Structured Prediction of Sequences and Trees Using Infinite Contexts

Ehsan Shareghi[1](✉), Gholamreza Haffari[1], Trevor Cohn[2], and Ann Nicholson[1]

[1] Monash University, Melbourne, Australia
{ehsan.shareghi,gholamreza.haffari,ann.nicholson}@monash.edu
[2] University of Melbourne, Melbourne, Australia
t.cohn@unimelb.edu.au

Abstract. Linguistic structures exhibit a rich array of global phenomena, however commonly used Markov models are unable to adequately describe these phenomena due to their strong locality assumptions. We propose a novel hierarchical model for structured prediction over sequences and trees which exploits global context by conditioning each generation decision on an *unbounded* context of prior decisions. This builds on the success of Markov models but without imposing a fixed bound in order to better represent global phenomena. To facilitate learning of this large and unbounded model, we use a hierarchical Pitman-Yor process prior which provides a recursive form of smoothing. We propose prediction algorithms based on A* and Markov Chain Monte Carlo sampling. Empirical results demonstrate the potential of our model compared to baseline finite-context Markov models on three tasks: morphological parsing, syntactic parsing and part-of-speech tagging.

Keywords: Structured prediction · Infinite markov model · Chinese restaurant process

1 Introduction

Markov models are widespread popular techniques for modelling the underlying structure of natural language, e.g., as sequences and trees. However local Markov assumptions often fail to capture phenomena outside the local Markov context, i.e., when the data generation process exhibits long range dependencies. A prime example is language modelling where only short range dependencies are captured by finite-order (i.e. n-gram) Markov models. However, it has been shown that going beyond finite order in a Markov model improves language modelling because natural language embodies a large array of long range depepndencies [Wood et al., 2009]. While *infinite* order Markov models have been extensively explored for language modelling [Gasthaus and Teh, 2010; Wood et al., 2011], this has not yet been done for structure prediction.

In this paper, we propose an infinite-order Markov model for predicting latent structures, namely tag sequences and trees. We show that this expressive model can be applied to various structure prediction tasks in NLP, such as syntactic and morphological parsing and part-of-speech tagging. We propose effective

A. Appice et al. (Eds.): ECML PKDD 2015, Part II, LNAI 9285, pp. 373–389, 2015.
DOI: 10.1007/978-3-319-23525-7_23

algorithms to tackle significant learning and inference challenges posed by the infinite Markov model.

More specifically, we propose an unbounded-depth, hierarchical, Bayesian non-parametric model for the generation of linguistic utterances and their corresponding structure (e.g., the sequence of POS tags or syntax trees). Our model conditions each decision in a tree generating process on an *unbounded* context consisting of the vertical chain of their ancestors, in the same way that infinite sequence models (e.g., ∞-gram language models) condition on an unbounded window of linear context [Mochihashi and Sumita, 2007; Wood et al., 2009].

Learning in this model is particularly challenging due to the large space of contexts and corresponding data sparsity. For this reason predictive distributions associated with contexts are smoothed using distribtions for successively smaller contexts via a hierarchical Pitman-Yor process, organised as a suffix trie. The infinite context makes it impossible to directly apply dynamic programing for structure prediction. We present two inference algorithms based on A* and Markov Chain Monte Carlo (MCMC) for predicting the best structure for a given input utterance.

The experiments on part-of-speech (POS) tagging show that our generative model obtains similar performance to the state-of-the-art Stanford POS tagger [Toutanova and Manning, 2000] for English and Swedish. For Danish, our model outperforms the Stanford tagger, which is impressive given the Stanford parser uses many more complex features and a discriminative training objective. Our experiments on morphological parsing and syntactic parsing show that our unbounded-context tree model adapts itself to the data to effectively capture sufficient context to outperform the PCFG baseline.

2 Background and Related Work

The parse tree of an utterance can be generated by combining a set of rules from a grammar, such as a context free grammar (CFG). A CFG is a 4-tuple $\mathcal{G} = (\mathcal{T}, \mathcal{N}, S, \mathcal{R})$, where $\mathcal{T}$ is a set of terminal symbols, $\mathcal{N}$ is a set of non-terminal symbols, $S \in \mathcal{N}$ is the distinguished root non-terminal and $\mathcal{R}$ is a set of productions (aka rewriting rules). A PCFG assigns a probability to each grammar rule, where $\sum_{B,C} P(A \rightarrow B\ C|A) = 1$. The grammar rules are often in Chomsky Normal Form (CNF), taking either the form $A \rightarrow B\ C$ or $A \rightarrow a$ where A, B, C are nonterminals, and a is a terminal.

Syntactic parsing is the task of predicting the parse tree of a given sentence. In syntactic parsing, the nonterminals of the underlying grammar are syntactic catergories, e.g. the input sentence (S), noun phrase (NP) and verb phrase (VP); the terminals are words. Morphological parsing is the task of breaking down an unsegmented input into words and their morphological structure. In this task, the grammar terminals are morphemes (smallest meaningful units of a language), and nonterminals represent the input Sequence, Word, prefix (P), etc. Tag sequences can also be represented as a tree structure, without loss of generality, in which rules take the form $A \rightarrow B\ a$ or $A \rightarrow a$ where A, B are POS tags, and a is a word. This unified view to syntactic parsing, morphological parsing, and POS tagging will allow us to apply our model and inference algorithms to these problems with only minor refinements (see Figure 1).

In PCFG, a tree is generated by starting with the root symbol and rewriting (substituting) it with a grammar rule, then continuing to rewrite frontier non-terminals with grammar rules until there are no remaining frontier non-terminals. When making the decision about the next rule to expand a frontier non-terminal, the only conditioning context used from the partially generated tree is the frontier non-terminal itself, i.e., the rewrite rule is assumed independent from the remainder of the tree given the frontier non-terminal. Our model relaxes this strong independence assumptions by considering unbounded vertical history when making the next inference decision. This takes into account a wider context when making the next parsing decision.

Perhaps the most relevant work is on unbounded history language models [Mochihashi and Sumita, 2007; Wood et al., 2009]. A prime work is Sequence Memoizer [Wood et al., 2011] which conditions the generation of the next word on an unbounded history of previously generated words. We build on these techniques to develop rich infinite-context models for structured prediction, leading to additional complexity and challenges.

For syntactic parsing, several infinite extensions of probabilistic context free grammars (PCFGs) have been proposed [Finkel et al., 2007; Liang et al., 2007]. These approaches achieve infinite grammars by allowing an unbounded set of non-terminals (hence grammar rules), but still make use of a bounded history when expanding each non-terminal. An alternative method allows for infinite grammars by considering segmentation of trees into arbitrarily large tree fragments, although only a limited history is used to conjoin fragments [Cohn et al., 2010; Johnson et al., 2006]. Our work achieves infinite grammars by growing the *vertical* history needed to make the next parsing decision, as opposed to growing the number of rules, non-terminals or states *horizontally*, as done in prior work.

Earlier work in syntactic parsing has also looked into growing both the history vertically and the rules horizontally, in a *bounded* setting. [Johnson, 1998] has increased the history for the parsing task by parent-annotation, i.e., annotating each non-terminal in the training parse trees by its parent, and then reading off the grammar rules from the resulting trees. [Klein and Manning, 2003] have considered vertical and horizontal markovization while using the head words' part-of-speech tag, and showed that increasing the size of the vertical contexts consistently improves the parsing performance. [Petrov et al., 2006], [Petrov and Klein, 2007] and [Matsuzaki et al., 2005] have treated non-terminal annotations as latent variables and estimated them from the data.

Likewise, finite-state hidden Markov models (HMMs) have been extended *horizontally* to have countably infinite number of states [Beal et al., 2001]. Previous works on applying Markov models to part-of-speech tagging either considered finite-order Markov models [Chen, 2000], or finite-order HMM [Thede and Harper, 1999]. We differ from these works by conditioning *both* the emissions and transitions on their *full* contexts.

3 The Model

Our model relaxes strong local Markov assumptions in PCFG to enable capturing phenomena outside of the local Markov context. The model conditions the generation of a rule in a tree on its unbounded vertical history, i.e., its ancestors on the path towards the root of the tree (see Figure 1). Thus the probability of a tree T is

$$P(T) = \prod_{(\mathbf{u},r)\in T} G_{[\mathbf{u}]}(r) \tag{1}$$

where r denotes the rule and $\mathbf{u}$ its history, and $G_{[\mathbf{u}]}(.)$ is the probability of the next inference decision (i.e., grammar rule) conditioned on the context $\mathbf{u}$. In other words, a tree T can be represented as a sequence of context-rule events $\{(\mathbf{u}, r) \in T\}$.

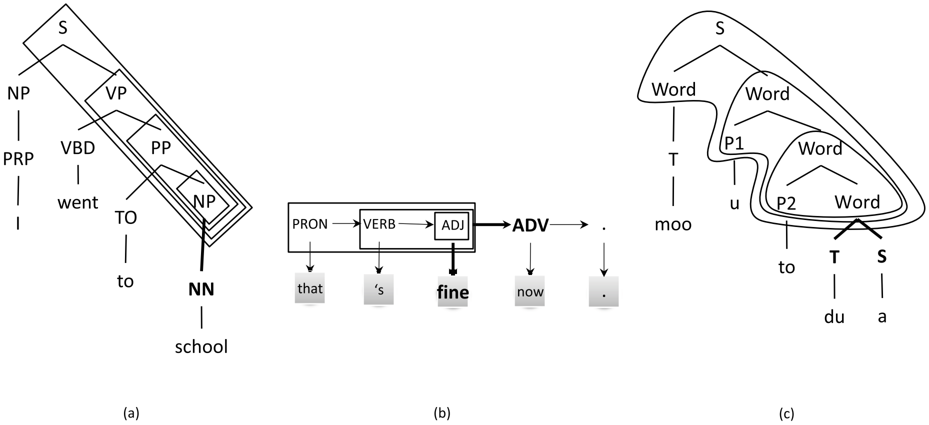


Fig. 1. Examples of infinite-order conditioning and smoothing mechanism. The bold symbols (**NN, ADV, fine, T, S**) are the part of the structure being generated, and the boxes correspond to the conditioning context. (a) Syntactic Parsing, (b) Infinite-order HMM for POS tagging, (c) Morphological Parsing.

When learning such a model from data, a vector of predictive probabilities for the next rule $G_{[\mathbf{u}]}(.)$ given each possible vertical context $\mathbf{u} \in \mathcal{U}$ must be learned, where depending on the problem $\mathcal{U}$ can denote the set of spines of non-terminals $\mathcal{N}^*$ (as in Fig. 1(a),(b)) or chains of rules $\mathcal{R}^*$(as in Fig. 1(c)). As the context size increases, the number of events observed for such long contexts in the training data drastically decreases which makes parameter estimation challenging, particularly when generalising to unseen contexts. Assuming our unbounded-depth model, we need suitable *smoothing* techniques to estimate conditional rule probabilities for large (and possibly infinite depth) contexts. We achieve smoothing by placing a hierarchical Bayesian prior over the set of probability distributions $\{G_{[\mathbf{u}]}\}_{u\in\mathcal{U}}$. We smooth $G_{[\mathbf{u}]}$ with a distribution conditioned on a shorter context $G_{[\pi(\mathbf{u})]}$, where $\pi(\mathbf{u})$ is the suffix of $\mathbf{u}$ containing all but the earliest event. This ties parameters of longer histories to their shorter suffixes in a hierarchical manner, and leads to sharing statistical strengths to overcome sparsity issues. Figure 1 shows our infinite-order Markov model and the smoothing mechanism described here.

More specifically, we assume that a distribution with the full history $G_{[u]}$ is related to a distribution with the most recent history $G_{[\pi(u)]}$ through the Pitman-Yor process PYP [Wood et al., 2011]:

$$G_{[\varepsilon]} \mid d_{[\varepsilon]}, c_{[\varepsilon]}, H \sim PYP(d_0, c_0, H) \tag{2}$$

$$G_{[\mathbf{u}]} \mid d_{|\mathbf{u}|}, c_{|\mathbf{u}|}, G_{[\pi(\mathbf{u})]} \sim PYP(d_{|\mathbf{u}|}, c_{|\mathbf{u}|}, G_{[\pi(\mathbf{u})]}) \tag{3}$$

where H denotes the base (e.g. uniform) distribution, and ε denotes the empty context. The Pitman-Yor process $PYP(d, c, H)$ is a distribution over distributions, where d is the discount parameter, c is the concentration parameter, and H is the base distribution. Note that $G_{[u]}$ depends on $G_{[\pi(u)]}$ which itself depends on $G_{[\pi(\pi(u))]}$, etc. This leads to a hierarchical Pitman-Yor process prior where context-dependent distributions are *hidden*. The formulation of the hierarchical PYP over different length contexts is illustrated in Figure 2.

Figure 3 demonstrates the property of PYP and how its behavior depends on discount d, and concentration c parameters. Note that the PYP allows a good fit to data distribution compared to the Dirichlet Process ($d = 0$; as used in prior work) which cannot adequately represent the long tail of events.

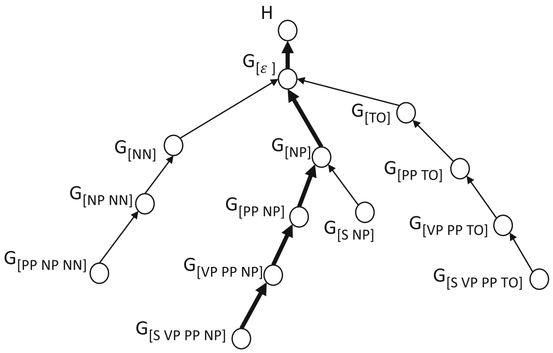
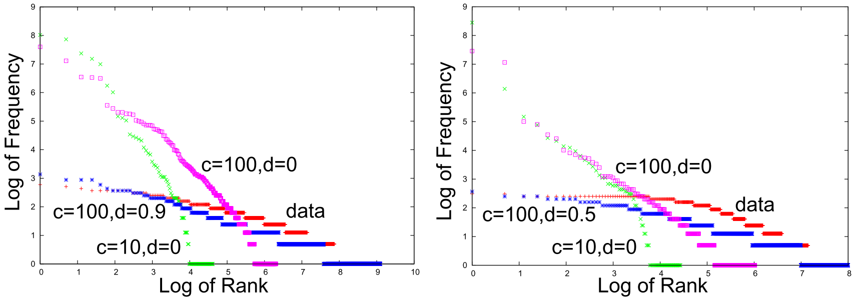


Fig. 2. Part of the smoothing mechanism corresponding to Figure 1(a). Each node represents a distribution G labeled with a context, and the directed edges demonstrate the direction of smoothing. The path in bold corresponds to the smoothing for the *rule* $NP \rightarrow NN$.



(a) $\mathbf{u}$: S NP

(b) $\mathbf{u}$: VERB

Fig. 3. log-log plot of rule frequency vs rank, illustrated for (a) syntactic parsing and (b) POS tagging. Besides the data distribution, we also show samples from three PYP distributions with different hyperparameter values, c, d.

4 Learning

Given a training tree-bank, i.e., a collection of utterances and their trees, we are interested in the posterior distribution over $\{G_{[\mathbf{u}]}\}_{\mathbf{u}\in\mathcal{U}}$. We make use of the approach developed in [Wood et al., 2011] for learning such suffix-based graphical models when learning infinite-depth language models. It makes use of Chinese Restaurant Process (CRP) representation of the Pitman-Yor process in order to marginalize out distributions $G_{[\mathbf{u}]}$ [Teh, 2006] and learn the predictive probabilities $P(r|\mathbf{u})$.

Under the CRP representation each context corresponds to a restaurant. As a new $(\mathbf{u}, r)$ is observed in the training data, a *customer* is entered to the restaurant, i.e., the trie node corresponding to $\mathbf{u}$. Whenever a customer enters a restaurant, it should be decided whether to seat him on an existing table serving the *dish* r, or to seat him on a new table and sending a proxy customer to the parent node in the trie to order r (i.e., based on $(\pi(\mathbf{u}), r)$). Fixing a seating arrangement $\mathbf{S}$ and PYP parameters $\boldsymbol{\theta}$ for all restaurants (i.e., the collection of concentration and discount parameters), the predictive probability of a rule based on our infinite-context rule model is:

$$P(r|\epsilon, \mathbf{S}, \boldsymbol{\theta}) = H(r) \tag{4}$$

$$P(r|\mathbf{u}, \mathbf{S}, \boldsymbol{\theta}) = \frac{n^{\mathbf{u}}_{r\cdot} - d_{|\mathbf{u}|} t^{\mathbf{u}}_{r}}{n^{|\mathbf{u}|}_{\cdot\cdot} + c_{|\mathbf{u}|}} + \frac{c_{|\mathbf{u}|} + d_{|\mathbf{u}|} t^{\mathbf{u}}_{\cdot}}{n^{\mathbf{u}}_{\cdot\cdot} + c_{|\mathbf{u}|}} P(r|\pi(\mathbf{u}), \mathbf{S}, \boldsymbol{\theta}) \tag{5}$$

where $d_{|\mathbf{u}|}$ and $c_{|\mathbf{u}|}$ are the discount and concentration parameters, $n^{\mathbf{u}}_{rk}$ is the number of customers at table k served the dish r in the restaurant $\mathbf{u}$ (accordingly $n^{\mathbf{u}}_{r\cdot}$ is the number of customers served the dish r and $n^{\mathbf{u}}_{\cdot\cdot}$ is the number of customers), and $t^{\mathbf{u}}_{r}$ is the number of tables serving dish r in the restaurant $\mathbf{u}$ (accordingly $t^{\mathbf{u}}_{\cdot}$ is the number of tables).

The seating arrangements (the state of all restaurants including their tables and customers sitting on each table) are hidden, so they need to be marginalized out:

$$P(r|\mathbf{u}, \mathcal{D}) = \int P(r|\mathbf{u}, \mathbf{S}, \boldsymbol{\theta}) P(\mathbf{S}, \boldsymbol{\theta}|\mathcal{D}) d(\mathbf{S}, \boldsymbol{\theta}) \tag{6}$$

where $\mathcal{D}$ is the training tree-bank. We approximate this integral by the so called "minimal assumption seating arrangement" and the MAP parameter setting $\boldsymbol{\theta}$ which maximizes the corresponding data posterior. Based on the minimal assumption, a new table is created only when there is no table serving the desired dish in a restaurant $\mathbf{u}$. That is, a proxy customer is created and sent to the parent node in the trie $\pi(\mathbf{u})$ for each unique dish type (sequence of events). This approximation is related to the well-known interpolated Kneser-Ney smoothing [Chen and Goodman, 1996], when applied to hierarchical Pitman-Yor process language models [Teh, 2006].

The parameter $\boldsymbol{\theta}$ is learned by maximising the posterior, given the seating arrangement corresponding to the minimal assumption. We put the following prior distributions over the parameters: $d_m \sim \text{Beta}(a_m, b_m)$ and $c_m \sim \text{Gamma}(\alpha_m, \beta_m)$. The posterior is the prior multiplied by the following likelihood term:

$$\prod_{r} H(r)^{n^0_{r\cdot}} \prod_{\mathbf{u}} \frac{[c_{|\mathbf{u}|}]^{t^{\mathbf{u}}_{\cdot}}_{d_{|\mathbf{u}|}}}{[c_{|\mathbf{u}|}]^{n^{\mathbf{u}}_{\cdot\cdot}}_{1}} \prod_{r} \prod_{k=1}^{t^{\mathbf{u}}_{\cdot}} [1 - d_{|\mathbf{u}|}]_1^{(n^{\mathbf{u}}_{rk} - 1)} \tag{7}$$

where $[a]^c_b$ denotes the generalised factorial function.[1] We maximize the posterior with the constraints $c_m \geq 0$ and $d_m \in [0, 1)$ using the L-BFGS-B optimisation method [Zhu et al., 1997], leading to the optimised discount and concentration for each context size.

5 Prediction

In this section, we propose algorithms for the challenging problem of predicting the highest scoring tree. The key ideas are to compactly represent the *space* of all possible trees for a given utterance, and then *search* for the best tree in this space in a *top-down* manner. By traversing the hyper-graph top-down, the search algorithms have access to the full history of grammar rules.

In the test time, we need to predict the tree structure of a given utterance $\mathbf{w}$ by maximizing the tree score:

$$\arg\max_{T} P(T|\mathcal{D}, \mathbf{w}) = \arg\max_{T} \prod_{(\mathbf{u},r) \in T} P(r|\mathbf{u}, \mathcal{D}) \tag{8}$$

The unbounded context allowed by our model makes it infeasible to apply dynamic programming, e.g. CYK [Cocke and Schwartz, 1970], for finding the

[1] $[a]^0_b = [a]^{-1}_b = 1$ and $[a]^b_c = \prod_{i=0}^{c-1}(a + ib)$.

highest scoring tree. CYK is a *bottom-up* algorithm which requires storing in a dynamic programming table the score of each utterance's sub-span conditioned on all possible contexts. Even truncating the context size to bound this term may be insufficient to allow CYK for prediction, due to the unreasonable computational complexity.

The space of all possible trees for a given utterance can be compactly represented as a *hyper-graph* [Klein and Manning, 2001]. Each hyper-graph node is labelled with a non-terminal and a sub-span of the utterance. There exists a hyper-edge from the nodes $B[i, j]$ and $C[j + 1, k]$ to the node $A[i, k]$ if the rule $A \rightarrow B\ C$ belongs to the grammar (Figure 4). Starting from the top node $S[0, N]$, our prediction algorithms search for the highest scoring tree sub-graph that covers all of the utterance terminals in the hyper-graph. Our top-down prediction algorithms have access to the full history needed by our model when deciding about the next hyper-edge to be added to the partial tree.

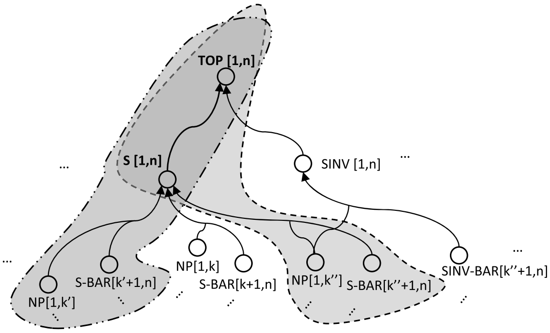


Fig. 4. Hyper-graph representation of the search space for a syntactic parsing example. The gray areas are examples of two partial hypotheses in A* priority queue.

5.1 A* Search

This algorithm incrementally expands frontier nodes of the best partial tree until a complete tree is constructed. In the expansion step, all possible rules for expanding all frontier non-terminals are considered and the resulting partial trees are inserted into a priority queue (see Figure 4), sorted based on the following score:

$$Score(T^+) = \log P(T) + \log G_{\mathbf{u}}(A \rightarrow B\ C) + h(T^+, A \rightarrow B\ C, i, k, j | G') \quad (9)$$

where T^+ is a partial tree *after* expanding a frontier non-terminal, $P(T)$ is the probability of the current partial tree, $G_{\mathbf{u}}(A \rightarrow B\ C)$ is the probability of expanding a non-terminal via a rule $A \rightarrow B\ C$ in the full context $\mathbf{u}$, and h is the heuristic function (i.e., the estimate of the score for the best tree completing T^+). We use various heuristic functions when expanding a node $A[i, j]$ in the hypergraph via a hyperedge with tails $B[i, k]$ and $C[k + 1, j]$:

- **Full Frontier:** which estimates the completion cost by

$$h(T^+, A \rightarrow B\ C, i, k, j|G') = \sum_{(A', i', j') \in \mathrm{Fr}(T^+)} \log P(A', i', j'|G') \qquad (10)$$

  where $\mathrm{Fr}(T^+)$ is the set of frontier nodes of the partial tree, and G' is a *simplified grammar* admitting dynamic programming. Here we choose the PCFG used the base measure H in the root of the PYP hierarchy. Accordingly the $\log P$ terms can be computed cheaply using the PCFG inside probabilities.
- **Local Frontier:** which only considers the completion of the following frontier nodes, and uses the completion cost of the sub-span using the selected rule:

$$h(T^+, A \rightarrow B\ C, i, k, j|G') = \log P(B, i, k|G') + \log P(C, k+1, j|G') \quad (11)$$

The above heuristics functions are not admissible, hence the A* algorithm is not guaranteed to find the optimal tree. However the PCFG provides reasonable estimates of the completion costs, and accordingly with a sufficiently wide beam, search error is likely to be low.

5.2 MCMC Sampling

We make use of Metropolis-Hastings (MH) algorithm, which is a Markov chain Monte Carlo (MCMC) method, for obtaining a sequence of random trees. We then combine these trees to construct the predicted tree.

We use a *PCFG* as our *proposal* distribution Q and draw samples from it. Each sampled tree is then accepted/rejected using the following acceptance rate:

$$\alpha(T, T') = \min \left\{ 1, \frac{P(T')Q(T)}{P(T)Q(T')} \right\} \qquad (12)$$

where T' is the sampled tree, T is the current tree, $P(T')$ is the probability of the proposed tree under our model, and $Q(T')$ is its probability under the proposal PCFG. Under some conditions, i.e., detailed balance and ergodicity, it is guarantheed that the stationary distribution of the underlying Markov chain (defined by the MH sampling) is the distribution that our model induces over the space of trees P. For each utterence, we sample a fresh tree for the whole utterance from a PCFG using the approach of [Johnson et al., 2007], which works by first computing the inside lattice under the proposal model (computed once and reused), followed by top-down sampling to recover a tree. Finally the proposed tree is scored using the MH test, according to which the tree is randomly accepted as the next sample or else rejected in which case the previous sample is retained.

Once the sampling is finished, we need to choose a tree based on statistics of the sampled collection of trees. One approach is to select the most frequently sampled tree, however this does not work effectively in such large search spaces because of high sampling variance. Note that local Gibbs samplers might be able to address this problem, at least partly, through resampling subtrees instead of full tree sampling (as done here). Local changes would allow for more rapid mixing from trees with some high and low scoring subtrees to trees with uniformly high scoring sub-structures. We leave local sampling for future work, noting that the obvious local operation of resampling complete sub-trees or local tree fragments would compromise detailed balance, and thus not constitute a valid MCMC sampler [Levenberg et al., 2012].

To address this problem, we use a Minimum Bayes Risk (MBR) decoding method to predict the best tree [Goodman, 1996] as follows: For each pair of a nonterminal-span, we record the count in the collection of sampled trees. Then using the Viterbi algorithm, we select the tree from the hypergraph for which the sum of the induced pairs of nonterminal-span is maximized. Roughly speaking, this allows to make local corrections that result in higher accuracy compared to the best sampled trees.

6 Experiments

In order to evaluate the proposed model and prediction algorithms, we performed two sets of experiments on tasks with different structural complexity. The statistics of the tasks and datasets are provided in Table 1.

6.1 Morphological Parsing

We consider the problem of morphological parsing of unsegmented inputs, i.e. seeking to model words and their morphological structure in the input stream. A morphological structure of a word breaks it into its building blocks: Prefixes, Stem, and Suffixes. For example, for the word *"antidisestablishmentarianism"*, the terms "anti", "dis" are the prefixes,"establish" is the stem, and "ment", "arian", and "ism" are the suffixes.

For this experiment, we model the Sesotho language, a Bantu language which combines rich productive agglutinative morphology with relatively simple phonology. We use the dataset from [Johnson, 2008], which comprises utterances marked with word boundaries. In contrast to Johnson's approach, our method is supervised, and consequently we require treebanked input for training and

Table 1. Statistics for PTB syntactic Parsing and part-of-speech tagging, showing the number of training and test sentences, average sentence length in words and number of grammar rules. For morph the numbers are averaged over the 10 folds.

Task	Train	Test	Len	Rules
morph	36479	4000	5	3080
parse	33180	2416	24	31920
pos EN	38219	5462	24	29499
pos DN	3638	1000	20	5269
pos SW	10653	389	18	9739

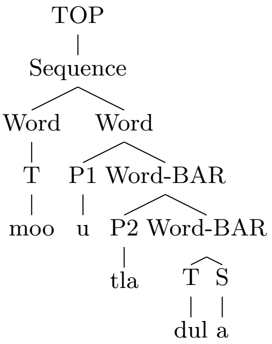


Fig. 5. Binarized morphological tree for the Sethoso sequence *"moo utladula"*.

evaluation. To form a proxy 'gold-standard', we augmented the input to include morphological trees with prefix (P), suffix (S) and stem (T) structure inferred automatically from segmented utterances using Johnson's Adaptor Grammar with his *word-smorph* grammar. In this grammar a word consists of a stem with an optional suffix, and zero to three prefixes: $Word \rightarrow (P1(P2(P3)))\ T\ (S)$, where $P1, P2, P3$ are prefixes, and T, and S are stem and suffix. An example input is shown in Figure 5. We right-binarized the trees and replaced segments with *count* ≤ 2 with two categories of $OUT\text{-}V$ and $OUT\text{-}C$ depending on their initial character being a vowel or consonant. We applied 10-fold cross validation, and the predicted trees were evaluated using EVALB evaluation package.

As reported in Table 2, the best result is achieved by A* search with local frontier heuristic. It might seem surprising that considering the full frontier heuristic results in lower performance. We speculate that this is because the PCFG over estimates the completion cost, due to its reduced conditioning context which leads to higher entropy distributions and lower probability estimates. The reduced effect of the heuristic in the local method moderates this issue. The MCMC sampler obtains similar results to the baseline PCFG.

In the morphological parsing task, the grammar has 3080 rules, and the average sentence length is 5 words. This leads to a reasonably-small search space, with the net effect that A* search (with beam size 200) is an effective parsing strategy. The small grammar size of this task has allowed us to use grammar rules as fine-grained conditioning contexts. In the remaining tasks of syntactic parsing and POS tagging, we will condition only on the spine. This is due to the intractable magnitude of the spaces generated by infinite order rule conditioning, which are problematic for MCMC sampling and A* search based on our preliminary experiments.

6.2 Syntactic Parsing

For syntactic parsing, we use the Penn. treebank (PTB) dataset [Marcus et al., 1993]. We used the standard data splits for training and testing (train sec 2-21;

Table 2. Morphological parsing results, showing 10-fold cross validation evaluation for unlabelled F-Measure (F1) and exact bracketing match (ACC). MCMC results are averaged over 10 runs.

Parser (Morphological)	F1	ACC
A* Search (Local Frontier)	95.99	89.77
A* Search (Full Frontier)	93.08	85.04
MCMC	91.33	78.86
PCFG CYK	91.27	79.39

validation sec 22; test sec 23). We followed [Petrov et al., 2006] preprocessing steps by right-binarizing the trees and replacing words with *count* ≤ 1 in the training sample with generic *unknown* word markers representing the tokens' lexical features and position. The results reported in Table 3 are produced by EVALB.

The results in Table 3 demonstrate the superiority of our model compared to the baseline PCFG. We note that the A* parser becomes less effective (even with a large beam size) for this task, which we attribute to the large search arising for the large grammar and long sentences. Our best results are achieved by MCMC, demonstrating the effectiveness of MCMC in large search spaces.

An interesting observation is how our results compare with those achieved by bounded vertical and horizontal Markovization reported in [Klein and Manning, 2003]. Our binarization corresponds to one of their simpler settings for horizontal markovization, namely $h = 0$ in their terminology, and note also that we ignore the head information which is used in their models. Despite this we still manage to equal their results obtained using vertical context of size 3 ($v = 3$), with 76.7 F1 score. Their best result, $F_1 = 79.74$, was achieved with $h \leq 2$, $v = 3$ (and tags for head words). We believe that our model would outperform theirs if we consider greater horizontal markovization and incorporate head word information. To facilitate a fair comparison with vertical markovization, we experimented with limiting the size of the vertical contexts to 2, 3 or 4 within our model. Using MCMC parsing we found that performance consistently improved as the size of the context was increased, scoring 68.1, 71.1, 75.0 F-measure respectively. This is below 76.7 F-measure of our unbounded-context model which adapts itself to data to effectively capture the right context.

Table 3. Syntactic parsing results for the Penn. treebank, showing labelled F-Measure (F1) and exact bracketing match (ACC).

	all		≤ 40	
Syntactic Parser	F1	ACC	F1	ACC
A* (Local Frontier)	75.33	16.12	76.21	16.85
A* (Full Frontier)	72.27	13.14	72.34	13.57
MCMC	76.74	18.23	78.21	18.99
PCFG CYK	58.91	4.11	60.25	4.42

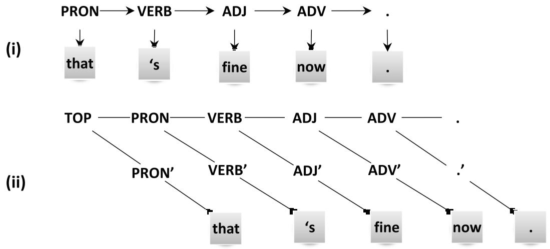


Fig. 6. The analogy between HMM (i) and our representation (ii) for the part-of-speech tags of the sentence *"that's fine now."*

The run-time of our parser under MCMC (with $30k$ samples) is $0.29 \times |S|$ secs, and under A*(Local) is $0.13 \times |S|$ secs, where $|S|$ is the length of the sentence. With a smaller number of samples the parsing time reduces linearly and the predictive accuracy only suffers slightly; for instance with 5k samples the F1 measure (all) falls by 0.8.

Overall our approach significantly outperforms the baseline PCFG, although note these results are well below the current state-of-the-art in parsing, which typically makes use of discriminative training with much richer features. We speculate that future enhancements could close the gap between our results and that of modern parsers, while offering the potential benefits of our generative model which allows further incorporation of different types of contexts (e.g., head words and n-gram lexical context).

6.3 Part-of-Speech Tagging

The part of speech (POS) corpora have been extracted from PTB (sections 0-18 for training and 22-24 for test) for English, and NAACL-HLT 2012 Shared task on Grammar Induction[2] for Danish and Swedish [Gelling et al., 2012]. We convert the sequence of part-of-speech tags for each sentence into a tree structure analogous to a Hidden Markov Model (HMM). For each POS tag we introduce a twin (e.g., ADJ' for ADJ) in order to encode HMM-like transition and emission probabilities in the grammar. As shown in Figure 6, this representation guarantees that all the rules in the structures are either in the form of $t_i \rightarrow t_j\ t'_j$ (transition) or $t' \rightarrow \texttt{word}$ (emission).

The tagging results are reported in Table 4, including comparison with the baseline PCFG ($\equiv$ HMM) and the state-of-the-art Stanford POS Tagger [Toutanova and Manning, 2000], which we trained and tested on these datasets. As illustrated in Table 4, our model consistently improves the PCFG baseline. While for Danish we outperform the state-of-the-art tagger, the results for English and Swedish we are a little behind the Stanford Tagger. This is a promising result since our model is only based on the rules and their contexts,

[2] http://wiki.cs.ox.ac.uk/InducingLinguisticStructure/SharedTask

Table 4. TL stands for Token-Level Accuracy, SL stands for Sentence-Level Accuracy. MCMC results are the average of 10 runs.

	English		**Danish**		**Swedish**	
POS Tagger	TL	SL	TL	SL	TL	SL
A*(Local Frontier)	95.50	54.11	89.85	35.10	87.04	32.13
A*(Full Frontier)	95.27	53.88	88.57	32.6	85.62	28.53
MCMC	96.04	54.25	95.55	72.93	89.97	34.45
PCFG CYK	94.69	47.22	89.04	31.7	89.76	33.93
Stanford Tagger	97.24	56.34	93.66	51.30	91.28	37.02

Table 5. (a) Percentage of the matched spines over the top-1000 frequent spines for each spine length in the trees predicted by our unbounded-context model ($v = \infty$) and the baseline limited-context model ($v = 2$). (b) The top-5 frequent contexts for NP, VP, DT, and JJ in the trees predicted by our model; the ones marked with (*) exist in the top-5 contexts in the gold standard trees as well.

	$v = \infty$				$v = 2$			
	parse	**POS**			**parse**	**POS**		
size	WSJ	EN	DN	SW	WSJ	EN	DN	SW
2	100	96	100	97	100	96	100	97
3	75	100	100	100	75	100	100	100
4	72	69	72	68	70	68	72	68
5	68	57	58	57	63	57	58	57
6	62	56	51	53	60	56	51	53
7	59	52	55	37	59	50	55	38
8	58	42	45	29	51	41	44	29
9	60	68	61	37	49	60	31	37
10	68	75	67	35	51	69	25	34

(a)

NP	**VP**
S*	S*
SINV*	S VP SBAR S*
S PP*	SINV*
S VP PP	S SBAR S
S VP*	S PRN
DT	**JJ**
S NP *	S NP*
S PP NP*	S PP NP*
S VP SBAR S NP*	S VP ADJP*
S VP PP NP*	SINV NP
S S-BAR NP	S VP SBAR S VP PP NP*

(b)

as opposed to the Stanford Tagger which uses complex hand-designed features and a complex form of discriminative training. Note the strong performance of MCMC sampling, which consistently outperforms A* search.

6.4 Analysis

For the analysis we focus on the syntactic parsing and POS tagging tasks. For each different spine size from 2 to 10, we extract the top-1000 frequent spines in the trees predicted based on our model, and compare them with those extracted from the gold standard trees. The numbers reported in Table 5(a), are the percentage of the intersection of these two sets. As reported in the table, in all cases (except one) the infinite order model ($v = \infty$) outperforms the model with limited size context ($v = 2$). Particularly in Danish POS tagging, our model predicts correctly 65% of top-1000 high-frequency spines of length 10 vs. 25% of the model with limited context. For syntactic parsing, the short range dependencies captured by limited context model ($v = 2$) over the spines of size 2 and

3 matches the results of our unbounded context model ($v = \infty$); however, the gap becomes wider for longer spines.

Our next analysis looks into the contexts of 4 linguistic categories in syntactic parsing: NP (noun phrase), VP (verb phrase), DT (determiner), and JJ (adjective). data set. We chose NP and VP mainly because they tend to appear in higher levels of the tree and most probably often in shorter contexts, and DT and JJ for the opposite reason. A list of the most frequent contexts for these syntactic categories in the trees predicted by our model is provided in Table 5(b); the ones marked with (*) exist in the gold standard trees as well. Our model successfully retrieves most of the long and short high-frequency contexts for the aforementioned syntactic categories.

7 Conclusion and Future Work

We have proposed a novel hierarchical model over linguistic trees which exploits global context by conditioning the generation of a rule in a tree on an unbounded tree context consisting of the vertical chain of its ancestors.

To facilitate learning of such a large and unbounded model, the predictive distributions associated with tree contexts are smoothed in a recursive manner using a hierarchical Pitman-Yor process. We have shown how to perform prediction based on our model to predict the parse tree of a given utterance using various search algorithms, e.g. A* and Markov Chain Monte Carlo. This consistently improved over baseline methods in several tasks, and produced state-of-the-art results for Danish part-of-speech tagging.

In future, we would like to consider sampling the seating arrangements and model hyperparameters, and seek to incorporate several different notions of context besides the chain of ancestors.

Acknowledgments. The authors are grateful to National ICT Australia (NICTA) for generous funding, as part of collaborative machine learning research projects. This work was funded in part by the Australian Research Council. We would like to thank the anonymous reviewers for their constructive comments.

References

Beal, M.J., Ghahramani, Z., Rasmussen, C.E.: The infinite hidden markov model. In: Advances in Neural Information Processing Systems, Vancouver, British Columbia, Canada, pp. 577–584 (2001)

Brants, T.: Tnt - A statistical part-of-speech tagger. In: Proceedings of the Sixth Conference on Applied Natural Language Processing, pp. 224–231 (2000)

Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In: Proceedings of the 34th Annual meeting on Association for Computational Linguistics, pp. 310–318. Association for Computational Linguistics (1996)

Cocke, J., Schwartz, J.T.: Programming languages and their compilers : preliminary notes. Technical report (1970)

Cohn, T., Blunsom, P., Goldwater, S.: Inducing tree-substitution grammars. The Journal of Machine Learning Research **11**, 3053–3096 (2010)

Finkel, J., Grenager, T., Manning, C.: The infinite tree. In: Proceedings of the 45th Annual Meeting of Association for Computational Linguistics, pp. 272–279 (2007)

Gasthaus, J., Teh, Y.W.: Improvements to the sequence memoizer. In: Advances in Neural Information Processing Systems, pp. 685–693 (2010)

Gelling, D., Cohn, T., Blunsom, P., Graca, J.: The PASCAL challenge on grammar induction. In: Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure, pp. 64–80. Association for Computational Linguistics (2012)

Goodman, J.: Parsing algorithms and metrics. In: Proceedings of the 34th Annual Meeting on Association for Computational Linguistics, ACL 1996, Stroudsburg, PA, USA, pp. 177–183. Association for Computational Linguistics (1996)

Johnson, M.: Pcfg models of linguistic tree representations. Computational Linguistics **24**(4), 613–632 (1998). ISSN 0891–2017

Johnson, M.: Unsupervised word segmentation for Sesotho using adaptor grammars. In: Proceedings of the 10th Meeting of ACL Special Interest Group on Computational Morphology and Phonology, Columbus, Ohio, pp. 20–27. Association for Computational Linguistics, June 2008

Johnson, M., Griffiths, T.L., Goldwater, S.: Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. In: Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4–7, pp. 641–648 (2006)

Johnson, M., Griffiths, T.L., Goldwater, S.: Bayesian inference for pcfgs via markov chain monte carlo. In: HLT-NAACL, pp. 139–146 (2007)

Klein, D., Manning, C.D.: Parsing and hypergraphs. In: Proceedings of the Seventh International Workshop on Parsing Technologies (IWPT-2001), October 17–19, Beijing, China (2001)

Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol. 1, pp. 423–430. Association for Computational Linguistics (2003)

Levenberg, A., Dyer, C., Blunsom, P.: A bayesian model for learning scfgs with discontiguous rules. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 223–232. Association for Computational Linguistics (2012)

Liang, P., Petrov, S., Jordan, M., Klein., D.: The infinite PCFG using hierarchical dirichlet processes. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 688–697 (2007)

Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: The penn treebank. Computational Linguistics **19**(2), 313–330 (1993)

Matsuzaki, T., Miyao, Y., Tsujii, J.: Probabilistic cfg with latent annotations. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL 2005, Stroudsburg, PA, USA, pp. 75–82. Association for Computational Linguistics (2005). doi:10.3115/1219840.1219850

Mochihashi, D., Sumita, E.: The infinite markov model. In: Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Systems, Vancouver, British Columbia, Canada (2007)

Petrov, S., Klein, D.: Learning and inference for hierarchically split PCFGs. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, Vancouver, British Columbia, Canada (2007)

Petrov, S., Barrett, L., Thibaux, R., Klein, D.: Learning accurate, compact, and interpretable tree annotation. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pp. 433–440. Association for Computational Linguistics (2006)

Teh, Y.W.: A hierarchical bayesian language model based on pitman-yor processes. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pp. 985–992. Association for Computational Linguistics (2006)

Thede, S.M., Harper, M.P.:. A second-order hidden markov model for part-of-speech tagging. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, ACL 1999, Stroudsburg, PA, USA, pp. 175–182. Association for Computational Linguistics (1999). ISBN 1-55860-609-3

Toutanova, K., Manning, M.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 63–70. Association for Computational Linguistics (2000)

Wood, F., Archambeau, C., Gasthaus, J., James, L., Teh, Y.W.: A stochastic memoizer for sequence data. In Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14–18, p. 142 (2009a)

Wood, F., Archambeau, C., Gasthaus, J., James, L., Teh, Y.W.: A stochastic memoizer for sequence data. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1129–1136. ACM (2009b)

Wood, F., Gasthaus, J., Archambeau, C., James, L., Teh, Y.W.: The sequence memoizer. Commun. ACM **54**(2), 91–98 (2011)

Zhu, C., Byrd, R.H., Lu, P., Nocedal, J.: Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software (TOMS) **23**(4), 550–560 (1997)