Positive Results for Parsing with a Bounded Stack using a Model-Based Right-Corner Transform

William Schuler
Dept. of Computer Science and Engineering
Minneapolis, MN
schuler@cs.umn.edu

Abstract

Statistical parsing models have recently been proposed that employ a bounded stack in time-series (left-to-right) recognition, using a right-corner transform defined over training trees to minimize stack use (Schuler et al., 2008). Corpus results have shown that a vast majority of naturally-occurring sentences can be parsed in this way using a very small stack bound of three to four elements. This suggests that the standard cubic-time CKY chart-parsing algorithm, which implicitly assumes an *unbounded* stack, may be wasting probability mass on trees whose complexity is beyond human recognition or generation capacity. This paper first describes a version of the right-corner transform that is defined over entire probabilistic grammars (cast as infinite sets of generable trees), in order to ensure a fair comparison between bounded-stack and unbounded PCFG parsing using a common underlying model; then it presents experimental results that show a bounded-stack right-corner parser using a transformed version of a grammar significantly outperforms an unbounded-stack CKY parser using the original grammar.

1 Introduction

Statistical parsing models have recently been proposed that employ a bounded stack in time-series (left-to-right) recognition, in order to directly and tractably incorporate incremental phenomena such as (co-)reference or disfluency into parsing decisions (Schuler et al., 2008; Miller and Schuler, 2008). These models make use of a right-corner tree transform, based on the left-corner transform described by Johnson (1998), and are supported by corpus results suggesting that most sentences (in English, at least) can be parsed using a very small stack bound of three to four elements (Schuler et al., 2008). This raises an interesting question: if most sentences can be recognized with only three or four elements of stack memory, is the standard cubic-time CKY chart-parsing algorithm, which implicitly assumes an *unbounded* stack, wasting probability mass on trees whose complexity is beyond human recognition or generation capacity?

This paper presents parsing accuracy results using transformed and untransformed versions of a corpus-trained probabilistic context-free grammar suggesting that this is indeed the case. Experimental results show a bounded-memory parser using a transformed version of a grammar significantly outperforms an unbounded-stack CKY parser using the original grammar.

Unlike the tree-based transforms described previously, the model-based transform described in this paper does not introduce additional context from corpus data beyond that contained in the original probabilistic grammar, making it possible to present a fair comparison between bounded- and unbounded-stack versions of the same model. Since this transform takes a probabilistic grammar as input, it can also easily accommodate horizontal and vertical Markovisation (annotating grammar symbols with parent and sibling categories) as described by Collins (1997) and subsequently.

The remainder of this paper is organized as follows: Section 2 describes related approaches to parsing with stack bounds; Section 3 describes an existing bounded-stack parsing framework using a right-corner transform defined over individual trees; Section 4 describes a redefinition of this transform to ap-

ply to entire probabilistic grammars, cast as infinite sets of generable trees; and Section 5 describes an evaluation of this transform on the Wall Street Journal corpus of the Penn Treebank showing improved results for a transformed bounded-stack version of a probabilistic grammar over the original unbounded grammar.

2 Related Work

The model examined here is formally similar to Combinatorial Categorial Grammar (CCG) (Steedman, 2000). But the CCG account is a competence model as well as a performance model, in that it seeks to unify category representations used in processing with learned generalizations about argument structure; whereas the model described in this paper is exclusively a performance model, allowing generalizations about lexical argument structures to be learned in some other representation, then combined with probabilistic information about parsing strategies to yield a set of derived incomplete constituents. As a result, the model described in this paper has a freer hand to satisfy strict working memory bounds, which may not permit some of the alternative composition operations proposed in the CCG account, thought to be associated with available prosody and quantifier scope analyses.[1]

Other models (Abney and Johnson, 1991; Gibson, 1991) seek to explain human processing difficulties as a result of memory capacity limits in parsing ordinary phrase structure trees. The Abney-Johnson and Gibson models adopt a left-corner parsing strategy, of which the right-corner transform described in this paper is a variant, in order to minimize memory usage. But the transform-based model described in this paper exploits a conception of chunking (Miller, 1956) — in this case, grouping recognized words into stacked-up incomplete constituents — to operate within much stricter estimates of human short-term memory bounds (Cowan, 2001) than assumed by Abney and Johnson.

[1]The lack of support for some of these available scope analyses may not necessarily be problematic for the present model. The complexity of interpreting nested raised quantifiers may place them beyond the capability of human interactive incremental interpretation, but not beyond the capability of post-hoc interpretation (understood after the listener has had time to think about it).

Several existing incremental systems are organized around a left-corner parsing strategy (Roark, 2001; Henderson, 2004). But these systems generally keep large numbers of constituents open for modifier attachment in each hypothesis. This allows modifiers to be attached as right children of any such open constituent. But if any number of open constituents are allowed, then either the assumption that stored elements have fixed syntactic (and semantic) structure will be violated, or the assumption that syntax operates within a bounded memory store will be violated, both of which are psycholinguistically attractive as simplifying assumptions. The HHMM model examined in this paper upholds both the fixed-element and bounded-memory assumptions by hypothesizing fixed reductions of right child constituents into incomplete parents in the same memory element, to make room for new constituents that may be introduced at a later time. These in-element reductions are defined naturally on phrase structure trees as the result of aligning right-corner transformed constituent structures to sequences of random variables in a factored time-series model.

3 Background

The recognition model examined in this paper is a factored time-series model, based on a Hierarchic Hidden Markov Model (Murphy and Paskin, 2001), which probabilistically estimates the contents of a memory store of three to four partially-completed constituents over time. Probabilities for expansions, transitions and reductions in this model can be defined over trees in a training corpus, transformed and mapped to the random variables in an HHMM (Schuler et al., 2008). In Section 4 these probabilities will be computed directly from a probabilistic context-free grammar, in order to evaluate the contribution of stack bounds without introducing additional corpus context into the model.

3.1 A Bounded-Stack Model

HHMMs are factored HMMs which mimic a bounded-memory pushdown automaton (PDA), supporting simple push and pop operations on a bounded stack-like memory store.

HMMs characterize speech or text as a sequence

of hidden states q_t (in this case, stacked-up syntactic categories) and observed states o_t (in this case, words) at corresponding time steps t. A most likely sequence of hidden states $\hat{q}_{1..T}$ can then be hypothesized given any sequence of observed states $o_{1..T}$:

$$\hat{q}_{1..T} = \underset{q_{1..T}}{\operatorname{argmax}} \mathsf{P}(q_{1..T} \mid o_{1..T}) \quad (1)$$

$$= \underset{q_{1..T}}{\operatorname{argmax}} \mathsf{P}(q_{1..T}) \cdot \mathsf{P}(o_{1..T} \mid q_{1..T}) \quad (2)$$

$$\stackrel{\text{def}}{=} \underset{q_{1..T}}{\operatorname{argmax}} \prod_{t=1}^{T} \mathsf{P}_{\Theta_{\text{A}}}(q_t \mid q_{t-1}) \cdot \mathsf{P}_{\Theta_{\text{B}}}(o_t \mid q_t) \quad (3)$$

using Bayes' Law (Equation 2) and Markov independence assumptions (Equation 3) to define a full $\mathsf{P}(q_{1..T} \mid o_{1..T})$ probability as the product of a *Transition Model (Θ_{A})* prior probability $\mathsf{P}(q_{1..T}) \stackrel{\text{def}}{=} \prod_t \mathsf{P}_{\Theta_{\text{A}}}(q_t \mid q_{t-1})$ and an *Observation Model (Θ_{B})* likelihood probability $\mathsf{P}(o_{1..T} \mid q_{1..T}) \stackrel{\text{def}}{=} \prod_t \mathsf{P}_{\Theta_{\text{B}}}(o_t \mid q_t)$.

Transition probabilities $\mathsf{P}_{\Theta_{\text{A}}}(q_t \mid q_{t-1})$ over complex hidden states q_t can be modeled using synchronized levels of stacked-up component HMMs in an HHMM. HHMM transition probabilities are calculated in two phases: a *reduce* phase (resulting in an intermediate, marginalized state f_t), in which component HMMs may terminate; and a *shift* phase (resulting in a modeled state q_t), in which unterminated HMMs transition, and terminated HMMs are reinitialized from their parent HMMs. Variables over intermediate f_t and modeled q_t states are factored into sequences of depth-specific variables – one for each of D levels in the HHMM hierarchy:

$$f_t = \langle f_t^1 \dots f_t^D \rangle \quad (4)$$

$$q_t = \langle q_t^1 \dots q_t^D \rangle \quad (5)$$

Transition probabilities are then calculated as a product of transition probabilities at each level, using level-specific *reduce* $\Theta_{\text{R},d}$ and *shift* $\Theta_{\text{S},d}$ models:

$$\mathsf{P}_{\Theta_{\text{A}}}(q_t | q_{t-1}) = \sum_{f_t} \mathsf{P}(f_t | q_{t-1}) \cdot \mathsf{P}(q_t | f_t\, q_{t-1}) \quad (6)$$

$$\stackrel{\text{def}}{=} \sum_{f_t^{1..D}} \prod_{d=1}^{D} \mathsf{P}_{\Theta_{\text{R},d}}(f_t^d | f_t^{d-1} q_{t-1}^d q_{t-1}^{d-1}) \cdot \mathsf{P}_{\Theta_{\text{S},d}}(q_t^d | f_t^{d-1} f_t^d\, q_{t-1}^d q_t^{d-1}) \quad (7)$$

with f_t^{D+1} and q_t^0 defined as constants. In Viterbi decoding, the sums are replaced with argmax operators. This decoding process preserves ambiguity by maintaining competing analyses of the entire memory store. A graphical representation of an HHMM with three levels is shown in Figure 1.

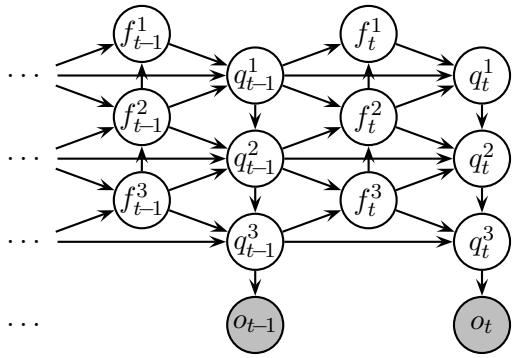


Figure 1: Graphical representation of a Hierarchic Hidden Markov Model. Circles denote random variables, and edges denote conditional dependencies. Shaded circles are observations.

Shift and reduce probabilities can then be defined in terms of finitely recursive Finite State Automata (FSAs) with probability distributions over transition, recursive expansion, and final-state status of states at each hierarchy level. In the version of HHMMs used in this paper, each intermediate variable is a reduction or non-reduction state $f_t^d \in G \cup \{\mathbf{1}, \mathbf{0}\}$ (indicating, respectively, a complete reduced constituent of some grammatical category from domain G, or a failure to reduce due to an 'active' transition being performed, or a failure to reduce due to an 'awaited' transition being performed, as defined in Section 4.3); and each modeled variable is a syntactic state $q_t^d \in G \times G$ (describing an incomplete constituent consisting of an active grammatical category from domain G and an awaited grammatical category from domain G). An intermediate variable f_t^d at depth d may indicate reduction or non-reduction according to $\Theta_{\text{F-Rd},d}$ if there is a reduction at the depth level immediately below d, but must indicate non-reduction ($\mathbf{0}$) with probability 1 if there was no reduction below:[2]

$$\mathsf{P}_{\Theta_{\text{R},d}}(f_t^d \mid f_t^{d+1} q_{t-1}^d q_{t-1}^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_t^{d+1} \notin G : [f_t^d = \mathbf{0}] \\ \text{if } f_t^{d+1} \in G : \mathsf{P}_{\Theta_{\text{F-Rd},d}}(f_t^d \mid q_{t-1}^d\, q_{t-1}^{d-1}) \end{cases} \quad (8)$$

[2]Here $[\cdot]$ is an indicator function: $[\phi] = 1$ if ϕ is true, 0 otherwise.

where $f_t^{D+1} \in G$ and $q_t^0 = \mathbf{ROOT}$.

Shift probabilities over the modeled variable q_t^d at each level are defined using level-specific transition $\Theta_{\text{Q-Tr},d}$ and expansion $\Theta_{\text{Q-Ex},d}$ models:

$$\mathsf{P}_{\Theta_{\text{S},d}}(q_t^d \mid f_t^{d+1}\, f_t^d\, q_{t-1}^d\, q_t^{d-1}) \stackrel{\text{def}}{=}$$
$$\begin{cases} \text{if } f_t^{d+1} \notin G,\ f_t^d \notin G\text{: } [q_t^d = q_{t-1}^d] \\ \text{if } f_t^{d+1} \in G,\ f_t^d \notin G\text{: } \mathsf{P}_{\Theta_{\text{Q-Tr},d}}(q_t^d \mid f_t^{d+1}\, f_t^d\, q_{t-1}^d\, q_t^{d-1}) \\ \text{if } f_t^{d+1} \in G,\ f_t^d \in G\text{: } \mathsf{P}_{\Theta_{\text{Q-Ex},d}}(q_t^d \mid q_t^{d-1}) \end{cases} \quad (9)$$

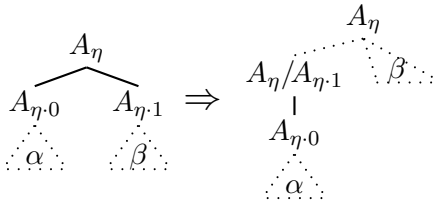
where $f_t^{D+1} \in G$ and $q_t^0 = \mathbf{ROOT}$. This model is conditioned on reduce variables at and immediately below the current FSA level. If there is no reduction immediately below the current level (the first case above), it deterministically copies the current FSA state forward to the next time step. If there is a reduction immediately below the current level but no reduction at the current level (the second case above), it transitions the FSA state at the current level, according to the distribution $\Theta_{\text{Q-Tr},d}$. And if there is a reduction at the current level (the third case above), it re-initializes this state given the state at the level above, according to the distribution $\Theta_{\text{Q-Ex},d}$. The overall effect is that higher-level FSAs are allowed to transition only when lower-level FSAs terminate. An HHMM therefore behaves like a probabilistic implementation of a pushdown automaton (or shift–reduce parser) with a finite stack, where the maximum stack depth is equal to the number of levels in the HHMM hierarchy.

3.2 Tree-Based Transforms

The right-corner transform used in this paper is simply the left-right dual of a left-corner transform (Johnson, 1998). It transforms all right branching sequences in a phrase structure tree into left branching sequences of symbols of the form $A_\eta/A_{\eta\cdot\mu}$, denoting an incomplete instance of an 'active' category A_η lacking an instance of an 'awaited' category $A_{\eta\cdot\mu}$ yet to come.[3] These incomplete constituent categories have the same form and much of the same meaning as non-constituent categories in a Combinatorial Categorial Grammar (Steedman, 2000).

Rewrite rules for the right-corner transform are shown below:[4]

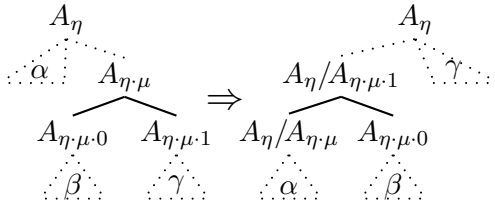
- Beginning case: the top of a right-expanding sequence in an ordinary phrase structure tree is mapped to the bottom of a left-expanding sequence in a right-corner transformed tree:



(10)

  This case of the right-corner transform may be considered a constrained version of CCG type raising.

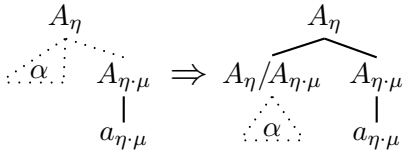
- Middle case: each subsequent branch in a right-expanding sequence of an ordinary phrase structure tree is mapped to a branch in a left-expanding sequence of the transformed tree:



(11)

  This case of the right-corner transform may be considered a constrained version of CCG forward function composition.

- Ending case: the bottom of a right-expanding sequence in an ordinary phrase structure tree is mapped to the top of a left-expanding sequence in a right-corner transformed tree:



(12)

  This case of the right-corner transform may be considered a constrained version of CCG forward function application.

The completeness of the above transform rules can be demonstrated by the fact that they cover all possible subtree configurations (with the exception of

[3] Here η and μ are node addresses in a binary-branching tree, defined as paths of left (0) or right (1) branches from the root.

[4] These rules can be applied recursively from bottom up on a source tree, synchronously associating subtree structures matched to variables α, β, and γ on the left side of each rule with transformed representations of these subtree structures on the right.

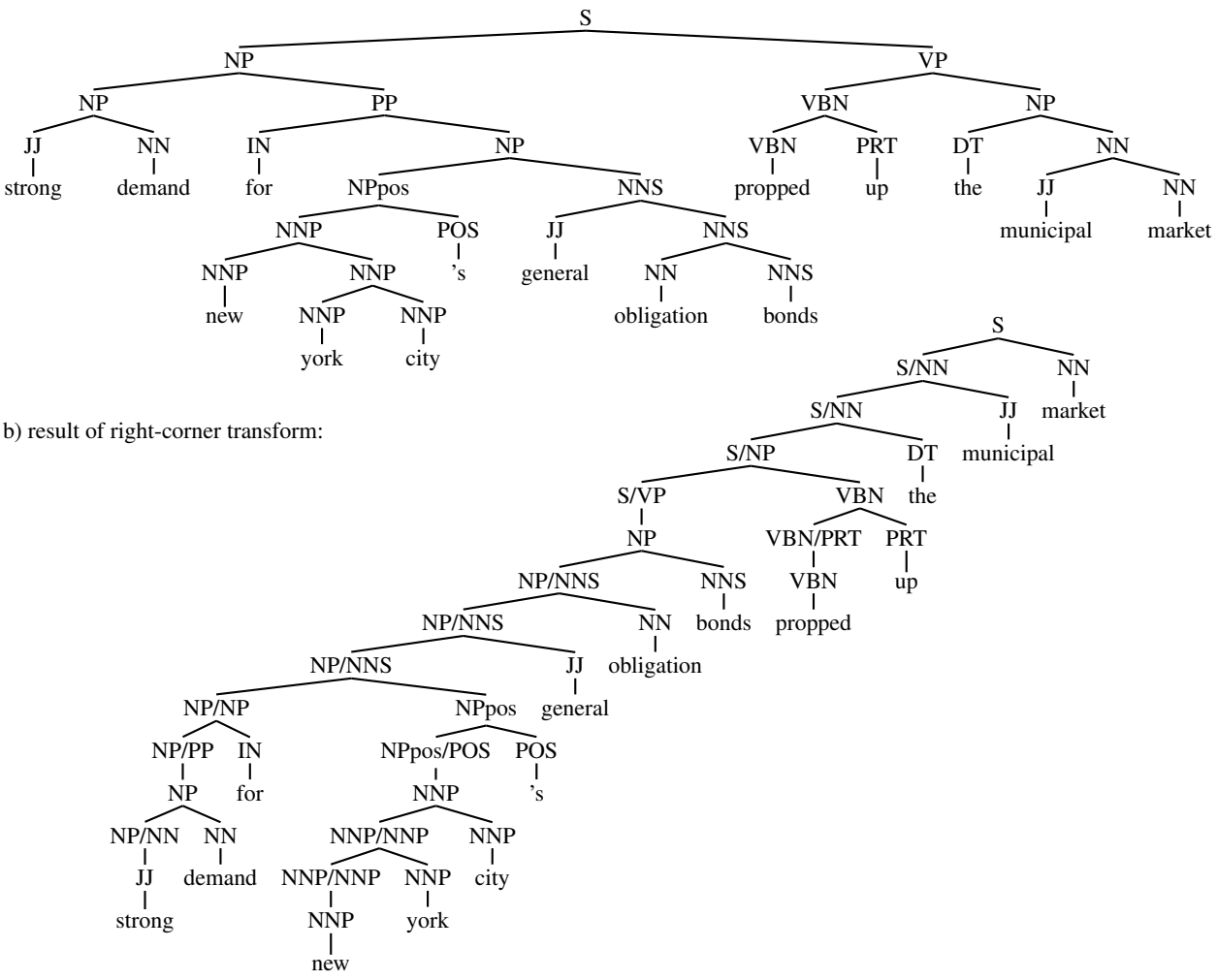


Figure 2: Trees resulting from a) a sample phrase structure tree for the sentence *Strong demand for New York City's general obligations bonds propped up the municipal market,* and b) a right-corner transform of this tree. Sequences of left children are recognized from the bottom up through in-element transitions in a Hierarchic Hidden Markov Model. Right children are recognized by expanding to additional stack elements.

bare terminals, which are simply copied). The soundness of the above transform rules can be demonstrated by the fact that each rule transforms a right-branching subtree into a left-branching subtree labeled with an incomplete constituent.

An example of a right-corner transformed tree is shown in Figure 2(b). An important property of this transform is that it is reversible. Rewrite rules for reversing a right-corner transform are simply the converse of those shown above.

Sequences of left children in the resulting mostly-left-branching trees are recognized from the bottom up, through transitions at the same stack element. Right children, which are much less frequent in the resulting trees, are recognized through cross-element expansions in a bounded-stack recognizer.

4 Model-Based Transforms

In order to compare bounded- and unbounded-stack versions of the same model, the formulation of the right-corner and bounded-stack transforms introduced in this paper does not map trees to trees, but rather maps probability models to probability models. This eliminates complications in comparing models with different numbers of dependent variables — and thus different numbers of free parame-

ters — because the model which ordinarily has more free parameters (the HHMM, in this case) is derived from the model that has fewer (the PCFG). Since they are derived from a simpler underlying model, the additional parameters of the HHMM are not free.

Mapping probability models from one format to another can be thought of as mapping the infinite sets of trees that are defined by these models from one format to another. Probabilities in the transformed model are therefore defined by calculating probabilities for the relevant substructures in the source model, then marginalizing out the values of nodes in these structures that do not appear in the desired expression in the target model.

A bounded-stack HHMM $\Theta_{\text{Q,F}}$ can therefore be derived from an unbounded PCFG Θ_{G} by:

1. organizing the rules in the source PCFG model Θ_{G} into direction-specific versions (distinguishing rules for expanding left and right children, which occur respectively as active and awaited constituent categories in incomplete constituent labels);

2. enforcing depth limits on these direction-specific rules; and

3. mapping these probabilities to HHMM random variable positions at the appropriate depth.

4.1 Direction-specific rules

An inspection of the tree-based right-corner transform rewrites defined in Section 3.2 will show two things. First, constituents occurring as left children in an original tree (with addresses ending in '0') always become active constituents (occurring before the slash, or without a slash) in incomplete constituent categories, and constituents occurring as right children in an original tree (with addresses ending in '1') always become awaited constituents (occurring after the slash); Second, left children expand locally downward in the transformed tree (so each $A_{\eta \cdot 0}/...$ locally dominates $A_{\eta \cdot 0 \cdot 0}/...$), whereas right children expand locally upward (so each $.../A_{\eta \cdot 1}$ is locally dominated by $.../A_{\eta \cdot 1 \cdot 1}$). This means that rules from the original grammar — if distinguished into rules applying only to left and right children (active and awaited constituents) — can still be locally modeled following a right-corner transform. A transformed tree can be generated in this way by expanding downward along the active constituents in a transformed tree, then turning around and expanding upward to fill in the awaited constituents, then turning around again to generate the active constituents at the next depth level, and so on.

4.2 Depth bounds

The locality of the original grammar rules in a right-corner transformed tree allows memory limits on incomplete constituents to be applied directly as depth bounds in the zig-zag generation traversal defined above. These depth limits correspond directly to the depth levels in an HHMM.

In the experiments described in Section 5, direction-specific and depth-specific versions of the original grammar rules are implemented in an ordinary CKY-style dynamic-programming parser, and can therefore simply be cut off at a particular depth level with no renormalization.

But in an HHMM, this will result in *label-bias* effects, in which expanded constituents may have no valid reduction, forcing the system to define distributions for composing constituents that are not compatible. For example, if a constituent is expanded at depth D, and that constituent has no expansions that can be completely processed within depth D, it will not be able to reduce, and will remain incompatible with the incomplete constituent above it. Probabilities for depth-bounded rules must therefore be renormalized to the domain of allowable trees that can be generated within D depth levels, in order to guarantee consistent probabilities for HHMM recognition.

This is done by determining the (depth- and direction-specific) probability $\mathsf{P}_{\Theta_{\text{B-L},d}}(\mathbf{1} \mid A_{\eta \cdot 0})$ or $\mathsf{P}_{\Theta_{\text{B-R},d}}(\mathbf{1} \mid A_{\eta \cdot 1})$ that a tree generated at each depth d and rooted by a left or right child will fit within depth D. These probabilities are then estimated using an approximate inference algorithm, similar to that used in value iteration (Bellman, 1957), which estimates probabilities of infinite trees by exploiting the fact that increasingly longer trees contribute exponentially decreasing probability mass (since each non-terminal expansion must avoid generating a terminal with some probability at each step from the top down), so a sum over probabilities of trees with increasing length k is guaranteed to converge. The algorithm calculates

probabilities of trees with increasing length k until convergence, or to some arbitrary limit K:

$$\mathsf{P}_{\Theta_{\text{B-L},d,k}}(\mathbf{1} \mid A_{\eta\cdot 0}) = \sum_{\substack{A_{\eta\cdot 1\cdot 0},\\ A_{\eta\cdot 1\cdot 1}}} \mathsf{P}_{\Theta_{\text{G}}}(A_{\eta\cdot 0} \rightarrow A_{\eta\cdot 0\cdot 0}\ A_{\eta\cdot 0\cdot 1}) \cdot \mathsf{P}_{\Theta_{\text{B-L},d,k-1}}(\mathbf{1} \mid A_{\eta\cdot 0\cdot 0}) \cdot \mathsf{P}_{\Theta_{\text{B-R},d,k-1}}(\mathbf{1} \mid A_{\eta\cdot 0\cdot 1}) \quad (13)$$

$$\mathsf{P}_{\Theta_{\text{B-R},d,k}}(\mathbf{1} \mid A_{\eta\cdot 1}) = \sum_{\substack{A_{\eta\cdot 1\cdot 0},\\ A_{\eta\cdot 1\cdot 1}}} \mathsf{P}_{\Theta_{\text{G}}}(A_{\eta\cdot 1} \rightarrow A_{\eta\cdot 1\cdot 0}\ A_{\eta\cdot 1\cdot 1}) \cdot \mathsf{P}_{\Theta_{\text{B-L},d+1,k-1}}(\mathbf{1} \mid A_{\eta\cdot 1\cdot 0}) \cdot \mathsf{P}_{\Theta_{\text{B-R},d,k-1}}(\mathbf{1} \mid A_{\eta\cdot 1\cdot 1}) \quad (14)$$

where $\mathsf{P}_{\Theta_{\text{B-L},d,k}}(\mathbf{1} \mid a_\eta) = \mathsf{P}_{\Theta_{\text{B-R},d,k}}(\mathbf{1} \mid a_\eta) = 1$ for terminals a_η, 0 otherwise.

Normalized probability distributions for depth-bounded expansions $\Theta_{\text{G-L},d}$ and $\Theta_{\text{G-R},d}$ can now be calculated using converged $\Theta_{\text{B-L},d}$ and $\Theta_{\text{B-R},d}$ estimates:

$$\mathsf{P}_{\Theta_{\text{G-L},d}}(A_{\eta\cdot 0} \rightarrow A_{\eta\cdot 0\cdot 0}\ A_{\eta\cdot 0\cdot 1}) = \mathsf{P}_{\Theta_{\text{G}}}(A_{\eta\cdot 0} \rightarrow A_{\eta\cdot 0\cdot 0}\ A_{\eta\cdot 0\cdot 1}) \cdot \mathsf{P}_{\Theta_{\text{B-L},d}}(\mathbf{1} \mid A_{\eta\cdot 0\cdot 0}) \cdot \mathsf{P}_{\Theta_{\text{B-R},d}}(\mathbf{1} \mid A_{\eta\cdot 0\cdot 1}) \quad (15)$$

$$\mathsf{P}_{\Theta_{\text{G-R},d}}(A_{\eta\cdot 1} \rightarrow A_{\eta\cdot 1\cdot 0}\ A_{\eta\cdot 1\cdot 1}) = \mathsf{P}_{\Theta_{\text{G}}}(A_{\eta\cdot 1} \rightarrow A_{\eta\cdot 1\cdot 0}\ A_{\eta\cdot 1\cdot 1}) \cdot \mathsf{P}_{\Theta_{\text{B-L},d+1}}(\mathbf{1} \mid A_{\eta\cdot 1\cdot 0}) \cdot \mathsf{P}_{\Theta_{\text{B-R},d}}(\mathbf{1} \mid A_{\eta\cdot 1\cdot 1}) \quad (16)$$

4.3 HHMM probabilities

Converting PCFGs to HHMMs requires the calculation of expected frequencies $\mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_\eta \overset{*}{\rightarrow} A_{\eta\cdot\mu})$ of generating symbols $A_{\eta\cdot\mu}$ in the left-progeny of a nonterminal symbol A_η (in other words, of $A_{\eta\cdot\mu}$ being a left child of A_η, or a left child of a left child of A_η, etc.). This is done by summing over subtrees of increasing length k using the same approximate inference technique described in Section 4.2, which guarantees convergence since each subtree of increasing length contributes exponentially decreasing probability mass to the sum:

$$\mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_\eta \overset{*}{\rightarrow} A_{\eta\cdot\mu}) = \sum_{k=0}^{\infty} \mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_\eta \overset{k}{\rightarrow} A_{\eta\cdot\mu}) \quad (17)$$

where:

$$\mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_\eta \overset{k}{\rightarrow} A_{\eta\cdot 0^k}) = \sum_{\substack{A_{\eta\cdot 0^{k-1}},\\ A_{\eta\cdot 0^{k-1}\cdot 1}}} \mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_\eta \overset{k-1}{\rightarrow} A_{\eta\cdot 0^{k-1}}) \cdot \mathsf{P}_{\Theta_{\text{G-L},d}}(A_{\eta\cdot 0^{k-1}} \rightarrow A_{\eta\cdot 0^k}\ A_{\eta\cdot 0^{k-1}\cdot 1}) \quad (18)$$

and $\mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_\eta \overset{0}{\rightarrow} A'_\eta) = [A_\eta = A'_\eta]$.

A complete HHMM can now be defined using depth-bounded right-corner PCFG probabilities. HHMM probabilities will be defined over syntactic states consisting of incomplete constituent categories $A_\eta/A_{\eta\cdot\mu}$.

Expansions depend on only the incomplete constituent category $../A_\eta$ (for any active category '..') at q_t^{d-1}:

$$\mathsf{P}_{\Theta_{\text{Q-Ex},d}}(a_{\eta\cdot 0\cdot\mu} \mid ../A_\eta) = \frac{\sum_{A_{\eta\cdot 0}, A_{\eta\cdot 1}} \mathsf{P}_{\Theta_{\text{G-R},d-1}}(A_\eta \rightarrow A_{\eta\cdot 0}\ A_{\eta\cdot 1}) \cdot \mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_{\eta\cdot 0} \overset{*}{\rightarrow} a_{\eta\cdot 0\cdot\mu})}{\sum_{A_{\eta\cdot 0}, A_{\eta\cdot 1}, a_{\eta\cdot 0\cdot\mu}} \mathsf{P}_{\Theta_{\text{G-R},d-1}}(A_\eta \rightarrow A_{\eta\cdot 0}\ A_{\eta\cdot 1}) \cdot \mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_{\eta\cdot 0} \overset{*}{\rightarrow} a_{\eta\cdot 0\cdot\mu})} \quad (19)$$

Transitions depend on whether an 'active' or 'awaited' transition was performed at the current level. If an active transition was performed (where $f_t^d = \mathbf{1}$), the transition depends on only the incomplete constituent category $A_{\eta\cdot 0\cdot\mu\cdot 0}/..$ (for any awaited category '..') at q_{t-1}^d, and the incomplete constituent category $../A_\eta$ (for any active category '..') at q_t^{d-1}:

$$\mathsf{P}_{\Theta_{\text{Q-Tr},d}}(A_{\eta\cdot 0\cdot\mu}/A_{\eta\cdot 0\cdot\mu\cdot 1} \mid .., \mathbf{1}, A_{\eta\cdot 0\cdot\mu\cdot 0}/.., ../A_\eta) = \frac{\sum_{A_{\eta\cdot 0}, A_{\eta\cdot 1}} \mathsf{P}_{\Theta_{\text{G-R},d-1}}(A_\eta \rightarrow A_{\eta\cdot 0}\ A_{\eta\cdot 1}) \cdot \mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_{\eta\cdot 0} \overset{*}{\rightarrow} A_{\eta\cdot 0\cdot\mu}) \cdot \mathsf{P}_{\Theta_{\text{G-L},d}}(A_{\eta\cdot 0\cdot\mu} \rightarrow A_{\eta\cdot 0\cdot\mu\cdot 0}\ A_{\eta\cdot 0\cdot\mu\cdot 1})}{\sum_{A_{\eta\cdot 0}, A_{\eta\cdot 1}, A_{\eta\cdot 0\cdot\mu}, A_{\eta\cdot 0\cdot\mu\cdot 1}} \mathsf{P}_{\Theta_{\text{G-R},d-1}}(A_\eta \rightarrow A_{\eta\cdot 0}\ A_{\eta\cdot 1}) \cdot \mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_{\eta\cdot 0} \overset{*}{\rightarrow} A_{\eta\cdot 0\cdot\mu}) \cdot \mathsf{P}_{\Theta_{\text{G-L},d}}(A_{\eta\cdot 0\cdot\mu} \rightarrow A_{\eta\cdot 0\cdot\mu\cdot 0}\ A_{\eta\cdot 0\cdot\mu\cdot 1})} \quad (20)$$

If an awaited transition was performed (where $f_t^d = \mathbf{0}$), the transition depends on only the complete constituent category $A_{\eta\cdot\mu\cdot 0}$ at f_t^{d+1}, and the incomplete constituent category $A_\eta/A_{\eta\cdot\mu}$ at q_{t-1}^d:

$$\mathsf{P}_{\Theta_{\text{Q-Tr},d}}(A_\eta/A_{\eta\cdot\mu\cdot 1} \mid A_{\eta\cdot\mu\cdot 0}, \mathbf{0}, A_\eta/A_{\eta\cdot\mu}, ..) = \frac{\mathsf{P}_{\Theta_{\text{G-R},d}}(A_{\eta\cdot\mu} \rightarrow A_{\eta\cdot\mu\cdot 0}\ A_{\eta\cdot\mu\cdot 1})}{\sum_{A_{\eta\cdot\mu\cdot 1}} \mathsf{P}_{\Theta_{\text{G-R},d}}(A_{\eta\cdot\mu} \rightarrow A_{\eta\cdot\mu\cdot 0}\ A_{\eta\cdot\mu\cdot 1})} \quad (21)$$

Reduce probabilities depend on the complete constituent category at f_t^{d+1}, and the incomplete constituent category $A_{\eta\cdot 0\cdot\mu\cdot 0}/..$ (for any awaited category '..') at q_{t-1}^d, and the incomplete constituent category $../A_\eta$ (for any active category '..') at q_{t-1}^{d-1}. If the complete constituent category at f_t^{d+1} does not match the awaited category of q_{t-1}^d, the probability is $[f_t^d = \mathbf{0}]$. If the complete constituent category at f_t^{d+1} does match the awaited category of q_{t-1}^d:

$$\mathsf{P}_{\Theta_{\text{F-Rd},d}}(\mathbf{1} \mid A_{\eta\cdot 0\cdot\mu}/..,\ ../A_\eta) = \frac{\sum_{A_{\eta\cdot 0}, A_{\eta\cdot 1}} \mathsf{P}_{\Theta_{\text{G-R},d-1}}(A_\eta \rightarrow A_{\eta\cdot 0}\ A_{\eta\cdot 1}) \cdot \begin{pmatrix} \mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_{\eta\cdot 0} \overset{*}{\rightarrow} A_{\eta\cdot 0\cdot\mu}) \\ -\mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_{\eta\cdot 0} \overset{0}{\rightarrow} A_{\eta\cdot 0\cdot\mu}) \end{pmatrix}}{\sum_{A_{\eta\cdot 0}, A_{\eta\cdot 1}} \mathsf{P}_{\Theta_{\text{G-R},d-1}}(A_\eta \rightarrow A_{\eta\cdot 0}\ A_{\eta\cdot 1}) \cdot \mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_{\eta\cdot 0} \overset{*}{\rightarrow} A_{\eta\cdot 0\cdot\mu})} \quad (22)$$

and:

$$\mathsf{P}_{\Theta_{\text{F-Rd},d}}(A_{\eta\cdot 0\cdot\mu} \mid A_{\eta\cdot 0\cdot\mu}/..,\ ../A_\eta) = \frac{\sum_{A_{\eta\cdot 0}, A_{\eta\cdot 1}} \mathsf{P}_{\Theta_{\text{G-R},d-1}}(A_\eta \rightarrow A_{\eta\cdot 0}\ A_{\eta\cdot 1}) \cdot \mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_{\eta\cdot 0} \overset{0}{\rightarrow} A_{\eta\cdot 0\cdot\mu})}{\sum_{A_{\eta\cdot 0}, A_{\eta\cdot 1}} \mathsf{P}_{\Theta_{\text{G-R},d-1}}(A_\eta \rightarrow A_{\eta\cdot 0}\ A_{\eta\cdot 1}) \cdot \mathsf{F}_{\Theta_{\text{G-L}^*,d}}(A_{\eta\cdot 0} \overset{*}{\rightarrow} A_{\eta\cdot 0\cdot\mu})} \quad (23)$$

The correctness of the above distributions can be demonstrated by the fact that all terms other than $\Theta_{\text{G-L},d}$ and $\Theta_{\text{G-R},d}$ probabilities will cancel out in any sequence of transitions between an expansion and a reduction, leaving only those terms that would appear as factors in an ordinary PCFG parse.[5]

5 Results

A PCFG model was extracted from sections 2–21 of the Wall Street Journal Treebank. In order to keep the transform process manageable, punctuation was removed from the corpus, and rules occurring less frequently than 10 times in the corpus were deleted from the PCFG. The right-corner and bounded-stack transforms described in the previous section were then applied to the PCFG. The original and bounded PCFG models were evaluated in a CKY recognizer on sections 22–24 of the Treebank, with results shown in Table 1. Results were significant only for sentences longer than 40 words. On these sentences, the bounded PCFG model achieves about a .15% reduction of error over the original PCFG ($p < .1$ using one-tailed pairwise t-test). This suggests that on long sentences the probability mass wasted due to parsing with an unbounded stack is substantial enough to impact parsing accuracy.

[5] It is important to note, however, that these probabilities are not necessarily incrementally balanced, so this correctness only applies to parsing with an infinite beam.

model (sect 22–24, no punct, len>40)	F
unbounded PCFG	66.03
simple bounded PCFG (D=4)	66.08

Table 1: Results of CKY parsing using simple bounded (to depth D) and unbounded PCFG.

model (sect 23, no punct, len≤40)	F
unbounded PCFG	75.56
approx norm bounded PCFG (D=4)	75.31
HHMM, (D=4, B=2000)	75.06
HHMM, (D=4, B=500)	74.73

Table 2: Parsing results for unbounded CKY, approximate normalized bounded CKY, and HHMMs with wide and narrow beam widths B; no punctuation.

A CKY recognizer was used in both cases in order to avoid introducing errors due to model approximation or beam limits necessary for incremental processing with large grammars. Results for CKY parsing with approximate normalized bounds and HHMM parsing from this bounded form are shown in Figure 2.

6 Conclusion

Previous work has explored bounded-stack parsing using a right-corner transform defined on trees to minimize stack usage. HHMM parsers trained on applications of this tree-based transform of training corpora have shown improvements over ordinary PCFG models, but this may have been attributable to the richer dependencies of the HHMM.

This paper has presented an approximate inference algorithm for transforming entire PCFGs, rather than individual trees, into equivalent right-corner bounded-stack HHMMs. Moreover, a comparison with an untransformed PCFG model suggests that the probability mass wasted due to parsing with an unbounded stack is substantial enough to impact parsing accuracy.

Acknowledgments

This research was supported by NSF CAREER award 0447685 and by NASA under award NNX08AC36A. The views expressed are not necessarily endorsed by the sponsors.

References

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

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

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

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

Edward Gibson. 1991. *A computational theory of human linguistic processing: Memory limitations and processing breakdown*. Ph.D. thesis, Carnegie Mellon.

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

Mark Johnson. 1998. Finite state approximation of constraint-based grammars using left-corner grammar transforms. In *Proceedings of COLING/ACL*, pages 619–623.

Tim Miller and William Schuler. 2008. A syntactic time-series model for parsing fluent and disfluent speech. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08)*.

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

Kevin P. Murphy and Mark A. Paskin. 2001. Linear time inference in hierarchical HMMs. In *Proc. NIPS*, pages 833–840.

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. 2008. Toward a psycholinguistically-motivated model of language. In *Proceedings of COLING*, pages 785–792, Manchester, UK, August.

Mark Steedman. 2000. *The syntactic process*. MIT Press/Bradford Books, Cambridge, MA.