings of the* 25^{th} *Annual Conference of the Cognitive Science Society*, pages 798–803, Boston, MA.

James L. McClelland and Jeffrey L. Elman. 1986. The trace model of speech perception. *Cognitive Psychology*, 18:1–86.

Dennis Norris and James M. McQueen. 2008. Shortlist b: A bayesian model of continuous speech recognition. *Psychological Review*, 115(2):357–395.

Dennis Norris. 1994. Shortlist: A connectionist model of continuous speech recognition. *Cognition*, 52:189–234.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of NAACL HLT 2007*, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.

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

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

William Schuler, Samir AbdelRahman, Tim Miller, and Lane Schwartz. 2010. Broad-coverage incremental parsing using human-like memory constraints. *Computational Linguistics*, 36(1):1–30.

William Schuler. 2009. Parsing with a bounded stack using a model-based right-corner transform. In *Proceedings of NAACL/HLT 2009*, NAACL '09, pages 344–352, Boulder, Colorado. Association for Computational Linguistics.

Michael K. Tanenhaus, Michael J. Spivey-Knowlton, Kathy M. Eberhard, and Julie E. Sedivy. 1995. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268:1632–1634.

A Grammar Formulation

Given D memory elements indexed by d (see Section 2.2) and a PCFG θ_G, the probability $\theta^{(k)}_{Ts,d}$ of a tree rooted at a left or right sibling $s \in \{L, R\}$ of category $c_\eta \in C$ requiring $d \in 1..D$ memory elements is defined recursively over paths of increasing length k:

$$\mathsf{P}_{\theta^{(0)}_{Ts,d}}(1 \mid c_\eta) \stackrel{\text{def}}{=} 0 \tag{25}$$

$$\begin{aligned}\mathsf{P}_{\theta^{(k)}_{TL,d}}(1 \mid c_\eta) \stackrel{\text{def}}{=} & \sum_{x_\eta} \mathsf{P}_{\theta_G}(c_\eta \rightarrow x_\eta) \\ & + \sum_{c_{\eta 0}, c_{\eta 1}} \mathsf{P}_{\theta_G}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1}) \\ & \cdot \mathsf{P}_{\theta^{(k-1)}_{TL,d}}(1 \mid c_{\eta 0}) \cdot \mathsf{P}_{\theta^{(k-1)}_{TR,d}}(1 \mid c_{\eta 1})\end{aligned} \tag{26}$$

$$\begin{aligned}\mathsf{P}_{\theta^{(k)}_{TR,d}}(1 \mid c_\eta) \stackrel{\text{def}}{=} & \sum_{x_\eta} \mathsf{P}_{\theta_G}(c_\eta \rightarrow x_\eta) \\ & + \sum_{c_{\eta 0}, c_{\eta 1}} \mathsf{P}_{\theta_G}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1}) \\ & \cdot \mathsf{P}_{\theta^{(k-1)}_{TL,d+1}}(1 \mid c_{\eta 0}) \cdot \mathsf{P}_{\theta^{(k-1)}_{TR,d}}(1 \mid c_{\eta 1})\end{aligned} \tag{27}$$

Note that the center embedding depth d increases only for left children of right children. This is because in a binary branching structure, center embeddings manifest as zigzags. Since the model is also sensitive to the depth d of each decomposition, the side- and depth-specific probabilities of $\theta_{GL,d}$ and $\theta_{GR,d}$ are defined as follows:

$$\mathsf{P}_{\theta_{GL,d}}(c_\eta \rightarrow x_\eta) \stackrel{\text{def}}{=} \frac{\mathsf{P}_{\theta_G}(c_\eta \rightarrow x_\eta)}{\mathsf{P}_{\theta^{(\infty)}_{TL,d}}(1 \mid c_\eta)} \tag{28}$$

$$\mathsf{P}_{\theta_{GR,d}}(c_\eta \rightarrow x_\eta) \stackrel{\text{def}}{=} \frac{\mathsf{P}_{\theta_G}(c_\eta \rightarrow x_\eta)}{\mathsf{P}_{\theta^{(\infty)}_{TR,d}}(1 \mid c_\eta)} \tag{29}$$

$$\begin{aligned}\mathsf{P}_{\theta_{GL,d}}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1}) \stackrel{\text{def}}{=} & \mathsf{P}_{\theta_G}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1}) \\ & \cdot \mathsf{P}_{\theta^{(\infty)}_{TL,d}}(1 \mid c_{\eta 0}) \cdot \mathsf{P}_{\theta^{(\infty)}_{TR,d}}(1 \mid c_{\eta 1}) \\ & \cdot \mathsf{P}_{\theta^{(\infty)}_{TL,d}}(1 \mid c_\eta)^{-1}\end{aligned} \tag{30}$$

$$\begin{aligned}\mathsf{P}_{\theta_{GR,d}}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1}) \stackrel{\text{def}}{=} & \mathsf{P}_{\theta_G}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1}) \\ & \cdot \mathsf{P}_{\theta^{(\infty)}_{TL,d+1}}(1 \mid c_{\eta 0}) \cdot \mathsf{P}_{\theta^{(\infty)}_{TR,d}}(1 \mid c_{\eta 1}) \\ & \cdot \mathsf{P}_{\theta^{(\infty)}_{TR,d}}(1 \mid c_\eta)^{-1}\end{aligned} \tag{31}$$

The model will also need an expected count $\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota}\ ...)$ of the given child constituent $c_{\eta\iota}$ dominating a prefix of constituent c_η. Expected versions of these counts may later be used to derive probabilities of memory store state transitions (see Sections 2.3, 3).

$$\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{0}{\rightarrow} c_\eta\ ...) \stackrel{\text{def}}{=} \sum_{x_\eta} \mathsf{P}_{\theta_{GR,d}}(c_\eta \rightarrow x_\eta) \tag{32}$$

$$\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{1}{\rightarrow} c_{\eta 0}\ ...) \stackrel{\text{def}}{=} \sum_{c_{\eta 1}} \mathsf{P}_{\theta_{GR,d}}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1}) \tag{33}$$

$$\begin{aligned}\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{k}{\rightarrow} c_{\eta\iota 0}\ ...) \stackrel{\text{def}}{=} & \sum_{c_{\eta\iota}} \mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{k-1}{\rightarrow} c_{\eta\iota}\ ...) \\ & \cdot \sum_{c_{\eta\iota 1}} \mathsf{P}_{\theta_{GL,d}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0}\ c_{\eta\iota 1})\end{aligned} \tag{34}$$

$$\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota}\ ...) \stackrel{\text{def}}{=} \sum_{k=0}^{\infty} \mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{k}{\rightarrow} c_{\eta\iota}\ ...) \tag{35}$$

$$\mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{+}{\rightarrow} c_{\eta\iota}\ ...) \stackrel{\text{def}}{=} \sum_{k=1}^{\infty} \mathsf{E}_{\theta_{G^*,d}}(c_\eta \stackrel{k}{\rightarrow} c_{\eta\iota}\ ...) \tag{36}$$

Equation 32 gives the probability of a constituent appearing as an observation, and Equation 33 gives the probability of a constituent appearing as a left

child. Equation 34 extends the previous two equations to account for a constituent appearing at an arbitrarily deep embedded path of length k. Taking the sum of all k path lengths (as in Equation 35) allows the model to account for constituents anywhere in the left progeny of the dominated subtree. Similarly, Equation 36 gives the expectation that the constituent is non-immediately dominated by c_η. In practice the infinite sum is estimated to some constant K using value iteration (Bellman, 1957).