A Supertag-Context Model for Weakly-Supervised CCG Parser Learning

Dan Garrette[*] **Chris Dyer**[†] **Jason Baldridge**[‡] **Noah A. Smith**[†]

[*]Computer Science & Engineering, University of Washington, dhg@cs.washington.edu
[†]School of Computer Science, Carnegie Mellon University, {cdyer,nasmith}@cs.cmu.edu
[‡]Department of Linguistics, University of Texas at Austin, jbaldrid@utexas.edu

Abstract

Combinatory Categorial Grammar (CCG) is a lexicalized grammar formalism in which words are associated with categories that specify the syntactic configurations in which they may occur. We present a novel parsing model with the capacity to capture the associative adjacent-category relationships intrinsic to CCG by parameterizing the relationships between each constituent label and the preterminal categories directly to its left and right, biasing the model toward constituent categories that can combine with their contexts. This builds on the intuitions of Klein and Manning's (2002) "constituent-context" model, which demonstrated the value of modeling context, but has the advantage of being able to exploit the properties of CCG. Our experiments show that our model outperforms a baseline in which this context information is not captured.

1 Introduction

Learning parsers from incomplete or indirect supervision is an important component of moving NLP research toward new domains and languages. But with less information, it becomes necessary to devise ways of making better use of the information that is available. In general, this means constructing inductive biases that take advantage of unannotated data to train probabilistic models.

One important example is the constituent-context model (CCM) of Klein and Manning (2002), which was specifically designed to capture the linguistic observation made by Radford (1988) that there are regularities to the contexts in which constituents appear. This phenomenon, known as *substitutability*, says that phrases of the same type appear in similar contexts. For example, the part-of-speech (POS) sequence ADJ NOUN frequently occurs between the tags DET and VERB. This DET—VERB context also frequently applies to the single-word sequence NOUN and to ADJ ADJ NOUN. From this, we might deduce that DET—VERB is a likely context for a noun phrase. CCM is able to learn which POS contexts are likely, and does so via a probabilistic generative model, providing a statistical, data-driven take on substitutability. However, since there is nothing intrinsic about the POS pair DET—VERB that indicates *a priori* that it is a likely constituent context, this fact must be inferred entirely from the data.

Baldridge (2008) observed that unlike opaque, atomic POS labels, the rich structures of Combinatory Categorial Grammar (CCG) (Steedman, 2000; Steedman and Baldridge, 2011) categories reflect universal grammatical properties. CCG is a lexicalized grammar formalism in which every constituent in a sentence is associated with a structured category that specifies its syntactic relationship to other constituents. For example, a category might encode that "this constituent can combine with a noun phrase to the right (an object) and then a noun phrase to the left (a subject) to produce a sentence" instead of simply VERB. CCG has proven useful as a framework for grammar induction due to its ability to incorporate linguistic knowledge to guide parser learning by, for example, specifying rules in lexical-expansion algorithms (Bisk and Hockenmaier, 2012; 2013) or encoding that information as priors within a Bayesian framework (Garrette et al., 2015).

Baldridge observed is that, cross-linguistically, grammars prefer simpler syntactic structures when possible, and that due to the natural correspondence of categories and syntactic structure, biasing toward simpler categories encourages simpler structures. In previous work, we were able to incorporate this preference into a Bayesian parsing model, biasing PCFG productions toward simpler categories by encoding a notion of category simplicity into a prior (Garrette et al., 2015). Baldridge further notes that due to the natural associativity of CCG, adjacent categories tend to be combinable. We previously showed that incorporating this intuition into a Bayesian prior can help train a CCG supertagger (Garrette et al., 2014).

In this paper, we present a novel parsing model that is designed specifically for the capacity to capture both of these universal, intrinsic properties of CCG. We do so by extending our previous, PCFG-based parsing model to include parameters that govern the relationship between constituent categories and the preterminal categories (also known as *supertags*) to the left and right. The advantage of modeling context within a CCG framework is that while CCM must learn which contexts are likely purely from the data, the CCG categories give us obvious *a priori* information about whether a context is likely for a given constituent based on whether the categories are combinable. Biasing our model towards both simple categories and connecting contexts encourages learning structures with simpler syntax and that have a better global "fit".

The Bayesian framework is well-matched to our problem since our inductive biases — those derived from universal grammar principles, weak supervision, and estimations based on unannotated data — can be encoded as priors, and we can use Markov chain Monte Carlo (MCMC) inference procedures to automatically blend these biases with unannotated text that reflects the way language is actually used "in the wild". Thus, we learn context information based on statistics in the data like CCM, but have the advantage of additional, *a priori* biases. It is important to note that the Bayesian setup allows us to use these universal biases as *soft* constraints: they guide the learner toward more appropriate grammars, but may be overridden when there is compelling contradictory evidence in the data.

Methodologically, this work serves as an example of how linguistic-theoretical commitments can be used to benefit data-driven methods, not only through the construction of a model family from a grammar, as done in our previous work, but also when exploiting statistical associations about which the theory is silent. While there has been much work in computational modeling of the interaction between universal grammar and observable data in the context of studying child language acquisition (e.g., Villavicencio, 2002; Goldwater, 2007), we are interested in applying these principles to the design of models and learning procedures that result in better parsing tools. Given our desire to train NLP models in low-supervision scenarios, the possibility of constructing inductive biases out of universal properties of language is enticing: if we can do this well, then it only needs to be done once, and can be applied to any language or domain without adaptation.

In this paper, we seek to learn from only raw data and an incomplete dictionary mapping some words to sets of potential supertags. In order to estimate the parameters of our model, we develop a blocked sampler based on that of Johnson et al. (2007) to sample parse trees for sentences in the raw training corpus according to their posterior probabilities. However, due to the very large sets of potential supertags used in a parse, computing inside charts is intractable, so we design a Metropolis-Hastings step that allows us to sample efficiently from the correct posterior. Our experiments show that the incorporation of supertag context parameters into the model improves learning, and that placing combinability-preferring priors on those parameters yields further gains in many scenarios.

2 Combinatory Categorial Grammar

In the CCG formalism, every constituent, including those at the lexical level, is associated with a structured CCG category that defines that constituent's relationships to the other constituents in the sentence. Categories are defined by a recursive structure, where a category is either *atomic* (possibly with *features*), or a function from one category to another, as indicated by a slash operator:

$$C \rightarrow \{\mathsf{s}, \mathsf{s}_{\mathsf{dcl}}, \mathsf{s}_{\mathsf{adj}}, \mathsf{s}_{\mathsf{b}}, \mathsf{np}, \mathsf{n}, \mathsf{n}_{\mathsf{num}}, \mathsf{pp}, ...\}$$
$$C \rightarrow \{(C/C), (C\backslash C)\}$$

Categories of adjacent constituents can be combined using one of a set of combination rules to form categories of higher-level constituents, as seen in Figure 1. The direction of the slash operator gives the behavior of the function. A category $(\mathsf{s}\backslash\mathsf{np})/\mathsf{pp}$ might describe an intransitive verb with a prepositional phrase complement; it combines on the right (/) with a constituent with category pp, and then on the left (\) with a noun phrase (np) that serves as its subject.

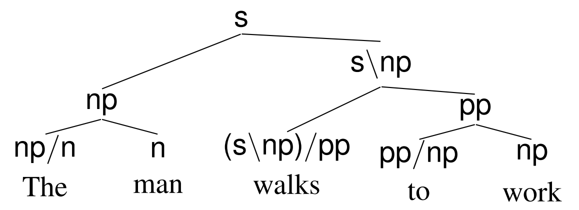

Figure 1: CCG parse for "The man walks to work."

We follow Lewis and Steedman (2014) in allowing a small set of generic, linguistically-plausible unary and binary grammar rules. We further add rules for combining with punctuation to the left and right and allow for the *merge* rule $X \rightarrow X\ X$ of Clark and Curran (2007).

3 Generative Model

In this section, we present our novel supertag-context model (SCM) that augments a standard PCFG with parameters governing the supertags to the left and right of each constituent.

The CCG formalism is said to be naturally *associative* since a constituent label is often able to combine on either the left or the right. As a motivating example, consider the sentence "The lazy dog sleeps", as shown in Figure 2. The word *lazy*, with category n/n, can either combine with *dog* (n) via the Forward Application rule (>), or with *The* (np/n) via the Forward Composition (>**B**) rule. Baldridge (2008) showed that this tendency for adjacent supertags to be combinable can be used to bias a sequence model in order to learn better CCG supertaggers. However, we can see that if the supertags of adjacent words *lazy* (n/n) and *dog* (n) combine, then they will produce the category n, which describes the entire constituent span "lazy dog". Since we have produced a new category that subsumes that entire span, a valid parse must next combine that n with one of the remaining supertags to the left or right, producing either (The·(lazy·dog))·sleeps or The·((lazy·dog)·sleeps). Because we know that one (or both) of these combinations must be valid, we will similarly want a strong prior on the connectivity between lazy·dog and its supertag context: The↔(lazy·dog)↔sleeps.

Assuming $\mathcal{T}$ is the full set of known categories, the generative process for our model is:

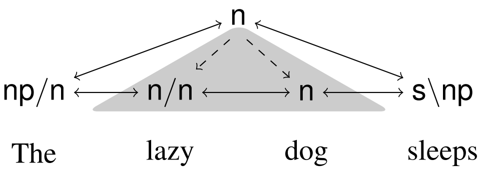

Figure 2: Higher-level category n subsumes the categories of its constituents. Thus, n should have a strong prior on combinability with its adjacent supertags np/n and s\np.

```
Parameters:
    $\theta^{\text{ROOT}} \sim \text{Dir}(\alpha^{\text{ROOT}}, \theta^{\text{ROOT-0}})$
    $\theta_{\mathbf{t}}^{\text{BIN}} \sim \text{Dir}(\alpha^{\text{BIN}}, \theta^{\text{BIN-0}}) \quad \forall \mathbf{t} \in \mathcal{T}$
    $\theta_{\mathbf{t}}^{\text{UN}} \sim \text{Dir}(\alpha^{\text{UN}}, \theta^{\text{UN-0}}) \quad \forall \mathbf{t} \in \mathcal{T}$
    $\theta_{\mathbf{t}}^{\text{TERM}} \sim \text{Dir}(\alpha^{\text{TERM}}, \theta_{\mathbf{t}}^{\text{TERM-0}}) \quad \forall \mathbf{t} \in \mathcal{T}$
    $\lambda_{\mathbf{t}} \sim \text{Dir}(\alpha_{\lambda}, \lambda^{0}) \quad \forall \mathbf{t} \in \mathcal{T}$
    $\theta_{\mathbf{t}}^{\text{LCTX}} \sim \text{Dir}(\alpha^{\text{LCTX}}, \theta_{\mathbf{t}}^{\text{LCTX-0}}) \quad \forall \mathbf{t} \in \mathcal{T}$
    $\theta_{\mathbf{t}}^{\text{RCTX}} \sim \text{Dir}(\alpha^{\text{RCTX}}, \theta_{\mathbf{t}}^{\text{RCTX-0}}) \quad \forall \mathbf{t} \in \mathcal{T}$
Sentence:
    do  $\mathbf{s} \sim \text{Cat}(\theta^{\text{ROOT}})$
        $\mathbf{y} \mid \mathbf{s} \sim \text{SCM}(\mathbf{s})$
    until the tree $\mathbf{y}$ is valid
  where $\langle \boldsymbol{\ell}, \mathbf{y}, \mathbf{r} \rangle \mid \mathbf{t} \sim \text{SCM}(\mathbf{t})$ is defined as:
    $z \sim \text{Cat}(\lambda_{\mathbf{t}})$
    if $z = \text{B}$ :   $\langle \mathbf{u}, \mathbf{v} \rangle \mid \mathbf{t} \sim \text{Cat}(\theta_{\mathbf{t}}^{\text{BIN}})$
        $\mathbf{y}_{\text{L}} \mid \mathbf{u} \sim \text{SCM}(\mathbf{u}), \quad \mathbf{y}_{\text{R}} \mid \mathbf{v} \sim \text{SCM}(\mathbf{v})$
        $\mathbf{y} = \langle \mathbf{y}_{\text{L}}, \mathbf{y}_{\text{R}} \rangle$
    if $z = \text{U}$ :   $\langle \mathbf{u} \rangle \mid \mathbf{t} \sim \text{Cat}(\theta_{\mathbf{t}}^{\text{UN}})$
        $\mathbf{y} \mid \mathbf{u} \sim \text{SCM}(\mathbf{u})$
    if $z = \text{T}$ :   $w \mid \mathbf{t} \sim \text{Cat}(\theta_{\mathbf{t}}^{\text{TERM}})$
        $\mathbf{y} = w$
    $\boldsymbol{\ell} \mid \mathbf{t} \sim \text{Cat}(\theta_{\mathbf{t}}^{\text{LCTX}}), \quad \mathbf{r} \mid \mathbf{t} \sim \text{Cat}(\theta_{\mathbf{t}}^{\text{RCTX}})$
```

The process begins by sampling the parameters from Dirichlet distributions: a distribution θ^{ROOT} over root categories, a conditional distribution $\theta_{\mathbf{t}}^{\text{BIN}}$ over binary branching productions given category $\mathbf{t}$, $\theta_{\mathbf{t}}^{\text{UN}}$ for unary rewrite productions, $\theta_{\mathbf{t}}^{\text{TERM}}$ for terminal (word) productions, and $\theta_{\mathbf{t}}^{\text{LCTX}}$ and $\theta_{\mathbf{t}}^{\text{RCTX}}$ for left and right contexts. We also sample parameters $\lambda_{\mathbf{t}}$ for the probability of $\mathbf{t}$ producing a binary branch, unary rewrite, or terminal word.

Next we sample a sentence. This begins by sampling first a root category $\mathbf{s}$ and then recursively sampling subtrees. For each subtree rooted by a category $\mathbf{t}$, we generate a left context supertag $\boldsymbol{\ell}$ and a right context supertag $\mathbf{r}$. Then, we sam-

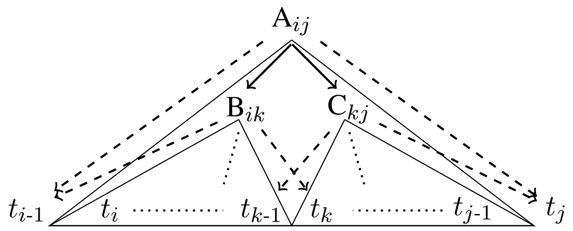

Figure 3: The generative process starting with non-terminal A_{ij}, where t_x is the supertag for w_x, the word at position x, and "A → B C" is a valid production in the grammar. We can see that non-terminal A_{ij} generates nonterminals B_{ik} and C_{kj} (solid arrows) as well as generating left context $t_{i\text{-}1}$ and right context t_j (dashed arrows); likewise for B_{ik} and C_{kj}. The triangle under a non-terminal indicates the complete subtree rooted by the node.

ple a production type z corresponding to either a (B) binary, (U) unary, or (T) terminal production. Depending on z, we then sample either a binary production $\langle \mathbf{u}, \mathbf{v} \rangle$ and recurse, a unary production $\langle \mathbf{u} \rangle$ and recurse, or a terminal word w and end that branch. A tree is complete when all branches end in terminal words. See Figure 3 for a graphical depiction of the generative behavior of the process. Finally, since it is possible to generate a supertag context category that does not match the actual category generated by the neighboring constituent, we must allow our process to reject such invalid trees and re-attempt to sample.

Like CCM, this model is *deficient* since the same supertags are generated multiple times, and parses with conflicting supertags are not valid. Since we are not generating from the model, this does not introduce difficulties (Klein and Manning, 2002).

One additional complication that must be addressed is that left-frontier non-terminal categories — those whose subtree span includes the first word of the sentence — do not have a left-side supertag to use as context. For these cases, we use the special sentence-start symbol $\langle \text{S} \rangle$ to serve as context. Similarly, we use the end symbol $\langle \text{E} \rangle$ for the right-side context of the right-frontier.

We next discuss how the prior distributions are constructed to encode desirable biases, using universal CCG properties.

3.1 Non-terminal production prior means

For the root, binary, and unary parameters, we want to choose prior means that encode our bias toward cross-linguistically-plausible categories. To formalize the notion of what it means for a category to be more "plausible", we extend the *category generator* of our previous work, which we will call P_{CAT}. We can define P_{CAT} using a probabilistic grammar (Garrette et al., 2014). The grammar may first generate a start or end category ($\langle \text{S} \rangle$,$\langle \text{E} \rangle$) with probability p_{se} or a special token-deletion category ($\langle \text{D} \rangle$; explained in §5) with probability p_{del}, or a standard CCG category C:

$$\begin{array}{ll} X \rightarrow \langle \text{S} \rangle \mid \langle \text{E} \rangle & p_{se} \\ X \rightarrow \langle \text{D} \rangle & p_{del} \\ X \rightarrow C & (1 - (2p_{se} + p_{del})) \cdot P_{\text{C}}(C) \end{array}$$

For each sentence s, there will be one $\langle \text{S} \rangle$ and one $\langle \text{E} \rangle$, so we set $p_{se} = 1/(25+2)$, since the average sentence length in the corpora is roughly 25. To discourage the model from deleting tokens (only applies during testing), we set $p_{del} = 10^{-100}$.

For P_{C}, the distribution over standard categories, we use a recursive definition based on the structure of a CCG category. If $\overline{p} = 1 - p$, then:[1]

$$\begin{array}{ll} C \rightarrow a & p_{term} \cdot p_{atom}(a) \\ C \rightarrow A/A & \overline{p}_{term} \cdot p_{fwd} \cdot (\, p_{mod} \cdot P_{\text{C}}(A) + \\ & \qquad \overline{p}_{mod} \cdot P_{\text{C}}(A)^2 \,) \\ C \rightarrow A/B & \overline{p}_{term} \cdot p_{fwd} \cdot \;\; \overline{p}_{mod} \cdot P_{\text{C}}(A) \cdot P_{\text{C}}(B) \\ C \rightarrow A\backslash A & \overline{p}_{term} \cdot \overline{p}_{fwd} \cdot (\, p_{mod} \cdot P_{\text{C}}(A) + \\ & \qquad \overline{p}_{mod} \cdot P_{\text{C}}(A)^2 \,) \\ C \rightarrow A\backslash B & \overline{p}_{term} \cdot \overline{p}_{fwd} \cdot \;\; \overline{p}_{mod} \cdot P_{\text{C}}(A) \cdot P_{\text{C}}(B) \end{array}$$

The category grammar captures important aspects of what makes a category more or less likely: (1) simplicity is preferred, with a higher p_{term} meaning a stronger emphasis on simplicity;[2] (2) atomic types may occur at different rates, as given by p_{atom}; (3) modifier categories (A/A or $A\backslash A$) are more likely than similar-complexity non-modifiers (such as an adverb that modifies a verb); and (4) operators may occur at different rates, as given by p_{fwd}.

We can use P_{CAT} to define priors on our production parameters that bias our model toward rules

[1] Note that this version has also updated the probability definitions for modifiers to be sums, incorporating the fact that any A/A is *also* a A/B (likewise for $A\backslash A$). This ensures that our grammar defines a valid probability distribution.

[2] The probability distribution over categories is guaranteed to be proper so long as $p_{term} > \frac{1}{2}$ since the probability of the depth of a tree will decrease geometrically (Chi, 1999).

that result in *a priori* more likely categories:[3]

$$\theta^{\text{ROOT-0}}(\mathbf{t}) = P_{\text{CAT}}(\mathbf{t})$$
$$\theta^{\text{BIN-0}}(\langle \mathbf{u}, \mathbf{v} \rangle) = P_{\text{CAT}}(\mathbf{u}) \cdot P_{\text{CAT}}(\mathbf{v})$$
$$\theta^{\text{UN-0}}(\langle \mathbf{u} \rangle) = P_{\text{CAT}}(\mathbf{u})$$

For simplicity, we assume the production-type mixture prior to be uniform: $\lambda^0 = \langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \rangle$.

3.2 Terminal production prior means

We employ the same procedure as our previous work for setting the terminal production prior distributions $\theta_{\mathbf{t}}^{\text{TERM-0}}(w)$ by estimating word-given-category relationships from the weak supervision: the tag dictionary and raw corpus (Garrette and Baldridge, 2012; Garrette et al., 2015).[4] This procedure attempts to automatically estimate the frequency of each word/tag combination by dividing the number of raw-corpus occurrences of each word in the dictionary evenly across all of its associated tags. These counts are then combined with estimates of the "openness" of each tag in order to assess its likelihood of appearing with new words.

3.3 Context parameter prior means

In order to encourage our model to choose trees in which the constituent labels "fit" into their supertag contexts, we want to bias our context parameters toward context categories that are *combinable* with the constituent label.

The right-side context of a non-terminal category — the probability of generating a category to the right of the current constituent's category — corresponds directly to the category transitions used for the HMM supertagger of Garrette et al. (2014). Thus, the right-side context prior mean $\theta_{\mathbf{t}}^{\text{RCTX-0}}$ can be biased in exactly the same way as the HMM supertagger's transitions: toward context supertags that connect to the constituent label.

To encode a notion of combinability, we follow Baldridge's (2008) definition. Briefly, let $\kappa(\mathbf{t}, \mathbf{u}) \in \{0, 1\}$ be an indicator of whether **t** combines with **u** (in that order). For any binary rule that can combine **t** to **u**, $\kappa(\mathbf{t}, \mathbf{u})$=1. To ensure that our prior captures the natural associativity of CCG, we define combinability in this context to include composition rules as well as application rules. If atoms have *features* associated, then the atoms are allowed to unify if the features match, or if at least one of them does not have a feature. In defining κ, it is also important to ignore possible arguments on the wrong side of the combination since they can be consumed without affecting the connection between the two. To achieve this for $\kappa(\mathbf{t}, \mathbf{u})$, it is assumed that it is possible to consume all preceding arguments of **t** and all following arguments of **u**. So $\kappa(\mathsf{np}, (\mathsf{s}\backslash\mathsf{np})/\mathsf{np}) = 1$. This helps to ensure the associativity discussed earlier. For "combining" with the start or end of a sentence, we define $\kappa(\langle \text{S} \rangle, \mathbf{u})$=1 when **u** seeks no left-side arguments (since there are no tags to the left with which to combine) and $\kappa(\mathbf{t}, \langle \text{E} \rangle)$=1 when **t** seeks no right-side arguments. So $\kappa(\langle \text{S} \rangle, \mathsf{np}/\mathsf{n})$=1, but $\kappa(\langle \text{S} \rangle, \mathsf{s}\backslash\mathsf{np})$=0. Finally, due to the frequent use of the unary rule that allows n to be rewritten as np, the atom np is allowed to unify with n if n is the argument. So $\kappa(\mathsf{n}, \mathsf{s}\backslash\mathsf{np}) = 1$, but $\kappa(\mathsf{np}/\mathsf{n}, \mathsf{np}) = 0$.

The prior mean of producing a right-context supertag **r** from a constituent category **t**, $P^{right}(\mathbf{r} \mid \mathbf{t})$, is defined so that combinable pairs are given higher probability than non-combinable pairs. We further experimented with a prior that biases toward both combinability *and* category likelihood, replacing the uniform treatment of categories with our prior over categories, yielding $P_{\text{CAT}}^{right}(\mathbf{r} \mid \mathbf{t})$. If $\mathcal{T}$ is the full set of known CCG categories:

$$P^{right}(\mathbf{r} \mid \mathbf{t}) = \begin{cases} \sigma \cdot 1/|\mathcal{T}| & \text{if } \kappa(\mathbf{t}, \mathbf{r}) \quad \sigma > 1 \\ 1/|\mathcal{T}| & \text{otherwise} \end{cases}$$

$$P_{\text{CAT}}^{right}(\mathbf{r} \mid \mathbf{t}) = \begin{cases} \sigma \cdot P_{\text{CAT}}(\mathbf{r}) & \text{if } \kappa(\mathbf{t}, \mathbf{r}) \quad \sigma > 1 \\ P_{\text{CAT}}(\mathbf{r}) & \text{otherwise} \end{cases}$$

Distributions $P^{left}(\boldsymbol{\ell} \mid \mathbf{t})$ and $P_{\text{CAT}}^{left}(\boldsymbol{\ell} \mid \mathbf{t})$ are defined in the same way, but with the combinability direction flipped: $\kappa(\boldsymbol{\ell}, \mathbf{t})$, since the left context supertag precedes the constituent category.

4 Posterior Inference

We wish to infer the distribution over CCG parses, given the model we just described and a corpus of sentences. Since there is no way to analytically compute these modes, we resort to Gibbs sampling to find an approximate solution. Our strategy is based on the approach presented by Johnson et al. (2007). At a high level, we alternate between resampling model parameters (θ^{ROOT}, θ^{BIN}, θ^{UN}, θ^{TERM}, λ, θ^{LCTX}, θ^{RCTX}) given the current set of parse trees and resampling those trees given the

[3] For our experiments, we normalize P_{CAT} by dividing by $\sum_{\mathbf{c} \in \mathcal{T}} P_{\text{CAT}}(\mathbf{c})$. This allows for experiments contrasting with a uniform prior $(1/|\mathcal{T}|)$ without adjusting α values.

[4] We refer the reader to the previous work (Garrette et al., 2015) for a fuller discussion and implementation details.

current model parameters and observed word sequences. To efficiently sample new model parameters, we exploit Dirichlet-multinomial conjugacy. By repeating these alternating steps and accumulating the productions, we obtain an approximation of the required posterior quantities.

Our inference procedure takes as input the distribution prior means, along with the raw corpus and tag dictionary. During sampling, we restrict the tag choices for a word w to categories allowed by the tag dictionary. Since real-world learning scenarios will always lack complete knowledge of the lexicon, we, too, want to allow for unknown words; for these, we assume the word may take any known supertag. We refer to the sequence of word tokens as **w** and a non-terminal category covering the span i through $j-1$ as y_{ij}.

While it is technically possible to sample directly from our context-sensitive model, the high number of potential supertags available for each context means that computing the inside chart for this model is intractable for most sentences. In order to overcome this limitation, we employ an accept/reject Metropolis-Hastings (MH) step. The basic idea is that we sample trees according to a simpler *proposal* distribution Q that approximates the full distribution and for which direct sampling is tractable, and then choose to accept or reject those trees based on the true distribution P.

For our model, there is a straightforward and intuitive choice for the proposal distribution: the PCFG model without our context parameters: (θ^{ROOT}, θ^{BIN}, θ^{UN}, θ^{TERM}, λ), which is known to have an efficient sampling method. Our acceptance step is therefore based on the remaining parameters: the context (θ^{LCTX}, θ^{RCTX}).

To sample from our proposal distribution, we use a blocked Gibbs sampler based on the one proposed by Goodman (1998) and used by Johnson et al. (2007) that samples entire parse trees. For a sentence **w**, the strategy is to use the Inside algorithm (Lari and Young, 1990) to inductively compute, for each potential non-terminal position spanning words w_i through w_{j-1} and category **t**, going "up" the tree, the probability of generating $w_i, \ldots, w_{j-1}$ via any arrangement of productions that is rooted by $y_{ij} = \mathbf{t}$.

$$\begin{aligned} p(w_i \mid y_{i,i+1} = \mathbf{t}) &= \lambda_{\mathbf{t}}(\text{T}) \cdot \theta^{\text{TERM}}_{\mathbf{t}}(w_i) \\ &\quad + \textstyle\sum_{\mathbf{t}\to\mathbf{u}} \lambda_{\mathbf{t}}(\text{U}) \cdot \theta^{\text{UN}}_{\mathbf{t}}(\langle \mathbf{u} \rangle) \\ &\qquad \cdot p(w_{i:j-1} \mid y_{ij} = \mathbf{u}) \end{aligned}$$

$$\begin{aligned} &p(w_{i:j-1} \mid y_{ij} = \mathbf{t}) = \\ &\quad \textstyle\sum_{\mathbf{t}\to\mathbf{u}} \lambda_{\mathbf{t}}(\text{U}) \cdot \theta^{\text{UN}}_{\mathbf{t}}(\langle \mathbf{u} \rangle) \\ &\qquad \cdot p(w_{i:j-1} \mid y_{ij} = \mathbf{u}) \\ &\quad + \textstyle\sum_{\mathbf{t}\to\mathbf{u}\ \mathbf{v}} \ \sum_{i<k<j} \lambda_{\mathbf{t}}(\text{B}) \cdot \theta^{\text{BIN}}_{\mathbf{t}}(\langle \mathbf{u}, \mathbf{v} \rangle) \\ &\qquad \cdot p(w_{i:k-1} \mid y_{ik} = \mathbf{u}) \\ &\qquad \cdot p(w_{k:j-1} \mid y_{kj} = \mathbf{v}) \end{aligned}$$

We then pass "downward" through the chart, sampling productions until we reach a terminal word on all branches.

$$\begin{aligned} y_{0n} &\sim \ \theta^{\text{ROOT}}_{\mathbf{t}} \cdot p(w_{0:n-1} \mid y_{0n} = \mathbf{t}) \\ x \mid y_{ij} &\sim \Big\langle \theta^{\text{BIN}}_{y_{ij}}(\langle \mathbf{u}, \mathbf{v} \rangle) \cdot p(w_{i:k-1} \mid y_{ik} = \mathbf{u}) \\ &\qquad \cdot p(w_{k:j-1} \mid y_{kj} = \mathbf{v}) \\ &\qquad \forall\, y_{ik}, y_{kj} \text{ when } j > i+1, \\ &\quad \theta^{\text{UN}}_{y_{ij}}(\langle \mathbf{u} \rangle) \cdot p(w_{i:j-1} \mid y'_{ij} = \mathbf{u}) \quad \forall\, y'_{ij}, \\ &\quad \theta^{\text{TERM}}_{y_{ij}}(w_i) \qquad \text{when } j = i+1 \Big\rangle \end{aligned}$$

where x is either a split point k and pair of categories y_{ik}, y_{kj} resulting from a binary rewrite rule, a single category y'_{ij} resulting from a unary rule, or a word w resulting from a terminal rule.

The MH procedure requires an *acceptance distribution* A that is used to accept or reject a tree sampled from the proposal Q. The probability of accepting new tree $\mathbf{y}'$ given the previous tree $\mathbf{y}$ is:

$$A(\mathbf{y}' \mid \mathbf{y}) = \min\left(1, \frac{P(\mathbf{y}')}{P(\mathbf{y})} \frac{Q(\mathbf{y})}{Q(\mathbf{y}')}\right)$$

Since Q is defined as a subset of P's parameters, it is the case that:

$$P(\mathbf{y}) = Q(\mathbf{y}) \cdot p(\mathbf{y} \mid \theta^{\text{LCTX}}, \theta^{\text{RCTX}})$$

After substituting this for each P in A, all of the Q factors cancel, yielding the acceptance distribution defined purely in terms of context parameters:

$$A(\mathbf{y}' \mid \mathbf{y}) = \min\left(1, \frac{p(\mathbf{y}' \mid \theta^{\text{LCTX}}, \theta^{\text{RCTX}})}{p(\mathbf{y} \mid \theta^{\text{LCTX}}, \theta^{\text{RCTX}})}\right)$$

For completeness, we note that the probability of a tree $\mathbf{y}$ given only the context parameters is:[5]

$$\begin{aligned} &p(\mathbf{y} \mid \theta^{\text{LCTX}}, \theta^{\text{RCTX}}) = \\ &\quad \prod_{0 \le i < j \le n} \theta^{\text{LCTX}}(y_{i-1,i} \mid y_{ij}) \cdot \theta^{\text{RCTX}}(y_{j,j+1} \mid y_{ij}) \end{aligned}$$

[5] Note that there may actually be multiple y_{ij} due to unary rules that "loop back" to the same position (i, j); all of these much be included in the product.

Before we begin sampling, we initialize each distribution to its prior mean ($\theta^{\text{ROOT}}{=}\theta^{\text{ROOT-0}}$, $\theta^{\text{BIN}}_{\mathbf{t}}{=}\theta^{\text{BIN-0}}$, etc). Since MH requires an initial set of trees to begin sampling, we parse the raw corpus with probabilistic CKY using these initial parameters (excluding the context parameters) to guess an initial tree for each raw sentence.

The sampler alternates sampling parse trees for the entire corpus of sentences using the above procedure with resampling the model parameters. Resampling the parameters requires empirical counts of each production. These counts are taken from the trees resulting from the previous round of sampling: new trees that have been "accepted" by the MH step, as well as existing trees for sentences in which the newly-sampled tree was rejected.

$$\begin{aligned}
\theta^{\text{ROOT}} &\sim \text{Dir}(\langle \alpha^{\text{ROOT}} \cdot \theta^{\text{ROOT-0}}(\mathbf{t}) \quad\; + C_{\mathit{root}}(\mathbf{t})\ \rangle_{\mathbf{t}\in\mathcal{T}})\\
\theta^{\text{BIN}}_{\mathbf{t}} &\sim \text{Dir}(\langle \alpha^{\text{BIN}} \cdot \theta^{\text{BIN-0}}(\langle \mathbf{u},\mathbf{v}\rangle) + C(\mathbf{t}{\rightarrow}\langle \mathbf{u},\mathbf{v}\rangle)\ \rangle_{\mathbf{u},\mathbf{v}\in\mathcal{T}})\\
\theta^{\text{UN}}_{\mathbf{t}} &\sim \text{Dir}(\langle \alpha^{\text{UN}} \cdot \theta^{\text{UN-0}}(\langle \mathbf{u}\rangle) \quad\; + C(\mathbf{t}{\rightarrow}\langle \mathbf{u}\rangle)\ \rangle_{\mathbf{u}\in\mathcal{T}})\\
\theta^{\text{TERM}}_{\mathbf{t}} &\sim \text{Dir}(\langle \alpha^{\text{TERM}} \cdot \theta^{\text{TERM-0}}_{\mathbf{t}}(w) \;\; + C(\mathbf{t} \rightarrow w)\ \rangle_{w\in V})\\
\lambda_{\mathbf{t}} &\sim \text{Dir}(\langle \alpha_{\lambda} \cdot \lambda^{0}(\text{B}) + \textstyle\sum_{\mathbf{u},\mathbf{v}\in\mathcal{T}} C(\mathbf{t}{\rightarrow}\langle \mathbf{u},\mathbf{v}\rangle),\\
&\qquad\quad \alpha_{\lambda} \cdot \lambda^{0}(\text{U}) + \textstyle\sum_{\mathbf{u}\in\mathcal{T}} C(\mathbf{t}{\rightarrow}\langle \mathbf{u}\rangle),\\
&\qquad\quad \alpha_{\lambda} \cdot \lambda^{0}(\text{T}) + \textstyle\sum_{w\in V} C(\mathbf{t}{\rightarrow}w) \quad \rangle)\\
\theta^{\text{LCTX}}_{\mathbf{t}} &\sim \text{Dir}(\langle \alpha^{\text{LCTX}} \cdot \theta^{\text{LCTX-0}}_{\mathbf{t}}(\boldsymbol{\ell}) + C_{\mathit{left}}\,(\mathbf{t},\boldsymbol{\ell})\rangle_{\boldsymbol{\ell}\in\mathcal{T}})\\
\theta^{\text{RCTX}}_{\mathbf{t}} &\sim \text{Dir}(\langle \alpha^{\text{RCTX}} \cdot \theta^{\text{RCTX-0}}_{\mathbf{t}}(\mathbf{r}) + C_{\mathit{right}}(\mathbf{t},\mathbf{r})\rangle_{\mathbf{r}\in\mathcal{T}})
\end{aligned}$$

It is important to note that this method of resampling allows the draws to incorporate both the data, in the form of counts, and the prior mean, which includes all of our carefully-constructed biases derived from both the intrinsic, universal CCG properties as well as the information we induced from the raw corpus and tag dictionary.

After all sampling iterations have completed, the final model is estimated by pooling the trees resulting from each sampling iteration, including trees accepted by the MH steps as well as the duplicated trees retained due to rejections. We use this pool of trees to compute model parameters using the same procedure as we used directly above to sample parameters, except that instead of drawing a Dirichlet sample based on the vector of counts, we simply normalize those counts. However, since we require a final model that can parse sentences efficiently, we drop the context parameters, making the model a standard PCFG, which allows us to use the probabilistic CKY algorithm.

5 Experiments

In our evaluation we compared our supertag-context approach to (our reimplementation of) the best-performing model of our previous work (Garrette et al., 2015), which SCM extends. We evaluated on the English CCGBank (Hockenmaier and Steedman, 2007), which is a transformation of the Penn Treebank (Marcus et al., 1993); the CTB-CCG (Tse and Curran, 2010) transformation of the Penn Chinese Treebank (Xue et al., 2005); and the CCG-TUT corpus (Bos et al., 2009), built from the TUT corpus of Italian text (Bosco et al., 2000).

Each corpus was divided into four distinct data sets: a set from which we extract the tag dictionaries, a set of raw (unannotated) sentences, a development set, and a test set. We use the same splits as Garrette et al. (2014). Since these treebanks use special representations for conjunctions, we chose to rewrite the trees to use conjunction categories of the form $(X\backslash X)/X$ rather than introducing special conjunction rules. In order to increase the amount of raw data available to the sampler, we supplemented the English data with raw, unannotated newswire sentences from the NYT Gigaword 5 corpus (Parker et al., 2011) and supplemented Italian with the out-of-domain WaCky corpus (Baroni et al., 1999). For English and Italian, this allowed us to use 100k raw tokens for training (Chinese uses 62k). For Chinese and Italian, for training efficiency, we used only raw sentences that were 50 words or fewer (note that we did *not* drop tag dictionary set or test set sentences).

The English development set was used to tune hyperparameters using grid search, and the same hyperparameters were then used for all three languages. For the category grammar, we used $p_{\mathit{punc}}{=}0.1$, $p_{\mathit{term}}{=}0.7$, $p_{\mathit{mod}}{=}0.2$, $p_{\mathit{fwd}}{=}0.5$. For the priors, we use $\alpha^{\text{ROOT}}{=}1$, $\alpha^{\text{BIN}}{=}100$, $\alpha^{\text{UN}}{=}100$, $\alpha^{\text{TERM}}{=}10^4$, $\alpha_{\lambda}{=}3$, $\alpha^{\text{LCTX}}{=}\alpha^{\text{RCTX}}{=}10^3$.[6] For the context prior, we used $\sigma{=}10^5$. We ran our sampler for 50 burn-in and 50 sampling iterations.

CCG parsers are typically evaluated on the *dependencies* they produce instead of their CCG derivations directly since there can be many different CCG parse trees that all represent the same dependency relationships (spurious ambiguity), and CCG-to-dependency conversion can collapse those differences. To convert a CCG tree into a dependency tree, we follow Lewis and Steedman

[6] In order to ensure that these concentration parameters, while high, were not dominating the posterior distributions, we ran experiments in which they were set much higher (including using the prior alone), and found that accuracies plummeted in those cases, demonstrating that there is a good balance with the prior.

		Size of the corpus (tokens) from which the *tag dictionary* is extracted					
		250k	**200k**	**150k**	**100k**	**50k**	**25k**
English	no context	60.43	61.22	59.69	58.61	56.26	54.70
	+context (uniform)	64.02	**63.89**	62.58	61.80	59.44	57.08
	+P^{left} / P^{right}	**65.44**	63.26	**64.28**	**62.90**	**59.63**	**57.86**
	+$P^{\mathit{left}}_{\text{CAT}}$ / $P^{\mathit{right}}_{\text{CAT}}$	59.34	59.89	59.32	58.47	57.85	55.77
Chinese	no context				32.70	32.07	28.99
	+context (uniform)				**36.02**	33.84	32.55
	+P^{left} / P^{right}				35.34	33.04	31.48
	+$P^{\mathit{left}}_{\text{CAT}}$ / $P^{\mathit{right}}_{\text{CAT}}$				35.15	**34.04**	**33.53**
Italian	no context						51.54
	+context (uniform)						**53.57**
	+P^{left} / P^{right}						52.54
	+$P^{\mathit{left}}_{\text{CAT}}$ / $P^{\mathit{right}}_{\text{CAT}}$						53.29

Table 1: Experimental results in three languages. For each language, four experiments were executed: (1) a no-context model baseline, Garrette et al. (2015) directly; (2) our supertag-context model, with uniform priors on contexts; (3) supertag-context model with priors that prefer combinability; (4) supertag-context model with priors that prefer combinability *and* simpler categories. Results are shown for six different levels of supervision, as determined by the size of the corpus used to extract a tag dictionary.

(2014). We traverse the parse tree, dictating at every branching node which words will be the dependents of which. For binary branching nodes of *forward* rules, the right side—the argument side—is the dependent, unless the left side is a modifier (X/X) of the right, in which case the left is the dependent. The opposite is true for *backward* rules. For *punctuation* rules, the punctuation is always the dependent. For *merge* rules, the right side is always made the parent. The results presented in this paper are dependency accuracy scores: the proportion of words that were assigned the correct parent (or "root" for the root of a tree).

When evaluating on test set sentences, if the model is unable to find a parse given the constraints of the tag dictionary, then we would have to take a score of zero for that sentence: every dependency would be "wrong". Thus, it is important that we make a best effort to find a parse. To accomplish this, we implemented a parsing backoff strategy. The parser first tries to find a valid parse that has either $\mathsf{s}_{\mathsf{dcl}}$ or np at its root. If that fails, then it searches for a parse with any root. If no parse is found yet, then the parser attempts to strategically allow tokens to subsume a neighbor by making it a dependent (first with a restricted root set, then without). This is similar to the "deletion" strategy employed by Zettlemoyer and Collins (2007), but we do it directly in the grammar. We add unary rules of the form $\langle \textsc{d} \rangle \rightarrow \mathbf{u}$ for every potential supertag $\mathbf{u}$ in the tree. Then, at each node spanning exactly two tokens (but no higher in the tree), we allow rules $\mathbf{t} \rightarrow \langle \langle \textsc{d} \rangle, \mathbf{v} \rangle$ and $\mathbf{t} \rightarrow \langle \mathbf{v}, \langle \textsc{d} \rangle \rangle$. Recall that in §3.1, we stated that $\langle \textsc{d} \rangle$ is given extremely low probability, meaning that the parser will avoid its use unless it is absolutely necessary. Additionally, since $\mathbf{u}$ will still remain as the preterminal, it will be the category examined as the context by adjacent constituents.

For each language and level of supervision, we executed four experiments. The no-context baseline used (a reimplementation of) the best model from our previous work (Garrette et al., 2015): using only the non-context parameters (θ^{ROOT}, θ^{BIN}, θ^{UN}, θ^{TERM}, λ) along with the category prior P_{CAT} to bias toward likely categories throughout the tree, and $\theta^{\text{TERM-0}}_{\mathbf{t}}$ estimated from the tag dictionary and raw corpus. We then added the supertag-context parameters (θ^{LCTX}, θ^{RCTX}), but used uniform priors for those (still using P_{CAT} for the rest). Then, we evaluated the supertag-context model using context parameter priors that bias toward categories that combine with their contexts: P^{left} and P^{right} (see §3.3). Finally, we evaluated the supertag-context model using context parameter priors that bias toward combinability *and* toward *a priori* more likely categories, based on the category grammar ($P^{\mathit{left}}_{\text{CAT}}$ and $P^{\mathit{right}}_{\text{CAT}}$).

Because we are interested in understanding how our models perform under varying amounts of supervision, we executed sequences of experiments in which we reduced the size of the corpus from which the tag dictionary is drawn, thus reducing the amount of information provided to the model. As this information is reduced, so is the size of the full inventory of known CCG categories that can be used as supertags. Additionally, a smaller tag dictionary means that there will be vastly more unknown words; since our model must assume that these words may take any supertag from the full set of known labels, the model must contend with a greatly increased level of ambiguity.

The results of our experiments are given in Table 1. We find that the incorporation of supertag-context parameters into a CCG model improves performance in every scenario we tested; we see gains of 2–5% across the board. Adding context parameters never hurts, and in most cases, using priors based on intrinsic, cross-lingual aspects of the CCG formalism to bias those parameters toward connectivity provides further gains. In particular, biasing the model toward trees in which constituent labels are combinable with their adjacent supertags frequently helps the model.

However, for English, we found that additionally biasing *context* priors toward simpler categories using $P_{\text{CAT}}^{\textit{left}}/P_{\text{CAT}}^{\textit{right}}$ degraded performance. This is likely due to the fact that the priors on production parameters ($\theta^{\text{BIN}}, \theta^{\text{UN}}$) are already biasing the model toward likely categories, and that having the context parameters do the same ends up over-emphasizing the need for simple categories, preventing the model from choosing more complex categories when they are needed. On the other hand, this bias helps in Chinese and Italian.

6 Related Work

Klein and Manning (2002)'s CCM is an unlabeled bracketing model that generates the span of part-of-speech tags that make up each constituent and the pair of tags surrounding each constituent span (as well as the spans and contexts of each non-constituent). They found that modeling constituent context aids in parser learning because it is able to capture the observation that the same contexts tend to appear repeatedly in a corpus, even with different constituents. While CCM is designed to learn which tag pairs make for likely contexts, without regard for the constituents themselves, our model attempts to learn the relationships between context categories and the types of the constituents, allowing us to take advantage of the natural *a priori* knowledge about which contexts fit with which constituent labels.

Other researchers have shown positive results for grammar induction by introducing relatively small amounts of linguistic knowledge. Naseem et al. (2010) induced dependency parsers by hand-constructing a small set of linguistically-universal dependency rules and using them as soft constraints during learning. These rules were useful for disambiguating between various structures in cases where the data alone suggests multiple valid analyses. Boonkwan and Steedman (2011) made use of language-specific linguistic knowledge collected from non-native linguists via a questionnaire that covered a variety of syntactic parameters. They were able to use this information to induce CCG parsers for multiple languages. Bisk and Hockenmaier (2012; 2013) induced CCG parsers by using a smaller number of linguistically-universal principles to propose syntactic categories for each word in a sentence, allowing EM to estimate the model parameters. This allowed them to induce the inventory of language-specific types from the training data, without prior language-specific knowledge.

7 Conclusion

Because of the structured nature of CCG categories and the logical framework in which they must assemble to form valid parse trees, the CCG formalism offers multiple opportunities to bias model learning based on universal, intrinsic properties of the grammar. In this paper we presented a novel parsing model with the capacity to capture the associative adjacent-category relationships intrinsic to CCG by parameterizing *supertag contexts*, the supertags appearing on either side of each constituent. In our Bayesian formulation, we place priors on those context parameters to bias the model toward trees in which constituent labels are *combinable* with their contexts, thus preferring trees that "fit" together better. Our experiments demonstrate that, across languages, this additional context helps in weak-supervision scenarios.

Acknowledgements

We would like to thank Yonatan Bisk for his valuable feedback. This work was supported by the U.S. Department of Defense through the U.S. Army Research Office grant W911NF-10-1-0533.

References

Jason Baldridge. 2008. Weakly supervised supertagging with grammar-informed initialization. In *Proc. of COLING*.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 1999. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3).

Yonatan Bisk and Julia Hockenmaier. 2012. Simple robust grammar induction with combinatory categorial grammar. In *Proc. of AAAI*.

Yonatan Bisk and Julia Hockenmaier. 2013. An HDP model for inducing combinatory categorial grammars. *Transactions of the Association for Computational Linguistics*, 1.

Prachya Boonkwan and Mark Steedman. 2011. Grammar induction from text using small syntactic prototypes. In *Proc. of IJCNLP*.

Johan Bos, Cristina Bosco, and Alessandro Mazzei. 2009. Converting a dependency treebank to a categorial grammar treebank for Italian. In M. Passarotti, Adam Przepiórkowski, S. Raynaud, and Frank Van Eynde, editors, *Proc. of the Eighth International Workshop on Treebanks and Linguistic Theories (TLT8)*.

Cristina Bosco, Vincenzo Lombardo, Daniela Vassallo, and Leonardo Lesmo. 2000. Building a treebank for Italian: a data-driven annotation schema. In *Proc. of LREC*.

Zhiyi Chi. 1999. Statistical properties of probabilistic context-free grammars. *Computational Linguistics*.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33.

Dan Garrette and Jason Baldridge. 2012. Type-supervised hidden Markov models for part-of-speech tagging with incomplete tag dictionaries. In *Proc. of EMNLP*.

Dan Garrette, Chris Dyer, Jason Baldridge, and Noah A. Smith. 2014. Weakly-supervised Bayesian learning of a CCG supertagger. In *Proc. of CoNLL*.

Dan Garrette, Chris Dyer, Jason Baldridge, and Noah A. Smith. 2015. Weakly-supervised grammar-informed Bayesian CCG parser learning. In *Proc. of AAAI*.

Sharon Goldwater. 2007. *Nonparametric Bayesian Models of Lexical Acquisition*. Ph.D. thesis, Brown University.

Joshua Goodman. 1998. *Parsing inside-out*. Ph.D. thesis, Harvard University.

Julia Hockenmaier and Mark Steedman. 2007. CCG-bank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3).

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proc. of NAACL*.

Dan Klein and Christopher D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proc. of ACL*.

Karim Lari and Steve J. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proc. of EMNLP*.

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

Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *Proc. of EMNLP*.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English Gigaword Fifth Edition LDC2011T07. Linguistic Data Consortium.

Andrew Radford. 1988. *Transformational Grammar*. Cambridge University Press.

Mark Steedman and Jason Baldridge. 2011. Combinatory categorial grammar. In Robert Borsley and Kersti Borjars, editors, *Non-Transformational Syntax: Formal and Explicit Models of Grammar*. Wiley-Blackwell.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press.

Daniel Tse and James R. Curran. 2010. Chinese CCG-bank: Extracting CCG derivations from the Penn Chinese Treebank. In *Proc. of COLING*.

Aline Villavicencio. 2002. *The acquisition of a unification-based generalised categorial grammar*. Ph.D. thesis, University of Cambridge.

Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207–238.

Luke S. Zettlemoyer and Michael Collins. 2007. Online learning of relaxed CCG grammars for parsing to logical form. In *Proc. of EMNLP*.