A Feature-Rich Constituent Context Model for Grammar Induction

Dave Golland
University of California, Berkeley
dsg@cs.berkeley.edu

John DeNero
Google
denero@google.com

Jakob Uszkoreit
Google
uszkoreit@google.com

Abstract

We present LLCCM, a log-linear variant of the constituent context model (CCM) of grammar induction. LLCCM retains the simplicity of the original CCM but extends robustly to long sentences. On sentences of up to length 40, LLCCM outperforms CCM by 13.9% bracketing F1 and outperforms a right-branching baseline in regimes where CCM does not.

1 Introduction

Unsupervised grammar induction is a fundamental challenge of statistical natural language processing (Lari and Young, 1990; Pereira and Schabes, 1992; Carroll and Charniak, 1992). The constituent context model (CCM) for inducing constituency parses (Klein and Manning, 2002) was the first unsupervised approach to surpass a right-branching baseline. However, the CCM only effectively models short sentences. This paper shows that a simple reparameterization of the model, which ties together the probabilities of related events, allows the CCM to extend robustly to long sentences.

Much recent research has explored dependency grammar induction. For instance, the dependency model with valence (DMV) of Klein and Manning (2004) has been extended to utilize multilingual information (Berg-Kirkpatrick and Klein, 2010; Cohen et al., 2011), lexical information (Headden III et al., 2009), and linguistic universals (Naseem et al., 2010). Nevertheless, simplistic dependency models like the DMV do not contain information present in a constituency parse, such as the attachment order of object and subject to a verb.

Unsupervised constituency parsing is also an active research area. Several studies (Seginer, 2007; Reichart and Rappoport, 2010; Ponvert et al., 2011) have considered the problem of inducing parses over raw lexical items rather than part-of-speech (POS) tags. Additional advances have come from more complex models, such as combining CCM and DMV (Klein and Manning, 2004) and modeling large tree fragments (Bod, 2006).

The CCM scores each parse as a product of probabilities of span and context subsequences. It was originally evaluated only on unpunctuated sentences up to length 10 (Klein and Manning, 2002), which account for only 15% of the WSJ corpus; our experiments confirm the observation in (Klein, 2005) that performance degrades dramatically on longer sentences. This problem is unsurprising: CCM scores each constituent type by a single, isolated multinomial parameter.

Our work leverages the idea that sharing information between local probabilities in a structured unsupervised model can lead to substantial accuracy gains, previously demonstrated for dependency grammar induction (Cohen and Smith, 2009; Berg-Kirkpatrick et al., 2010). Our model, Log-Linear CCM (LLCCM), shares information between the probabilities of related constituents by expressing them as a log-linear combination of features trained using the gradient-based learning procedure of Berg-Kirkpatrick et al. (2010). In this way, the probability of generating a constituent is informed by related constituents.

Our model improves unsupervised constituency parsing of sentences longer than 10 words. On sentences of up to length 40 (96% of all sentences in the Penn Treebank), LLCCM outperforms CCM by 13.9% (unlabeled) bracketing F1 and, unlike CCM, outperforms a right-branching baseline on sentences longer than 15 words.

2 Model

The CCM is a generative model for the unsupervised induction of binary constituency parses over sequences of part-of-speech (POS) tags (Klein and Manning, 2002). Conditioned on the constituency or distituency of each span in the parse, CCM generates both the complete sequence of terminals it contains and the terminals in the surrounding context.

Formally, the CCM is a probabilistic model that jointly generates a sentence, s, and a bracketing, B, specifying whether each contiguous subsequence is a constituent or not, in which case the span is called a distituent. Each subsequence of POS tags, or SPAN, α, occurs in a CONTEXT, β, which is an ordered pair of preceding and following tags. A bracketing is a boolean matrix B, indicating which spans (i, j) are constituents ($\mathrm{B}_{ij} = true$) and which are distituents ($\mathrm{B}_{ij} = false$). A bracketing is considered legal if its constituents are nested and form a binary tree $T(\mathrm{B})$.

The joint distribution is given by:

$$\mathrm{P}(s, \mathrm{B}) = \mathrm{P}_{\mathrm{T}}(\mathrm{B}) \cdot \prod_{i,j \in T(\mathrm{B})} \mathrm{P}_{\mathrm{S}}(\alpha(i,j,s)|true)\, \mathrm{P}_{\mathrm{C}}(\beta(i,j,s)|true) \cdot \prod_{i,j \notin T(B)} \mathrm{P}_{\mathrm{S}}(\alpha(i,j,s)|false)\, \mathrm{P}_{\mathrm{C}}(\beta(i,j,s)|false)$$

The prior over unobserved bracketings $\mathrm{P}_{\mathrm{T}}(\mathrm{B})$ is fixed to be the uniform distribution over all legal bracketings. The other distributions, $\mathrm{P}_{\mathrm{S}}(\cdot)$ and $\mathrm{P}_{\mathrm{C}}(\cdot)$, are multinomials whose isolated parameters are estimated to maximize the likelihood of a set of observed sentences $\{s_n\}$ using EM (Dempster et al., 1977).[1]

2.1 The Log-Linear CCM

A fundamental limitation of the CCM is that it contains a single isolated parameter for every span. The number of different possible span types increases exponentially in span length, leading to data sparsity as the sentence length increases.

[1] As mentioned in (Klein and Manning, 2002), the CCM model is deficient because it assigns probability mass to yields and spans that cannot consistently combine to form a valid sentence. Our model does not address this issue, and hence it is similarly deficient.

The Log-Linear CCM (LLCCM) reparameterizes the distributions in the CCM using intuitive features to address the limitations of CCM while retaining its predictive power. The set of proposed features includes a BASIC feature for each parameter of the original CCM, enabling the LLCCM to retain the full expressive power of the CCM. In addition, LLCCM contains a set of coarse features that activate across distinct spans.

To introduce features into the CCM, we express each of its local conditional distributions as a multiclass logistic regression model. Each local distribution, $\mathrm{P}_t(y|x)$ for $t \in \{\text{SPAN}, \text{CONTEXT}\}$, conditions on label $x \in \{true, false\}$ and generates an event (span or context) y. We can define each local distribution in terms of a weight vector, $\mathbf{w}$, and feature vector, $\mathbf{f}_{xyt}$, using a log-linear model:

$$\mathrm{P}_t(y|x) = \frac{\exp\langle \mathbf{w}, \mathbf{f}_{xyt} \rangle}{\sum_{y'} \exp\langle \mathbf{w}, \mathbf{f}_{xy't} \rangle} \quad (1)$$

This technique for parameter transformation was shown to be effective in unsupervised models for part-of-speech induction, dependency grammar induction, word alignment, and word segmentation (Berg-Kirkpatrick et al., 2010). In our case, replacing multinomials via featurized models not only improves model accuracy, but also lets the model apply effectively to a new regime of long sentences.

2.2 Feature Templates

In the SPAN model, for each span $y = [\alpha_1, \ldots, \alpha_n]$ and label x, we use the following feature templates:

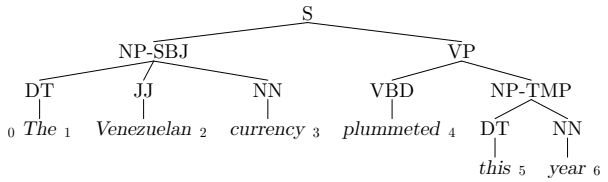
BASIC:	$\mathbb{I}[y = \cdot \wedge x = \cdot]$
BOUNDARY:	$\mathbb{I}[\alpha_1 = \cdot \wedge \alpha_n = \cdot \wedge x = \cdot]$
PREFIX:	$\mathbb{I}[\alpha_1 = \cdot \wedge x = \cdot]$
SUFFIX:	$\mathbb{I}[\alpha_n = \cdot \wedge x = \cdot]$

Just as the external CONTEXT is a signal of constituency, so too is the internal "context." For example, there are many distinct noun phrases with different spans that all begin with DT and end with NN; a fact expressed by the BOUNDARY feature (Table 1).

In the CONTEXT model, for each context $y = [\beta_1, \beta_2]$ and constituent/distituent decision x, we use the following feature templates:

BASIC:	$\mathbb{I}[y = \cdot \wedge x = \cdot]$
L-CONTEXT:	$\mathbb{I}[\beta_1 = \cdot \wedge x = \cdot]$
R-CONTEXT:	$\mathbb{I}[\beta_2 = \cdot \wedge x = \cdot]$

Consider the following example extracted from the WSJ:



Both spans (0, 3) and (4, 6) are constituents corresponding to noun phrases whose features are shown in Table 1:

	Feature Name	(0,3)	(4, 6)
span	BASIC-DT-JJ-NN:	1	0
	BASIC-DT-NN:	0	1
	BOUNDARY-DT-NN:	1	1
	PREFIX-DT:	1	1
	SUFFIX-NN:	1	1
context	BASIC-◇-VBD:	1	0
	BASIC-VBD-◇:	0	1
	L-CONTEXT-◇:	1	0
	L-CONTEXT-VBD:	0	1
	R-CONTEXT-VBD:	1	0
	R-CONTEXT-◇:	0	1

Table 1: Span and context features for constituent spans (0, 3) and (4, 6). The symbol ◇ indicates a sentence boundary.

Notice that although the BASIC span features are active for at most one span, the remaining features fire for both spans, effectively sharing information between the local probabilities of these events.

The coarser CONTEXT features factor the context pair into its components, which allow the LLCCM to more easily learn, for example, that a constituent is unlikely to immediately follow a determiner.

3 Training

In the EM algorithm for estimating CCM parameters, the E-Step computes posteriors over bracketings using the Inside-Outside algorithm. The M-Step chooses parameters that maximize the expected complete log likelihood of the data.

The weights, $\mathbf{w}$, of LLCCM are estimated to maximize the data log likelihood of the training sentences $\{s_n\}$, summing out all possible bracketings B for each sentence:

$$L(\mathbf{w}) = \sum_{s_n} \log \sum_{B} \mathrm{P}_{\mathbf{w}}(s_n, B)$$

We optimize this objective via L-BFGS (Liu and Nocedal, 1989), which requires us to compute the objective gradient. Berg-Kirkpatrick et al. (2010) showed that the data log likelihood gradient is equivalent to the gradient of the expected complete log likelihood (the objective maximized in the M-step of EM) at the point from which expectations are computed. This gradient can be computed in three steps.

First, we compute the local probabilities of the CCM, $\mathrm{P}_t(y|x)$, from the current $\mathbf{w}$ using Equation (1). We approximate the normalization over an exponential number of terms by only summing over spans that appeared in the training corpus.

Second, we compute posteriors over bracketings, $\mathrm{P}(i, j|s_n)$, just as in the E-step of CCM training,[2] in order to determine the expected counts:

$$e_{xy,\textsc{Span}} = \sum_{s_n} \sum_{ij} \mathbb{I}\left[\alpha(i, j, s_n) = y\right] \delta(x)$$

$$e_{xy,\textsc{Context}} = \sum_{s_n} \sum_{ij} \mathbb{I}\left[\beta(i, j, s_n) = y\right] \delta(x)$$

where $\delta(true) = \mathrm{P}(i, j|s_n)$, and $\delta(false) = 1 - \delta(true)$.

We summarize these expected count quantities as:

$$e_{xyt} = \begin{cases} e_{xy,\textsc{Span}} & \text{if } t = \textsc{Span} \\ e_{xy,\textsc{Context}} & \text{if } t = \textsc{Context} \end{cases}$$

Finally, we compute the gradient with respect to $\mathbf{w}$, expressed in terms of these expected counts and conditional probabilities:

$$\nabla L(\mathbf{w}) = \sum_{xyt} e_{xyt} \mathbf{f}_{xyt} - G(\mathbf{w})$$

$$G(\mathbf{w}) = \sum_{xt} \left(\sum_{y} e_{xyt} \right) \sum_{y'} \mathrm{P}_t(y|x) \mathbf{f}_{xy't}$$

Following (Klein and Manning, 2002), we initialize the model weights by optimizing against posterior probabilities fixed to the split-uniform distribution, which generates binary trees by randomly choosing a split point and recursing on each side of the split.[3]

[2] We follow the dynamic program presented in Appendix A.1 of (Klein, 2005).

[3] In Appendix B.2, Klein (2005) shows this posterior can be expressed in closed form. As in previous work, we start the initialization optimization with the zero vector, and terminate after 10 iterations to regularize against achieving a local maximum.

3.1 Efficiently Computing the Gradient

The following quantity appears in $G(\mathbf{w})$:

$$\gamma_t(x) = \sum_y e_{xyt}$$

Which expands as follows depending on t:

$$\gamma_{\text{SPAN}}(x) = \sum_y \sum_{s_n} \sum_{ij} \mathbb{I}\left[\alpha(i,j,s_n) = y\right] \delta(x)$$

$$\gamma_{\text{CONTEXT}}(x) = \sum_y \sum_{s_n} \sum_{ij} \mathbb{I}\left[\beta(i,j,s_n) = y\right] \delta(x)$$

In each of these expressions, the $\delta(x)$ term can be factored outside the sum over y. Each fixed (i,j) and s_n pair has exactly one span and context, hence the quantities $\sum_y \mathbb{I}\left[\alpha(i,j,s_n) = y\right]$ and $\sum_y \mathbb{I}\left[\beta(i,j,s_n) = y\right]$ are both equal to 1.

$$\gamma_t(x) = \sum_{s_n} \sum_{ij} \delta(x)$$

This expression further simplifies to a constant. The sum of the posterior probabilities, $\delta(true)$, over all positions is equal to the total number of constituents in the tree. Any binary tree over N terminals contains exactly $2N-1$ constituents and $\frac{1}{2}(N-2)(N-1)$ distituents.

$$\gamma_t(x) = \begin{cases} \sum_{s_n} (2|s_n| - 1) & \text{if } x = true \\ \frac{1}{2}\sum_{s_n} (|s_n| - 2)(|s_n| - 1) & \text{if } x = false \end{cases}$$

where $|s_n|$ denotes the length of sentence s_n.

Thus, $G(\mathbf{w})$ can be precomputed once for the entire dataset at each minimization step. Moreover, $\gamma_t(x)$ can be precomputed once before all iterations.

3.2 Relationship to Smoothing

The original CCM uses additive smoothing in its M-step to capture the fact that distituents outnumber constituents. For each span or context, CCM adds 10 counts: 2 as a constituent and 8 as a distituent.[4] We note that these smoothing parameters are tailored to short sentences: in a binary tree, the number of constituents grows linearly with sentence length, whereas the number of distituents grows quadratically. Therefore, the ratio of constituents to distituents is not constant across sentence lengths. In contrast, by virtue of the log-linear model, LLCCM assigns positive probability to all spans or contexts without explicit smoothing.

[4]These counts are specified in (Klein, 2005); Klein and Manning (2002) added 10 constituent and 50 distituent counts.

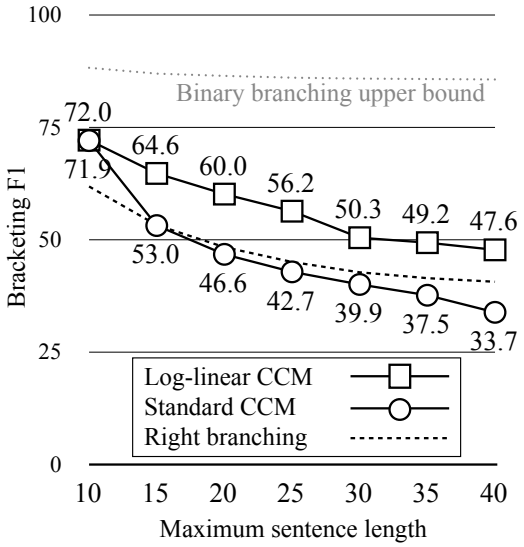


Figure 1: CCM and LLCCM trained and tested on sentences of a fixed length. LLCCM performs well on longer sentences. The binary branching upper bound correponds to UBOUND from (Klein and Manning, 2002).

4 Experiments

We train our models on gold POS sequences from all sections (0-24) of the WSJ (Marcus et al., 1993) with punctuation removed. We report bracketing F1 scores between the binary trees predicted by the models on these sequences and the treebank parses.

We train and evaluate both a CCM implementation (Luque, 2011) and our LLCCM on sentences up to a fixed length n, for $n \in \{10, 15, \ldots, 40\}$. Figure 1 shows that LLCCM substantially outperforms the CCM on longer sentences. After length 15, CCM accuracy falls below the right branching baseline, whereas LLCCM remains significantly better than right-branching through length 40.

5 Conclusion

Our log-linear variant of the CCM extends robustly to long sentences, enabling constituent grammar induction to be used in settings that typically include long sentences, such as machine translation reordering (Chiang, 2005; DeNero and Uszkoreit, 2011; Dyer et al., 2011).

Acknowledgments

We thank Taylor Berg-Kirkpatrick and Dan Klein for helpful discussions regarding the work on which this paper is based. This work was partially supported by the National Science Foundation through a Graduate Research Fellowship to the first author.

References

Taylor Berg-Kirkpatrick and Dan Klein. 2010. Phylogenetic grammar induction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1288–1297, Uppsala, Sweden, July. Association for Computational Linguistics.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590, Los Angeles, California, June. Association for Computational Linguistics.

Rens Bod. 2006. Unsupervised parsing with U-DOP. In *Proceedings of the Conference on Computational Natural Language Learning*.

Glenn Carroll and Eugene Charniak. 1992. Two experiments on learning probabilistic dependency grammars from corpora. In *Workshop Notes for Statistically-Based NLP Techniques, AAAI*, pages 1–13.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 263–270, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Shay B. Cohen and Noah A. Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 74–82, Boulder, Colorado, June. Association for Computational Linguistics.

Shay B. Cohen, Dipanjan Das, and Noah A. Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 50–61, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Arthur Dempster, Nan Laird, and Donald Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

John DeNero and Jakob Uszkoreit. 2011. Inducing sentence structure from parallel corpora for reordering. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Chris Dyer, Kevin Gimpel, Jonathan H. Clark, and Noah A. Smith. 2011. The CMU-ARK German-English translation system. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 337–343, Edinburgh, Scotland, July. Association for Computational Linguistics.

William P. Headden III, Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 101–109, Boulder, Colorado, June. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics, Main Volume*, pages 478–485, Barcelona, Spain, July.

Dan Klein. 2005. *The Unsupervised Learning of Natural Language Structure*. Ph.D. thesis.

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.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory method for large scale optimization. *Mathematical Programming B*, 45(3):503–528.

Franco Luque. 2011. Una implementación del modelo DMV+CCM para parsing no supervisado. In *2do Workshop Argentino en Procesamiento de Lenguaje Natural*.

Mitchell P. Marcus, Beatrice Santorini, and Mary A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Tahira Naseem and Regina Barzilay. 2011. Using semantic cues to learn syntax. In *AAAI*.

Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1244, Cambridge, MA, October. Association for Computational Linguistics.

Fernando Pereira and Yves Schabes. 1992. Inside-outside reestimation from partially bracketed corpora.

In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Newark, Delaware, USA, June. Association for Computational Linguistics.

Elias Ponvert, Jason Baldridge, and Katrin Erk. 2011. Simple unsupervised grammar induction from raw text with cascaded finite state models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Portland, Oregon, USA, June. Association for Computational Linguistics.

Roi Reichart and Ari Rappoport. 2010. Improved fully unsupervised parsing with zoomed learning. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 684–693, Cambridge, MA, October. Association for Computational Linguistics.

Yoav Seginer. 2007. Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 384–391, Prague, Czech Republic, June. Association for Computational Linguistics.