Three Dependency-and-Boundary Models for Grammar Induction

Valentin I. Spitkovsky
Stanford University and Google Inc.
valentin@cs.stanford.edu

Hiyan Alshawi
Google Inc., Mountain View, CA, 94043
hiyan@google.com

Daniel Jurafsky
Stanford University, Stanford, CA, 94305
jurafsky@stanford.edu

Abstract

We present a new family of models for unsupervised parsing, *Dependency and Boundary* models, that use cues at constituent boundaries to inform head-outward dependency tree generation. We build on three intuitions that are explicit in phrase-structure grammars but only implicit in standard dependency formulations: (i) Distributions of words that occur at sentence boundaries — such as English determiners — resemble constituent edges. (ii) Punctuation at sentence boundaries further helps distinguish full sentences from fragments like headlines and titles, allowing us to model grammatical differences between complete and incomplete sentences. (iii) Sentence-internal punctuation boundaries help with longer-distance dependencies, since punctuation correlates with constituent edges. Our models induce state-of-the-art dependency grammars for many languages without special knowledge of optimal input sentence lengths or biased, manually-tuned initializers.

1 Introduction

Natural language is ripe with all manner of boundaries at the surface level that align with hierarchical syntactic structure. From the significance of function words (Berant et al., 2006) and punctuation marks (Seginer, 2007; Ponvert et al., 2010) as separators between constituents in longer sentences — to the importance of isolated words in children's early vocabulary acquisition (Brent and Siskind, 2001) — word boundaries play a crucial role in language learning. We will show that boundary information can also be useful in dependency grammar induction models, which traditionally focus on head rather than fringe words (Carroll and Charniak, 1992).

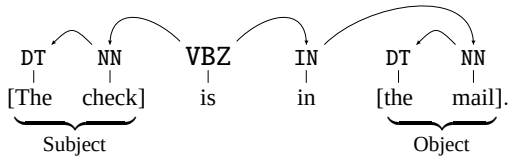

Figure 1: A partial analysis of our running example.

Consider the example in Figure 1. Because the determiner (DT) appears at the left edge of the sentence, it should be possible to learn that determiners may generally be present at left edges of phrases. This information could then be used to correctly parse the sentence-internal determiner in *the mail*. Similarly, the fact that the noun head (NN) of the object *the mail* appears at the right edge of the sentence could help identify the noun *check* as the right edge of the subject NP. As with jigsaw puzzles, working inwards from boundaries helps determine sentence-internal structures of both noun phrases, neither of which would be quite so clear if viewed separately.

Furthermore, properties of noun-phrase edges are partially shared with prepositional- and verb-phrase units that contain these nouns. Because typical head-driven grammars model valence separately for each class of head, however, they cannot see that the left fringe boundary, *The check*, of the verb-phrase is shared with its daughter's, *check*. Neither of these insights is available to traditional dependency formulations, which could learn from the boundaries of this sentence only that determiners might have no left- and that nouns might have no right-dependents.

We propose a family of dependency parsing models that are capable of inducing longer-range implications from sentence edges than just fertilities of their fringe words. Our ideas conveniently lend themselves to implementations that can reuse much of the standard grammar induction machinery, including efficient dynamic programming routines for the relevant expectation-maximization algorithms.

2 The Dependency and Boundary Models

Our models follow a standard generative story for head-outward automata (Alshawi, 1996a), restricted to the split-head case (see below),[1] over lexical word classes $\{c_w\}$: first, a sentence root c_r is chosen, with probability $\mathbb{P}_{\texttt{ATTACH}}(c_r \mid \diamond;\ \texttt{L})$; $\diamond$ is a special start symbol that, by convention (Klein and Manning, 2004; Eisner, 1996), produces exactly one child, to its left. Next, the process recurses. Each (head) word c_h generates a left-dependent with probability $1 - \mathbb{P}_{\texttt{STOP}}(\ \cdot \mid \texttt{L};\ \cdots)$, where dots represent additional parameterization on which it may be conditioned. If the child is indeed generated, its identity c_d is chosen with probability $\mathbb{P}_{\texttt{ATTACH}}(c_d \mid c_h;\ \cdots)$, influenced by the identity of the parent c_h and possibly other parameters (again represented by dots). The child then generates its own subtree recursively and the whole process continues, moving away from the head, until c_h fails to generate a left-dependent. At that point, an analogous procedure is repeated to c_h's right, this time using stopping factors $\mathbb{P}_{\texttt{STOP}}(\ \cdot \mid \texttt{R};\ \cdots)$. All parse trees derived in this way are guaranteed to be projective and can be described by split-head grammars.

Instances of these split-head automata have been heavily used in grammar induction (Paskin, 2001b; Klein and Manning, 2004; Headden et al., 2009, *inter alia*), in part because they allow for efficient implementations (Eisner and Satta, 1999, §8) of the inside-outside re-estimation algorithm (Baker, 1979). The basic tenet of split-head grammars is that every head word generates its left-dependents independently of its right-dependents. This assumption implies, for instance, that words' left- and right-valences — their numbers of children to each side — are also independent. But it does *not* imply that descendants that are closer to the head cannot influence the generation of farther dependents on the same side. Nevertheless, many popular grammars for unsupervised parsing behave as if a word had to generate all of its children (to one side) — or at least their count — *before* allowing any of these children themselves to recurse.

For example, Klein and Manning's (2004) dependency model with valence (DMV) could be implemented as both head-outward and head-inward automata. (In fact, arbitrary permutations of siblings to a given side of their parent would not affect the likelihood of the modified tree, with the DMV.) We propose to make fuller use of split-head automata's head-outward nature by drawing on information in partially-generated parses, which contain useful predictors that, until now, had not been exploited even in featurized systems for grammar induction (Cohen and Smith, 2009; Berg-Kirkpatrick et al., 2010).

Some of these predictors, including the identity — or even number (McClosky, 2008) — of already-generated siblings, can be prohibitively expensive in sentences above a short length k. For example, they break certain modularity constraints imposed by the charts used in $O(k^3)$-optimized algorithms (Paskin, 2001a; Eisner, 2000). However, in bottom-up parsing and training from text, everything about the yield — i.e., the ordered sequence of all already-generated descendants, on the side of the head that is in the process of spawning off an additional child — is not only known but also readily accessible. Taking advantage of this availability, we designed three new models for dependency grammar induction.

2.1 Dependency and Boundary Model One

DBM-1 conditions all stopping decisions on adjacency and the identity of the fringe word c_e — the currently-farthest descendant (edge) derived by head c_h in the given head-outward direction ($dir \in \{\texttt{L}, \texttt{R}\}$):

$$\mathbb{P}_{\texttt{STOP}}(\ \cdot \mid dir;\ adj, c_e).$$

In the adjacent case ($adj = \texttt{T}$), c_h is deciding whether to have any children on a given side: a first child's subtree would be right next to the head, so the head and the fringe words coincide ($c_h = c_e$). In the non-adjacent case ($adj = \texttt{F}$), these will be different words and their classes will, in general, not be the same.[2] Thus, non-adjacent stopping decisions will be made independently of a head word's identity. Therefore, all word classes will be equally likely to continue to grow or not, for a specific proposed fringe boundary.

For example, production of *The check is* involves two non-adjacent stopping decisions on the left: one by the noun *check* and one by the verb *is*, both of which stop after generating a first child. In DBM-1,

[1] Unrestricted head-outward automata are strictly more powerful (e.g., they recognize the language $a^n b^n$ in finite state) than the split-head variants, which process one side before the other.

[2] Fringe words differ also from other standard dependency features (Eisner, 1996, §2.3): parse siblings and adjacent words.

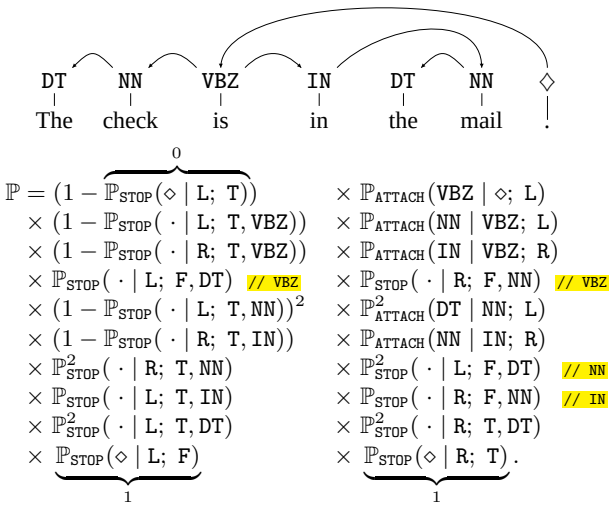

Figure 2: Our running example — a simple sentence and its unlabeled dependency parse structure's probability, as factored by DBM-1; highlighted comments specify heads associated to non-adjacent stopping probability factors.

this outcome is captured by squaring a shared parameter belonging to the left-fringe determiner *The*: $\mathbb{P}_{\text{STOP}}(\ \cdot\ \mid \text{L};\ \text{F}, \text{DT})^2$ — instead of by a product of two factors, such as $\mathbb{P}_{\text{STOP}}(\ \cdot\ \mid \text{L};\ \text{F}, \text{NN}) \cdot \mathbb{P}_{\text{STOP}}(\ \cdot\ \mid \text{L};\ \text{F}, \text{VBZ})$.

In these grammars, dependents' attachment probabilities, given heads, are additionally conditioned only on their relative positions — as in traditional models (Klein and Manning, 2004; Paskin, 2001b):

$$\mathbb{P}_{\text{ATTACH}}(c_d \mid c_h;\ dir).$$

Figure 2 shows a completely factored example.

2.2 Dependency and Boundary Model Two

DBM-2 allows different but related grammars to coexist in a single model. Specifically, we presuppose that all sentences are assigned to one of two classes: complete and incomplete ($comp \in \{\text{T}, \text{F}\}$, for now taken as exogenous). This model assumes that word-word (i.e., head-dependent) interactions in the two domains are the same. However, sentence lengths — for which stopping probabilities are responsible — and distributions of root words may be different.

Consequently, an additional $comp$ parameter is added to the context of two relevant types of factors:

$$\mathbb{P}_{\text{STOP}}(\ \cdot\ \mid dir;\ adj, c_e, \mathit{comp});$$
$$\text{and } \mathbb{P}_{\text{ATTACH}}(c_r \mid \diamond;\ \text{L}, \mathit{comp}).$$

For example, the new stopping factors could capture the fact that incomplete fragments — such as the noun-phrases *George Morton*, headlines *Energy* and *Odds and Ends*, a line item *c - Domestic car*, dollar quantity *Revenue: $3.57 billion*, the time *1:11am*, and the like — tend to be much shorter than complete sentences. The new root-attachment factors could further track that incomplete sentences generally lack verbs, in contrast to other short sentences, e.g., *Excerpts follow:*, *Are you kidding?*, *Yes, he did.*, *It's huge.*, *Indeed it is.*, *I said, 'NOW?'*, *"Absolutely," he said.*, *I am waiting.*, *Mrs. Yeargin declined.*, *McGraw-Hill was outraged.*, *"It happens."*, *I'm OK, Jack.*, *Who cares?*, *Never mind.* and so on.

All other attachment probabilities $\mathbb{P}_{\text{ATTACH}}(c_d \mid c_h;\ dir)$ remain unchanged, as in DBM-1. In practice, $comp$ can indicate presence of sentence-final punctuation.

2.3 Dependency and Boundary Model Three

DBM-3 adds further conditioning on punctuation context. We introduce another boolean parameter, $cross$, which indicates the presence of intervening punctuation between a proposed head word c_h and its dependent c_d. Using this information, longer-distance punctuation-crossing arcs can be modeled separately from other, lower-level dependencies, via

$$\mathbb{P}_{\text{ATTACH}}(c_d \mid c_h;\ dir, \mathit{cross}).$$

For instance, in *Continentals believe that the strongest growth area will be southern Europe.*, four words appear between *that* and *will*. Conditioning on (the absence of) intervening punctuation could help tell true long-distance relations from impostors.

All other probabilities, $\mathbb{P}_{\text{STOP}}(\ \cdot\ \mid dir;\ adj, c_e, comp)$ and $\mathbb{P}_{\text{ATTACH}}(c_r \mid \diamond;\ \text{L}, comp)$, remain the same as in DBM-2.

2.4 Summary of DBMs and Related Models

Head-outward automata (Alshawi, 1996a; Alshawi, 1996b; Alshawi et al., 2000) played a central part as generative models for probabilistic grammars, starting with their early adoption in supervised split-head constituent parsers (Collins, 1997; Collins, 2003). Table 1 lists some parameterizations that have since been used by unsupervised dependency grammar inducers sharing their backbone split-head process.

3 Experimental Set-Up and Methodology

We first motivate each model by analyzing the Wall Street Journal (WSJ) portion of the Penn English Treebank (Marcus et al., 1993),[3] before delving into

[3]We converted labeled constituents into unlabeled dependencies using deterministic "head-percolation" rules (Collins,

Split-Head Dependency Grammar		$\mathbb{P}_{\texttt{ATTACH}}$ *(head-root)*	$\mathbb{P}_{\texttt{ATTACH}}$ *(dependent-head)*	$\mathbb{P}_{\texttt{STOP}}$ *(adjacent and not)*
GB	(Paskin, 2001b)	$1 \;/\; \lvert\{w\}\rvert$	$d \mid h;\ dir$	$1 \;/\; 2$
DMV	(Klein and Manning, 2004)	$c_r \mid \diamond;\ \texttt{L}$	$c_d \mid c_h;\ dir$	$\cdot \mid dir;\ adj,\ c_h$
EVG	(Headden et al., 2009)	$c_r \mid \diamond;\ \texttt{L}$	$c_d \mid c_h;\ dir,\ adj$	$\cdot \mid dir;\ adj,\ c_h$
DBM-1	(§2.1)	$c_r \mid \diamond;\ \texttt{L}$	$c_d \mid c_h;\ dir$	$\cdot \mid dir;\ adj,\ c_e$
DBM-2	(§2.2)	$c_r \mid \diamond;\ \texttt{L},\ comp$	$c_d \mid c_h;\ dir$	$\cdot \mid dir;\ adj,\ c_e,\ comp$
DBM-3	(§2.3)	$c_r \mid \diamond;\ \texttt{L},\ comp$	$c_d \mid c_h;\ dir,\ cross$	$\cdot \mid dir;\ adj,\ c_e,\ comp$

Table 1: Parameterizations of the split-head-outward generative process used by DBMs and in previous models.

grammar induction experiments. Although motivating solely from this treebank biases our discussion towards a very specific genre of just one language, it has the advantage of allowing us to make concrete claims that are backed up by significant statistics.

In the grammar induction experiments that follow, we will test each model's incremental contribution to accuracies empirically, across many disparate languages. We worked with all 23 (disjoint) train/test splits from the 2006/7 CoNLL shared tasks (Buchholz and Marsi, 2006; Nivre et al., 2007), spanning 19 languages.[4] For each data set, we induced a baseline grammar using the DMV. We excluded all training sentences with more than 15 tokens to create a conservative bias, because in this set-up the baseline is known to excel (Spitkovsky et al., 2009). Grammar inducers were initialized using (the same) uniformly-at-random chosen parse trees of training sentences (Cohen and Smith, 2010); thereafter, we applied "add one" smoothing at every training step.

To fairly compare the models under consideration — which could have quite different starting perplexities and ensuing consecutive relative likelihoods — we experimented with two termination strategies. In one case, we blindly ran each learner through 40 steps of inside-outside re-estimation, ignoring any convergence criteria; in the other case, we ran until numerical convergence of soft EM's objective function or until the likelihood of resulting Viterbi parse trees suffered — an "early-stopping lateen EM" strategy (Spitkovsky et al., 2011a, §2.3). We evaluated against all sentences of the blind test sets (except one 145-token item in Arabic '07 data).

Table 2 shows experimental results, averaged over all 19 languages, for the DMV baselines and DBM-1 and 2. We did not test DBM-3 in this set-up because most sentence-internal punctuation occurs in longer sentences; instead, DBM-3 will be tested later (see §7), using most sentences,[5] in the final training step of a curriculum strategy (Bengio et al., 2009) that we will propose for DBMs. For the three models tested on shorter inputs (up to 15 tokens) both terminating criteria exhibited the same trend; lateen EM consistently scored slightly higher than 40 EM iterations.

Termination Criterion	DMV	DBM-1	DBM-2
40 steps of EM	33.5	38.8	40.7
early-stopping lateen EM	34.0	39.0	**40.9**

Table 2: Directed dependency accuracies, averaged over all 2006/7 CoNLL evaluation sets (all sentences), for the DMV and two new dependency-and-boundary grammar inducers (DBM-1,2) — using two termination strategies.[6]

4 Dependency and Boundary Model One

The primary difference between DBM-1 and traditional models, such as the DMV, is that DBM-1 conditions non-adjacent stopping decisions on the identities of fringe words in partial yields (see §2.1).

4.1 Analytical Motivation

Treebank data suggests that the class of the fringe word — its part-of-speech, c_e — is a better predictor of (non-adjacent) stopping decisions, in a given direction dir, than the head's own class c_h. A statistical analysis of logistic regressions fitted to the data shows that the (c_h, dir) predictor explains only about 7% of the total variation (see Table 3). This seems low, although it is much better compared to direction alone (which explains less than 2%) and slightly better than using the (current) number of the head's de-

1999), discarding any empty nodes, etc., as is standard practice.

[4] We did not test on WSJ data because such evaluation would not be blind, as parse trees from the PTB are our motivating examples; instead, performance on WSJ serves as a strong baseline in a separate study (Spitkovsky et al., 2012a): bootstrapping of DBMs from mostly incomplete inter-punctuation fragments.

[5] Results for DBM-3 — given only standard input sentences, up to length fifteen — would be nearly identical to DBM-2's.

[6] We down-weighed the four languages appearing in both CoNLL years (see Table 8) by 50% in all reported averages.

Non-Adjacent Stop Predictor	R^2_{adj}	AIC_c
(dir)	0.0149	1,120,200
(n, dir)	0.0726	1,049,175
(c_h, dir)	0.0728	1,047,157
(c_e, dir)	0.2361	904,102.4
(c_h, c_e, dir)	0.3320	789,594.3

Table 3: Coefficients of determination (R^2) and Akaike information criteria (AIC), both adjusted for the number of parameters, for several single-predictor logistic models of non-adjacent stops, given direction dir; c_h is the class of the head, n is its number of descendants (so far) to that side, and c_e represents the farthest descendant (the edge).

scendants on that side, n, instead of the head's class. In contrast, using c_e in place of c_h boosts explanatory power to 24%, keeping the number of parameters the same. If one were willing to roughly square the size of the model, explanatory power could be improved further, to 33% (see Table 3), using both c_e and c_h.

Fringe boundaries thus appear to be informative even in the supervised case, which is not surprising, since using just one probability factor (and its complement) to generate very short (geometric coin-flip) sequences is a recipe for high entropy. But as suggested earlier, fringes should be extra attractive in unsupervised settings because yields are observable, whereas heads almost always remain hidden. Moreover, every sentence exposes two true edges (Hänig, 2010): integrated over many sample sentence beginnings and ends, cumulative knowledge about such markers can guide a grammar inducer inside long inputs, where structure is murky. Table 4 shows distributions of all part-of-speech (POS) tags in the treebank versus in sentence-initial, sentence-final and sentence-root positions. WSJ often leads with determiners, proper nouns, prepositions and pronouns — all good candidates for starting English phrases; and its sentences usually end with various noun types, again consistent with our running example.

4.2 Experimental Results

Table 2 shows DBM-1 to be substantially more accurate than the DMV, on average: 38.8 versus 33.5% after 40 steps of EM.[7] Lateen termination improved both models' accuracies slightly, to 39.0 and 34.0%, respectively, with DBM-1 scoring five points higher.

[7]DBM-1's 39% average accuracy with uniform-at-random initialization is two points above DMV's scores with the "ad-hoc harmonic" strategy, 37% (Spitkovsky et al., 2011a, Table 5).

POS	*% of All Tokens*	*First Tokens*	*Last Tokens*	*Sent. Roots*	*Frag. Roots*
NN	15.94	4.31	36.67	0.10	23.40
IN	11.85	13.54	0.57	0.24	4.33
NNP	11.09	20.49	12.85	0.02	32.02
DT	9.84	23.34	0.34	0.00	0.04
JJ	7.32	4.33	3.74	0.01	1.15
NNS	7.19	4.49	20.64	0.15	17.12
CD	4.37	1.29	6.92	0.00	3.27
RB	3.71	5.96	3.88	0.00	1.50
VBD	3.65	0.09	3.52	46.65	0.93
VB	3.17	0.44	1.67	0.48	6.81
CC	2.86	5.93	0.00	0.00	0.00
TO	2.67	0.37	0.05	0.02	0.44
VBZ	2.57	0.17	1.65	28.31	0.93
VBN	2.42	0.61	2.57	0.65	1.28
PRP	2.08	9.04	1.34	0.00	0.00
VBG	1.77	1.26	0.64	0.10	0.97
VBP	1.50	0.05	0.61	14.33	0.71
MD	1.17	0.07	0.05	8.88	0.57
POS	1.05	0.00	0.11	0.01	0.04
PRP$	1.00	0.90	0.00	0.00	0.00
WDT	0.52	0.08	0.00	0.01	0.13
JJR	0.39	0.18	0.43	0.00	0.09
RP	0.32	0.00	0.42	0.00	0.00
NNPS	0.30	0.20	0.56	0.00	2.96
WP	0.28	0.42	0.01	0.01	0.04
WRB	0.26	0.78	0.02	0.01	0.31
JJS	0.23	0.27	0.06	0.00	0.00
RBR	0.21	0.20	0.54	0.00	0.04
EX	0.10	0.75	0.00	0.00	0.00
RBS	0.05	0.06	0.01	0.00	0.00
PDT	0.04	0.08	0.00	0.00	0.00
FW	0.03	0.01	0.05	0.00	0.09
WP$	0.02	0.00	0.00	0.00	0.00
UH	0.01	0.08	0.05	0.00	0.62
SYM	0.01	0.11	0.01	0.00	0.18
LS	0.01	0.09	0.00	0.00	0.00

Table 4: Empirical distributions for non-punctuation part-of-speech tags in WSJ, ordered by overall frequency, as well as distributions for sentence boundaries and for the roots of complete and incomplete sentences. (A uniform distribution would have $1/36 = 2.\overline{7}\%$ for all POS-tags.)

$\sqrt{1 - \sum_x \sqrt{p_x q_x}}$	*All*	*First*	*Last*	*Sent.*	*Frag.*
Uniform	0.48	0.58	0.64	0.79	0.65
All		0.35	0.40	0.79	0.42
First			0.59	**0.94**	0.57
Last				**0.83**	0.29
Sent.					**0.86**

Table 5: A distance matrix for all pairs of probability distributions over POS-tags shown in Table 4 and the uniform distribution; the BC- (or Hellinger) distance (Bhattacharyya, 1943; Nikulin, 2002) between discrete distributions p and q (over $x \in \mathcal{X}$) ranges from zero (iff $p = q$) to one (iff $p \cdot q = 0$, i.e., when they do not overlap at all).

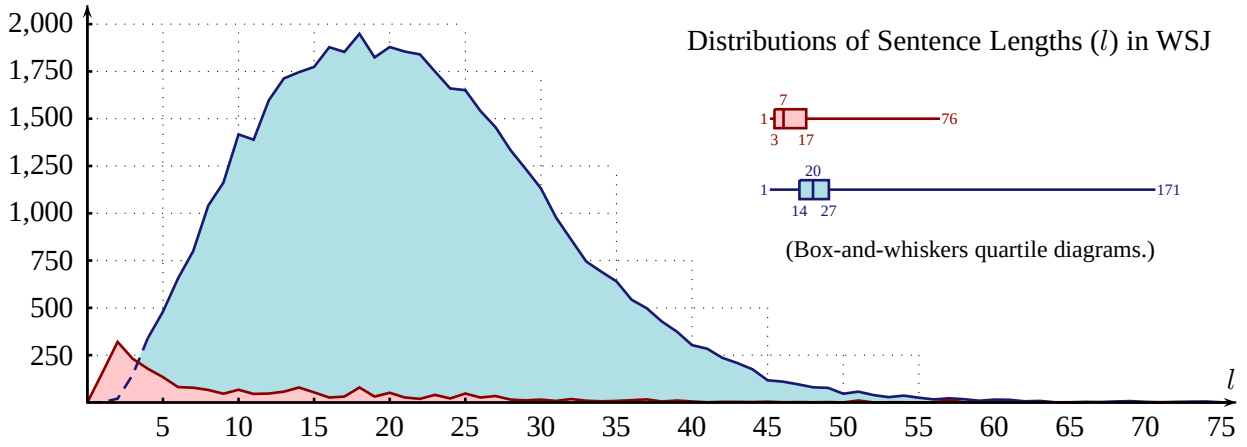

Figure 3: Histograms of lengths (in tokens) for 2,261 non-clausal fragments (red) and other sentences (blue) in WSJ.

5 Dependency and Boundary Model Two

DBM-2 adapts DBM-1 grammars to two classes of inputs (complete sentences and incomplete fragments) by forking off new, separate multinomials for stopping decisions and root-distributions (see §2.2).

5.1 Analytical Motivation

Unrepresentative short sentences — such as headlines and titles — are common in news-style data and pose a known nuisance to grammar inducers. Previous research sometimes took radical measures to combat the problem: for example, Gillenwater et al. (2009) excluded all sentences with three or fewer tokens from their experiments; and Mareček and Zabokrtský (2011) enforced an "anti-noun-root" policy to steer their Gibbs sampler away from the undercurrents caused by the many short noun-phrase fragments (among sentences up to length 15, in Czech data). We refer to such snippets of text as "incomplete sentences" and focus our study of WSJ on non-clausal data (as signaled by top-level constituent annotations whose first character is not S).[8]

Table 4 shows that roots of incomplete sentences, which are dominated by nouns, barely resemble the other roots, drawn from more traditional verb and modal types. In fact, these two empirical root distributions are more distant from one another than either is from the uniform distribution, in the space of discrete probability distributions over POS-tags (see Table 5). Of the distributions we considered, only sentence boundaries are as or more different from (complete) roots, suggesting that heads of fragments too may warrant their own multinomial in the model.

Further, incomplete sentences are uncharacteristically short (see Figure 3). It is this property that makes them particularly treacherous to grammar inducers, since by offering few options of root positions they increase the chances that a learner will incorrectly induce nouns to be heads. Given that expected lengths are directly related to stopping decisions, it could make sense to also model the stopping probabilities of incomplete sentences separately.

5.2 Experimental Results

Since it is not possible to consult parse trees during grammar induction (to check whether an input sentence is clausal), we opted for a proxy: presence of sentence-final punctuation. Using punctuation to divide input sentences into two groups, DBM-2 scored higher: 40.9, up from 39.0% accuracy (see Table 2).

After evaluating these multi-lingual experiments, we checked how well our proxy corresponds to actual clausal sentences in WSJ. Table 6 shows the binary confusion matrix having a fairly low (but positive) Pearson correlation coefficient. False positives

$r_\phi \approx 0.31$	*Clausal*	*non-Clausal*	*Total*
Punctuation	46,829	1,936	48,765
no Punctuation	118	325	443
Total	46,947	2,261	49,208

Table 6: A contingency table for clausal sentences and trailing punctuation in WSJ; the mean square contingency coefficient r_ϕ signifies a low degree of correlation. (For two binary variables, r_ϕ is equivalent to Karl Pearson's better-known product-moment correlation coefficient, ρ.)

[8] I.e., separating top-level types {S, SINV, SBARQ, SQ, SBAR} from the rest (ordered by frequency): {NP, FRAG, X, PP, ...}.

include parenthesized expressions that are marked as noun-phrases, such as *(See related story: "Fed Ready to Inject Big Funds": WSJ Oct. 16, 1989)*; false negatives can be headlines having a main verb, e.g., *Population Drain Ends For Midwestern States*. Thus, our proxy is not perfect but seems to be tolerable in practice. We suspect that identities of punctuation marks (Collins, 2003, Footnote 13) — both sentence-final and sentence-initial — could be of extra assistance in grammar induction, specifically for grouping imperatives, questions, and so forth.

6 Dependency and Boundary Model Three

DBM-3 exploits sentence-internal punctuation contexts by modeling punctuation-crossing dependency arcs separately from other attachments (see §2.3).

6.1 Analytical Motivation

Many common syntactic relations, such as between a determiner and a noun, are unlikely to hold over long distances. (In fact, 45% of all head-percolated dependencies in WSJ are between adjacent words.) However, some common constructions are more remote: e.g., subordinating conjunctions are, on average, 4.8 tokens away from their dependent modal verbs. Sometimes longer-distance dependencies can be vetted using sentence-internal punctuation marks.

It happens that the presence of punctuation between such conjunction (`IN`) and verb (`MD`) types serves as a clue that they are not connected (see Table 7*a*); by contrast, a simpler cue — whether these words are adjacent — is, in this case, hardly of any use (see Table 7*b*). Conditioning on crossing punctuation could be of help then, playing a role similar to that of comma-counting (Collins, 1997, §2.1) — and "verb intervening" (Bikel, 2004, §5.1) — in early head-outward models for supervised parsing.

a) $r_\phi \approx -0.40$	*Attached*	*not Attached*	*Total*
Punctuation	337	7,645	7,982
no Punctuation	2,144	4,040	6,184
Total	2,481	11,685	14,166
non-Adjacent	2,478	11,673	14,151
Adjacent	3	12	15
b) $r_\phi \approx +0.00$	*Attached*	*not Attached*	*Total*

Table 7: Contingency tables for `IN` right-attaching `MD`, among closest ordered pairs of these tokens in WSJ sentences with punctuation, versus: (a) presence of intervening punctuation; and (b) presence of intermediate words.

6.2 Experimental Results Postponed

As we mentioned earlier (see §3), there is little point in testing DBM-3 with shorter sentences, since most sentence-internal punctuation occurs in longer inputs. Instead, we will test this model in a final step of a staged training strategy, with more data (see §7.3).

7 A Curriculum Strategy for DBMs

We propose to train up to DBM-3 iteratively — by beginning with DBM-1 and gradually increasing model complexity through DBM-2, drawing on the intuitions of IBM translation models 1–4 (Brown et al., 1993). Instead of using sentences of up to 15 tokens, as in all previous experiments (§4–5), we will now make use of nearly all available training data: up to length 45 (out of concern for efficiency), during later stages. In the first stage, however, we will use only a subset of the data with DBM-1, in a process sometimes called *curriculum learning* (Bengio et al., 2009; Krueger and Dayan, 2009, *inter alia*). Our grammar inducers will thus be "starting small" in both senses suggested by Elman (1993): simultaneously scaffolding on model- *and* data-complexity.

7.1 Scaffolding Stage #1: DBM-1

We begin by training DBM-1 on sentences without sentence-internal punctuation but with at least one trailing punctuation mark. Our goal is to avoid, when possible, overly specific arbitrary parameters like the "15 tokens or less" threshold used to select training sentences. Unlike DBM-2 and 3, DBM-1 does not model punctuation or sentence fragments, so we instead explicitly restrict its attention to this cleaner subset of the training data, which takes advantage of the fact that punctuation may generally correlate with sentence complexity (Frank, 2000).[9]

Aside from input sentence selection, our experimental set-up here remained identical to previous training of DBMs (§4–5). Using this new input data, DBM-1 averaged 40.7% accuracy (see Table 8). This is slightly higher than the 39.0% when using sentences up to length 15, suggesting that our heuristic for clean, simple sentences may be a useful one.

[9] More incremental training strategies are the subject of an upcoming (companion) manuscript (Spitkovsky et al., 2012a).

CoNLL Year & Language		*Directed Dependency Accuracies for:* this Work DMV	DBM-1	DBM-2	DBM-3	+*inference* (@10)	*Best of State-of-the-Art Systems* Monolingual; POS- (i) Agnostic	(ii) Identified	Cross-Lingual (iii) Transfer
Arabic	2006	12.9	10.6	11.0	11.1	10.9 *(34.5)*	**33.4** $SCAJ_6$	—	**50.2** S_{bg}
	’7	36.6	43.9	44.0	44.4	44.9 *(48.8)*	**55.6** RF	*54.6* RF_{H1}	—
Basque	’7	32.7	34.1	33.0	32.7	33.3 *(36.5)*	**43.6** $SCAJ_5$	*34.7* MZ_{NR}	—
Bulgarian	’7	24.7	59.4	63.6	64.6	**65.2** *(70.4)*	44.3 $SCAJ_5$	*53.9* $RF_{H1\&2}$	**70.3** S_{pt}
Catalan	’7	41.1	61.3	61.1	61.1	62.1 *(78.1)*	**63.8** $SCAJ_5$	*56.3* MZ_{NR}	—
Chinese	’6	50.4	63.1	63.0	63.2	63.2 *(65.7)*	**63.6** $SCAJ_6$	—	—
	’7	55.3	56.8	57.0	57.1	57.0 *(59.8)*	**58.5** $SCAJ_6$	*34.6* MZ_{NR}	—
Czech	’6	31.5	51.3	52.8	53.0	**55.1** *(61.8)*	50.5 $SCAJ_5$	—	—
	’7	34.5	50.5	51.2	53.3	**54.2** *(67.3)*	49.8 $SCAJ_5$	*42.4* $RF_{H1\&2}$	—
Danish	’6	22.4	21.3	19.9	21.8	22.2 *(27.4)*	**46.0** RF	**53.1** $RF_{H1\&2}$	**56.5** S_{ar}
Dutch	’6	44.9	45.9	46.5	46.0	**46.6** *(48.6)*	32.5 $SCAJ_5$	**48.8** $RF_{H1\&2}$	**65.7** $MPH_{m:p}$
English	’7	32.3	29.2	28.6	29.0	29.6 *(51.4)*	**50.3** SAJ	*23.8* MZ_{NR}	*45.7* MPH_{el}
German	’6	27.7	36.3	37.9	38.4	**39.1** *(52.1)*	33.5 $SCAJ_5$	*21.8* MZ_{NR}	**56.7** $MPH_{m:d}$
Greek	’6	36.3	28.1	26.1	26.1	26.9 *(36.8)*	**39.0** MZ	*33.4* MZ_{NR}	**65.1** $MPH_{m:p}$
Hungarian	’7	23.6	43.2	52.1	57.4	**58.2** *(68.4)*	48.0 MZ	*48.1* MZ_{NR}	—
Italian	’7	25.5	41.7	39.8	39.9	40.7 *(41.8)*	**57.5** MZ	**60.6** MZ_{NR}	**69.1** MPH_{pt}
Japanese	’6	42.2	22.8	22.7	22.7	22.7 *(32.5)*	**56.6** $SCAJ_5$	*53.5* MZ_{NR}	—
Portuguese	’6	37.1	68.9	72.3	71.1	**72.4** *(80.6)*	43.2 MZ	*55.8* $RF_{H1\&2}$	**76.9** S_{bg}
Slovenian	’6	33.4	30.4	33.0	34.1	**35.2** *(36.8)*	33.6 $SCAJ_5$	*34.6* MZ_{NR}	—
Spanish	’6	22.0	25.0	26.7	27.1	28.2 *(51.8)*	**53.0** MZ	**54.6** MZ_{NR}	**68.4** MPH_{it}
Swedish	’6	30.7	48.6	50.3	50.0	**50.7** *(63.2)*	50.0 $SCAJ_6$	*34.3* $RF_{H1\&2}$	**68.0** $MPH_{m:p}$
Turkish	’6	**43.4**	32.9	33.7	33.4	34.4 *(38.1)*	40.9 SAJ	**61.3** RF_{H1}	—
	’7	**58.5**	44.6	44.2	43.7	44.8 *(44.4)*	48.8 $SCAJ_6$	—	—
Average:		33.6	40.7	41.7	42.2	**42.9** *(51.9)*	38.2 $SCAJ_6$	(*best average*, **not** an average of bests)	

Table 8: Average accuracies over CoNLL evaluation sets (all sentences), for the DMV baseline and DBM1–3 trained with a curriculum strategy, and state-of-the-art results for systems that: (i) are also POS-agnostic and monolingual, including SCAJ (Spitkovsky et al., 2011a, Tables 5–6) and SAJ (Spitkovsky et al., 2011b); (ii) rely on gold POS-tag identities to discourage noun roots (Mareček and Zabokrtský, 2011, MZ) or to encourage verbs (Rasooli and Faili, 2012, RF); and (iii) transfer delexicalized parsers (Søgaard, 2011a, S) from resource-rich languages with translations (McDonald et al., 2011, MPH). DMV and DBM-1 trained on simple sentences, from uniform; DBM-2 and 3 trained on most sentences, from DBM-1 and 2, respectively; *+inference* is DBM-3 with punctuation constraints.

7.2 Scaffolding Stage #2: DBM-2 ← DBM-1

Next, we trained on all sentences up to length 45. Since these inputs are punctuation-rich, in both remaining stages we used the constrained Viterbi EM set-up suggested by Spitkovsky et al. (2011b) instead of plain soft EM; we employ an early termination strategy, quitting hard EM as soon as soft EM’s objective suffers (Spitkovsky et al., 2011a). Punctuation was converted into Viterbi-decoding constraints during training using the so-called *loose* method, which stipulates that all words in an inter-punctuation fragment must be dominated by a single (head) word, also from that fragment — with only these head words allowed to attach the head words of other fragments, across punctuation boundaries.

To adapt to full data, we initialized DBM-2 using Viterbi parses from the previous stage (§7.1), plus uniformly-at-random chosen dependency trees for the new complex and incomplete sentences, subject to punctuation-induced constraints. This approach improved parsing accuracies to 41.7% (see Table 8).

7.3 Scaffolding Stage #3: DBM-3 ← DBM-2

Next, we repeated the training process of the previous stage (§7.2) using DBM-3. To initialize this model, we combined the final instance of DBM-2 with uniform multinomials for punctuation-crossing attachment probabilities (see §2.3). As a result, average performance improved to 42.2% (see Table 8).

Lastly, we applied punctuation constraints also in inference. Here we used the *sprawl* method — a more relaxed approach than in training, allowing arbitrary words to attach inter-punctuation fragments (provided that each entire fragment still be derived

by one of its words) — as suggested by Spitkovsky et al. (2011b). This technique increased DBM-3's average accuracy to 42.9% (see Table 8). Our final result substantially improves over the baseline's 33.6% and compares favorably to previous work.[10]

8 Discussion and the State-of-the-Art

DBMs come from a long line of head-outward models for dependency grammar induction yet their generative processes feature important novelties. One is conditioning on more observable state — specifically, the left and right end words of a phrase being constructed — than in previous work. Another is allowing multiple grammars — e.g., of complete and incomplete sentences — to coexist in a single model. These improvements could make DBMs quick-and-easy to bootstrap directly from any available partial bracketings (Pereira and Schabes, 1992), for example capitalized phrases (Spitkovsky et al., 2012b).

The second part of our work — the use of a curriculum strategy to train DBM-1 through 3 — eliminates having to know tuned cut-offs, such as sentences with up to a predetermined number of tokens. Although this approach adds some complexity, we chose conservatively, to avoid overfitting settings of sentence length, convergence criteria, etc.: stage one's data is dictated by DBM-1 (which ignores punctuation); subsequent stages initialize additional pieces uniformly: uniform-at-random parses for new data and uniform multinomials for new parameters.

Even without curriculum learning — trained with vanilla EM — DBM-2 and 1 are already strong. Further boosts to accuracy could come from employing more sophisticated optimization algorithms, e.g., better EM (Samdani et al., 2012), constrained Gibbs sampling (Mareček and Zabokrtský, 2011) or locally-normalized features (Berg-Kirkpatrick et al., 2010). Other orthogonal dependency grammar induction techniques — including ones based on universal rules (Naseem et al., 2010) — may also benefit in combination with DBMs. Direct comparisons to previous work require some care, however, as there are several classes of systems that make different assumptions about training data (see Table 8).

[10]Note that DBM-1's 39% average accuracy with standard training (see Table 2) was already nearly a full point higher than that of any single previous best system ($SCAJ_6$ — see Table 8).

8.1 Monolingual POS-Agnostic Inducers

The first type of grammar inducers, including our own approach, uses standard training and test data sets for each language, with gold part-of-speech tags as anonymized word classes. For the purposes of this discussion, we also include in this group transductive learners that may train on data from the test sets. Our DBM-3 (decoded with punctuation constraints) does well among such systems — for which accuracies on *all* sentence lengths of the evaluation sets are reported — attaining highest scores for 8 of 19 languages; the DMV baseline is still state-of-the-art for one language; and the remaining 10 bests are split among five other recent systems (see Table 8).[11] Half of the five came from various lateen EM strategies (Spitkovsky et al., 2011a) for escaping and/or avoiding local optima. These heuristics are compatible with how we trained our DBMs and could potentially provide further improvement to accuracies.

Overall, the final scores of DBM-3 were better, on average, than those of any other single system: 42.9 versus 38.2% (Spitkovsky et al., 2011a, Table 6). The progression of scores for DBM-1 through 3 without using punctuation constraints in inference — 40.7, 41.7 and 42.2% — fell entirely above this previous state-of-the-art result as well; the DMV baseline — also trained on sentences without internal but with final punctuation — averaged 33.6%.

8.2 Monolingual POS-Identified Inducers

The second class of techniques assumes knowledge about identities of part-of-speech tags (Naseem et al., 2010), i.e., which word tokens are verbs, which ones are nouns, etc. Such grammar inducers generally do better than the first kind — e.g., by encouraging verbocentricity (Gimpel and Smith, 2011) — though even here our results appear to be competitive. In fact, to our surprise, only in 5 of 19 languages a "POS-identified" system performed better than all of the "POS-agnostic" ones (see Table 8).

8.3 Multi-Lingual Semi-Supervised Parsers

The final broad class of related algorithms we considered extends beyond monolingual data and uses

[11]For Turkish '06, the "right-attach" baseline outperforms even the DMV, at 65.4% (Rasooli and Faili, 2012, Table 1); an important difference between 2006 and 2007 CoNLL data sets has to do with segmentation of morphologically-rich languages.

both identities of POS-tags and/or parallel bitexts to transfer (supervised) delexicalized parsers across languages. Parser projection is by far the most successful approach to date and we hope that it too may stand to gain from our modeling improvements. Of the 10 languages for which we found results in the literature, transferred parsers underperformed the grammar inducers in only one case: on English (see Table 8). The unsupervised system that performed better used a special "weighted" initializer (Spitkovsky et al., 2011b, §3.1) that worked well for English (but less so for many other languages).

DBMs may be able to improve initialization. For example, modeling of incomplete sentences could help in incremental initialization strategies like *baby steps* (Spitkovsky et al., 2009), which are likely sensitive to the proverbial "bum steer" from unrepresentative short fragments, *pace* Tu and Honavar (2011).

8.4 Miscellaneous Systems on Short Sentences

Several recent systems (Cohen et al., 2011; Søgaard, 2011b; Naseem et al., 2010; Gillenwater et al., 2010; Berg-Kirkpatrick and Klein, 2010, *inter alia*) are absent from Table 8 because they do not report performance for all sentence lengths. To facilitate comparison with this body of important previous work, we also tabulated final accuracies for the "up-to-ten words" task under heading *@10*: 51.9%, on average.

9 Conclusion

Although a dependency parse for a sentence can be mapped to a constituency parse (Xia and Palmer, 2001), the probabilistic models generating them use different conditioning: dependency grammars focus on the relationship between arguments and heads, constituency grammars on the coherence of chunks covered by non-terminals. Since redundant views of data can make learning easier (Blum and Mitchell, 1998), integrating aspects of both constituency and dependency ought to be able to help grammar induction. We have shown that this insight is correct: dependency grammar inducers can gain from modeling boundary information that is fundamental to constituency (i.e., phrase-structure) formalisms.

DBMs are a step in the direction towards modeling constituent boundaries jointly with head dependencies. Further steps must involve more tightly coupling the two frameworks, as well as showing ways to incorporate both kinds of information in other state-of-the art grammar induction paradigms.

Acknowledgments

We thank Roi Reichart and Marta Recasens, for many helpful comments on draft versions of this paper, and Marie-Catherine de Marneffe, Roy Schwartz, Mengqiu Wang and the anonymous reviewers, for their apt recommendations. Funded, in part, by Defense Advanced Research Projects Agency (DARPA) Machine Reading Program, under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the US government. First author is grateful to Cindy Chan for her friendship and support over many long months leading up to this publication.

References

H. Alshawi, S. Bangalore, and S. Douglas. 2000. Learning dependency translation models as collections of finite-state head transducers. *Computational Linguistics*, 26.

H. Alshawi. 1996a. Head automata for speech translation. In *ICSLP*.

H. Alshawi. 1996b. Method and apparatus for an improved language recognition system. US Patent 1999/5870706.

J. K. Baker. 1979. Trainable grammars for speech recognition. In *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*.

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. 2009. Curriculum learning. In *ICML*.

J. Berant, Y. Gross, M. Mussel, B. Sandbank, E. Ruppin, and S. Edelman. 2006. Boosting unsupervised grammar induction by splitting complex sentences on function words. In *BUCLD*.

T. Berg-Kirkpatrick and D. Klein. 2010. Phylogenetic grammar induction. In *ACL*.

T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero, and D. Klein. 2010. Painless unsupervised learning with features. In *NAACL-HLT*.

A. Bhattacharyya. 1943. On a measure of divergence between two statistical populations defined by their probability distributions. *BCMS*, 35.

D. M. Bikel. 2004. Intricacies of Collins' parsing model. *Computational Linguistics*, 30.

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT*.

M. R. Brent and J. M. Siskind. 2001. The role of exposure to isolated words in early vocabulary development. *Cognition*, 81.

P. F. Brown, V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19.

S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *CoNLL*.

G. Carroll and E. Charniak. 1992. Two experiments on learning probabilistic dependency grammars from corpora. Technical report, Brown University.

S. B. Cohen and N. A. Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *NAACL-HLT*.

S. B. Cohen and N. A. Smith. 2010. Viterbi training for PCFGs: Hardness results and competitiveness of uniform initialization. In *ACL*.

S. B. Cohen, D. Das, and N. A. Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *EMNLP*.

M. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *ACL*.

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

M. Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29.

J. Eisner and G. Satta. 1999. Efficient parsing for bilexical context-free grammars and head-automaton grammars. In *ACL*.

J. M. Eisner. 1996. An empirical comparison of probability models for dependency grammar. Technical report, IRCS.

J. Eisner. 2000. Bilexical grammars and their cubic-time parsing algorithms. In H. C. Bunt and A. Nijholt, editors, *Advances in Probabilistic and Other Parsing Technologies*. Kluwer Academic Publishers.

J. L. Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48.

R. Frank. 2000. From regular to context-free to mildly context-sensitive tree rewriting systems: The path of child language acquisition. In A. Abeillé and O. Rambow, editors, *Tree Adjoining Grammars: Formalisms, Linguistic Analysis and Processing*. CSLI Publications.

J. Gillenwater, K. Ganchev, J. Graça, B. Taskar, and F. Pereira. 2009. Sparsity in grammar induction. In *GRLL*.

J. Gillenwater, K. Ganchev, J. Graça, F. Pereira, and B. Taskar. 2010. Posterior sparsity in unsupervised dependency parsing. Technical report, University of Pennsylvania.

K. Gimpel and N. A. Smith. 2011. Concavity and initialization for unsupervised dependency grammar induction. Technical report, CMU.

C. Hänig. 2010. Improvements in unsupervised co-occurrence based parsing. In *CoNLL*.

W. P. Headden, III, M. Johnson, and D. McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *NAACL-HLT*.

D. Klein and C. D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *ACL*.

K. A. Krueger and P. Dayan. 2009. Flexible shaping: How learning in small steps helps. *Cognition*, 110.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19.

D. Mareček and Z. Zabokrtský. 2011. Gibbs sampling with treeness constraint in unsupervised dependency parsing. In *ROBUS*.

D. McClosky. 2008. Modeling valence effects in unsupervised grammar induction. Technical report, Brown University.

R. McDonald, S. Petrov, and K. Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *EMNLP*.

T. Naseem, H. Chen, R. Barzilay, and M. Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *EMNLP*.

M. S. Nikulin. 2002. Hellinger distance. In M. Hazewinkel, editor, *Encyclopaedia of Mathematics*. Kluwer Academic Publishers.

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *EMNLP-CoNLL*.

M. A. Paskin. 2001a. Cubic-time parsing and learning algorithms for grammatical bigram models. Technical report, UCB.

M. A. Paskin. 2001b. Grammatical bigrams. In *NIPS*.

F. Pereira and Y. Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *ACL*.

E. Ponvert, J. Baldridge, and K. Erk. 2010. Simple unsupervised identification of low-level constituents. In *ICSC*.

M. S. Rasooli and H. Faili. 2012. Fast unsupervised dependency parsing with arc-standard transitions. In *ROBUS-UNSUP*.

R. Samdani, M.-W. Chang, and D. Roth. 2012. Unified expectation maximization. In *NAACL-HLT*.

Y. Seginer. 2007. *Learning Syntactic Structure*. Ph.D. thesis, University of Amsterdam.

A. Søgaard. 2011a. Data point selection for cross-language adaptation of dependency parsers. In *ACL-HLT*.

A. Søgaard. 2011b. From ranked words to dependency trees: two-stage unsupervised non-projective dependency parsing. In *TextGraphs*.

V. I. Spitkovsky, H. Alshawi, and D. Jurafsky. 2009. Baby Steps: How "Less is More" in unsupervised dependency parsing. In *GRLL*.

V. I. Spitkovsky, H. Alshawi, and D. Jurafsky. 2011a. Lateen EM: Unsupervised training with multiple objectives, applied to dependency grammar induction. In *EMNLP*.

V. I. Spitkovsky, H. Alshawi, and D. Jurafsky. 2011b. Punctuation: Making a point in unsupervised dependency parsing. In *CoNLL*.

V. I. Spitkovsky, H. Alshawi, and D. Jurafsky. 2012a. Bootstrapping dependency grammar inducers from incomplete sentence fragments via austere models. In *ICGI*.

V. I. Spitkovsky, H. Alshawi, and D. Jurafsky. 2012b. Capitalization cues improve dependency grammar induction. In *WILS*.

K. Tu and V. Honavar. 2011. On the utility of curricula in unsupervised learning of probabilistic grammars. In *IJCAI*.

F. Xia and M. Palmer. 2001. Converting dependency structures to phrase structures. In *HLT*.