From Baby Steps to Leapfrog: How "Less is More" in Unsupervised Dependency Parsing*

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 three approaches for unsupervised grammar induction that are sensitive to data complexity and apply them to Klein and Manning's Dependency Model with Valence. The first, *Baby Steps*, bootstraps itself via iterated learning of increasingly longer sentences and requires no initialization. This method substantially exceeds Klein and Manning's published scores and achieves 39.4% accuracy on Section 23 (all sentences) of the Wall Street Journal corpus. The second, *Less is More*, uses a low-complexity subset of the available data: sentences up to length 15. Focusing on fewer but simpler examples trades off quantity against ambiguity; it attains 44.1% accuracy, using the standard linguistically-informed prior and batch training, beating state-of-the-art. *Leapfrog*, our third heuristic, combines *Less is More* with *Baby Steps* by mixing their models of shorter sentences, then rapidly ramping up exposure to the full training set, driving up accuracy to 45.0%. These trends generalize to the Brown corpus; awareness of data complexity may improve other parsing models and unsupervised algorithms.

1 Introduction

Unsupervised learning of hierarchical syntactic structure from free-form natural language text is a hard problem whose eventual solution promises to benefit applications ranging from question answering to speech recognition and machine translation. A restricted version that targets dependencies and assumes partial annotation, e.g., sentence boundaries, tokenization and typically even part-of-speech (POS) tagging, has received much attention, eliciting a diverse array of techniques (Smith and Eisner, 2005; Seginer, 2007; Cohen et al., 2008). Klein and Manning's (2004) Dependency Model with Valence (DMV) was the first to beat a simple parsing heuristic — the right-branching baseline. Today's state-of-the-art systems (Headden et al., 2009; Cohen and Smith, 2009) are still rooted in the DMV.

Despite recent advances, unsupervised parsers lag far behind their supervised counterparts. Although large amounts of unlabeled data are known to improve semi-supervised parsing (Suzuki et al., 2009), the best unsupervised systems use less data than is available for supervised training, relying on complex models instead: Headden et al.'s (2009) Extended Valence Grammar (EVG) combats data sparsity with smoothing alone, training on the same small subset of the tree-bank as the classic implementation of the DMV; Cohen and Smith (2009) use more complicated algorithms (variational EM and MBR decoding) and stronger linguistic hints (tying related parts of speech and syntactically similar bilingual data).

We explore what can be achieved through judicious use of data and simple, scalable techniques. Our first approach iterates over a series of training sets that gradually increase in size and complexity, forming an initialization-independent scaffolding for learning a grammar. It works with Klein and Manning's simple model (the original DMV) and training algorithm (classic EM) but eliminates their crucial dependence on manually-tuned priors. The second technique is consistent with the intuition that learning is most successful within a band of the size-complexity spectrum. Both could be applied to more

*Partially funded by NSF award IIS-0811974; first author supported by the Fannie & John Hertz Foundation Fellowship.

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 751–759,
Los Angeles, California, June 2010.

intricate models and advanced learning algorithms. We combine them in a third, efficient hybrid method.

2 Intuition

Focusing on simple examples helps guide unsupervised learning,[1] as blindly added confusing data can easily mislead training. We suggest that unless it is increased gradually, unbridled, complexity can overwhelm a system. How to grade an example's difficulty? The cardinality of its solution space presents a natural proxy. In the case of parsing, the number of possible syntactic trees grows exponentially with sentence length. For longer sentences, the unsupervised optimization problem becomes severely under-constrained, whereas for shorter sentences, learning is tightly reined in by data. In the extreme case of a single-word sentence, there is no choice but to parse it correctly. At two words, a raw 50% chance of telling the head from its dependent is still high, but as length increases, the accuracy of even educated guessing rapidly plummets. In model re-estimation, long sentences amplify ambiguity and pollute fractional counts with noise. At times, batch systems are better off using less data.

Baby Steps: Global non-convex optimization is hard. We propose a meta-heuristic that takes the guesswork out of initializing local search. Beginning with an easy (convex) case, it slowly extends it to the fully complex target task by taking tiny steps in the problem space, trying not to stray far from the relevant neighborhoods of the solution space. A series of nested subsets of increasingly longer sentences that culminates in the complete data set offers a natural progression. Its base case — sentences of length one — has a trivial solution that requires neither initialization nor search yet reveals something of sentence heads. The next step — sentences of length one and two — refines initial impressions of heads, introduces dependents, and exposes their identities and relative positions. Although not representative of the full grammar, short sentences capture enough information to paint most of the picture needed by slightly longer sentences. They set up an easier, incremental subsequent learning task. Step $k+1$ augments training input to include lengths $1, 2, \ldots, k, k+1$ of the full data set and executes local search starting from the (smoothed) model estimated by step k. This truly is grammar induction.

Less is More: For standard batch training, just using simple, short sentences is not enough. They are rare and do not reveal the full grammar. We find a "sweet spot" — sentence lengths that are neither too long (excluding the truly daunting examples) nor too few (supplying enough accessible information), using Baby Steps' learning curve as a guide. We train where it flattens out, since remaining sentences contribute little (incremental) educational value.[2]

Leapfrog: As an alternative to discarding data, a better use of resources is to combine the results of batch and iterative training up to the sweet spot data gradation, then iterate with a large step size.

3 Related Work

Two types of scaffolding for guiding language learning debuted in Elman's (1993) experiments with "starting small": data complexity (restricting input) and model complexity (restricting memory). In both cases, gradually increasing complexity allowed artificial neural networks to master a pseudo-natural grammar they otherwise failed to learn. Initially-limited capacity resembled maturational changes in working memory and attention span that occur over time in children (Kail, 1984), in line with the "less is more" proposal (Newport, 1988; 1990). Although Rohde and Plaut (1999) failed to replicate this[3] result with simple recurrent networks, other machine learning techniques reliably benefit from scaffolded model complexity on a variety of language tasks. In word-alignment, Brown et al. (1993) used IBM Models 1-4 as "stepping stones" to training Model 5. Other prominent examples include "coarse-to-fine"

[1] It mirrors the effect that boosting hard examples has for supervised training (Freund and Schapire, 1997).

[2] This is akin to McClosky et al.'s (2006) "Goldilocks effect."

[3] Worse, they found that limiting input *hindered* language acquisition. And making the grammar more English-like (by introducing and strengthening semantic constraints), *increased* the already significant advantage for "starting large!" With iterative training invoking the optimizer multiple times, creating extra opportunities to converge, Rohde and Plaut (1999) suspected that Elman's (1993) simulations simply did not allow networks exposed exclusively to complex inputs sufficient training time. Our extremely generous, low termination threshold for EM (see §5.1) addresses this concern. However, given the DMV's purely syntactic POS tag-based approach (see §5), it would be prudent to re-test Baby Steps with a lexicalized model.

approaches to parsing, translation and speech recognition (Charniak and Johnson, 2005; Charniak et al., 2006; Petrov et al., 2008; Petrov, 2009), and recently unsupervised POS tagging (Ravi and Knight, 2009). Initial models tend to be particularly simple,[4] and each refinement towards a full model introduces only limited complexity, supporting incrementality.

Filtering complex data, the focus of our work, is unconventional in natural language processing. Such scaffolding qualifies as *shaping* — a method of instruction (routinely exploited in animal training) in which the teacher decomposes a complete task into sub-components, providing an easier path to learning. When Skinner (1938) coined the term, he described it as a "method of successive approximations." Ideas that gradually make a task more difficult have been explored in robotics (typically, for navigation), with reinforcement learning (Singh, 1992; Sanger, 1994; Saksida et al., 1997; Dorigo and Colombetti, 1998; Savage, 1998; Savage, 2001). Recently, Krueger and Dayan (2009) showed that shaping speeds up language acquisition and leads to better generalization in abstract neural networks. Bengio et al. (2009) confirmed this for deep deterministic and stochastic networks, using simple multi-stage *curriculum* strategies. They conjectured that a well-chosen sequence of training criteria — different sets of weights on the examples — could act as a continuation method (Allgower and Georg, 1990), helping find better local optima for non-convex objectives. Elman's learners constrained the peaky solution space by focusing on just the right data (simple sentences that introduced basic representational categories) at just the right time (early on, when their plasticity was greatest). Self-shaping, they simplified tasks through deliberate omission (or misunderstanding). Analogously, Baby Steps induces an early structural locality bias (Smith and Eisner, 2006), then relaxes it, as if annealing (Smith and Eisner, 2004). Its curriculum of binary weights initially discards complex examples responsible for "high-frequency noise," with earlier, "smoothed" objectives revealing more of the global picture.

There are important differences between our results and prior work. In contrast to Elman, we use a large data set (WSJ) of real English. Unlike Bengio et al. and Krueger and Dayan, we shape a parser, not a language model. Baby Steps is similar, in spirit, to Smith and Eisner's methods. Deterministic annealing (DA) shares nice properties with Baby Steps, but performs worse than EM for (constituent) parsing; Baby Steps handedly defeats standard training. Structural annealing works well, but requires a hand-tuned annealing schedule and direct manipulation of the objective function; Baby Steps works "out of the box," its locality biases a natural consequence of a complexity/data-guided tour of optimization problems. Skewed DA incorporates a good initializer by interpolating between two probability distributions, whereas our hybrid, Leapfrog, admits multiple initializers by mixing structures instead. "Less is More" is novel and confirms the tacit consensus implicit in training on small data sets (e.g., WSJ10).

4 Data Sets and Metrics

Klein and Manning (2004) both trained and tested the DMV on the same customized subset (WSJ10) of Penn English Treebank's Wall Street Journal portion (Marcus et al., 1993). Its 49,208 annotated parse trees were pruned[5] down to 7,422 sentences of at most 10 terminals, spanning 35 unique POS tags. Following standard practice, automatic "head-percolation" rules (Collins, 1999) were used to convert the remaining trees into dependencies. Forced to produce a single "best" parse, their algorithm was judged on accuracy: its *directed score* was the fraction of correct dependencies; a more flattering[6] *undirected score* was also used. We employ the same metrics, emphasizing directed scores, and generalize WSJk to be the subset of pre-processed sentences with at most k terminals. Our experiments focus on $k \in \{1, \ldots, 45\}$, but we also test on WSJ100 and Section 23 of WSJ$^\infty$ (the entire WSJ), as well as the held-out Brown100 (similarly derived from the Brown corpus (Francis and Kucera, 1979)). See Figure 1 for these corpora's sentence and token counts.

[4] Brown et al.'s (1993) Model 1 (and, similarly, the first baby step) has a global optimum that can be computed exactly, so that no initial or subsequent parameters depend on initialization.

[5] Stripped of all empty sub-trees, punctuation, and terminals (tagged # and $) not pronounced where they appear, those sentences still containing more than ten tokens were thrown out.

[6] Ignoring polarity of parent-child relations partially obscured effects of alternate analyses (systematic choices between modals and main verbs for heads of sentences, determiners for noun phrases, etc.) and facilitated comparison with prior work.

Corpus	*Sentences*	*POS Tokens*	*Corpus*	*Sentences*	*POS Tokens*
WSJ1	159	159	WSJ13	12,270	110,760
WSJ2	499	839	WSJ14	14,095	136,310
WSJ3	876	1,970	WSJ15	15,922	163,715
WSJ4	1,394	4,042	WSJ20	25,523	336,555
WSJ5	2,008	7,112	WSJ25	34,431	540,895
WSJ6	2,745	11,534	WSJ30	41,227	730,099
WSJ7	3,623	17,680	WSJ35	45,191	860,053
WSJ8	4,730	26,536	WSJ40	47,385	942,801
WSJ9	5,938	37,408	WSJ45	48,418	986,830
WSJ10	7,422	52,248	WSJ100	49,206	1,028,054
WSJ11	8,856	68,022	Section 23	2,353	48,201
WSJ12	10,500	87,750	Brown100	24,208	391,796

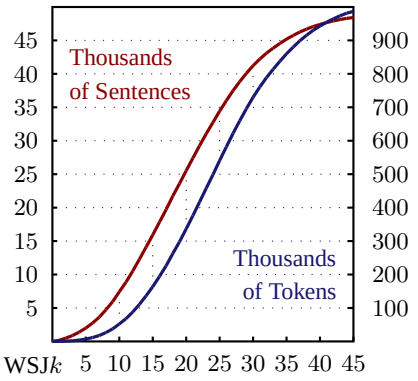
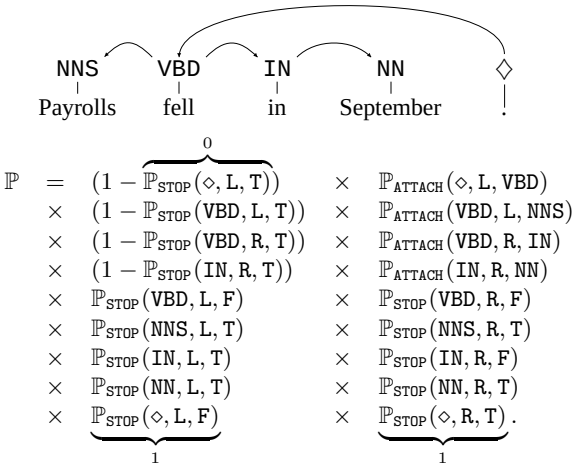


Figure 1: Sizes of WSJ$\{1, \ldots, 45, 100\}$, Section 23 of WSJ$^{\infty}$ and Brown100.



$$
\begin{aligned}
\mathbb{P} \;=\; & \overbrace{(1 - \mathbb{P}_{\text{STOP}}(\diamond, \text{L}, \text{T}))}^{0} && \times\; \mathbb{P}_{\text{ATTACH}}(\diamond, \text{L}, \text{VBD}) \\
\times\; & (1 - \mathbb{P}_{\text{STOP}}(\text{VBD}, \text{L}, \text{T})) && \times\; \mathbb{P}_{\text{ATTACH}}(\text{VBD}, \text{L}, \text{NNS}) \\
\times\; & (1 - \mathbb{P}_{\text{STOP}}(\text{VBD}, \text{R}, \text{T})) && \times\; \mathbb{P}_{\text{ATTACH}}(\text{VBD}, \text{R}, \text{IN}) \\
\times\; & (1 - \mathbb{P}_{\text{STOP}}(\text{IN}, \text{R}, \text{T})) && \times\; \mathbb{P}_{\text{ATTACH}}(\text{IN}, \text{R}, \text{NN}) \\
\times\; & \mathbb{P}_{\text{STOP}}(\text{VBD}, \text{L}, \text{F}) && \times\; \mathbb{P}_{\text{STOP}}(\text{VBD}, \text{R}, \text{F}) \\
\times\; & \mathbb{P}_{\text{STOP}}(\text{NNS}, \text{L}, \text{T}) && \times\; \mathbb{P}_{\text{STOP}}(\text{NNS}, \text{R}, \text{T}) \\
\times\; & \mathbb{P}_{\text{STOP}}(\text{IN}, \text{L}, \text{T}) && \times\; \mathbb{P}_{\text{STOP}}(\text{IN}, \text{R}, \text{F}) \\
\times\; & \mathbb{P}_{\text{STOP}}(\text{NN}, \text{L}, \text{T}) && \times\; \mathbb{P}_{\text{STOP}}(\text{NN}, \text{R}, \text{T}) \\
\times\; & \underbrace{\mathbb{P}_{\text{STOP}}(\diamond, \text{L}, \text{F})}_{1} && \times\; \underbrace{\mathbb{P}_{\text{STOP}}(\diamond, \text{R}, \text{T})}_{1} .
\end{aligned}
$$

Figure 2: A simple dependency structure for a short sentence and its probability, as factored by the DMV.

5 New Algorithms for the Classic Model

The DMV (Klein and Manning, 2004) is a single-state head automata model (Alshawi, 1996) over lexical word classes $\{c_w\}$ — POS tags. Its generative story for a sub-tree rooted at a head (of class c_h) rests on three types of independent decisions: (i) initial direction $dir \in \{\text{L}, \text{R}\}$ in which to attach children, via probability $\mathbb{P}_{\text{ORDER}}(c_h)$; (ii) whether to seal dir, stopping with probability $\mathbb{P}_{\text{STOP}}(c_h, dir, adj)$, conditioned on $adj \in \{\text{T}, \text{F}\}$ (true iff considering dir's first, i.e., *adjacent*, child); and (iii) attachments (of class c_a), according to $\mathbb{P}_{\text{ATTACH}}(c_h, dir, c_a)$. This produces only projective trees.[7] A root token $\diamond$ generates the head of a sentence as its left (and only) child. Figure 2 displays an example that ignores (sums out) $\mathbb{P}_{\text{ORDER}}$.

The DMV lends itself to unsupervised learning via inside-outside re-estimation (Baker, 1979). Klein and Manning did not use smoothing and started with an "ad-hoc harmonic" completion: aiming for balanced trees, non-root heads attached dependents in inverse proportion to (a constant plus) their distance; $\diamond$ generated heads uniformly at random. This non-distributional heuristic created favorable initial conditions that nudged EM towards typical linguistic dependency structures.

5.1 Algorithm #0: Ad-Hoc* — A Variation on Original Ad-Hoc Initialization

Since some of the important implementation details are not available in the literature (Klein and Manning, 2004; Klein, 2005), we had to improvise initialization and terminating conditions. We suspect that our choices throughout this section do not match Klein and Manning's actual training of the DMV.

We use the following ad-hoc harmonic scores (for all tokens other than $\diamond$): $\tilde{\mathbb{P}}_{\text{ORDER}} \equiv 1/2$;

$$\tilde{\mathbb{P}}_{\text{STOP}} \equiv (d_s + \delta_s)^{-1} = (d_s + 3)^{-1},\ d_s \geq 0;$$
$$\tilde{\mathbb{P}}_{\text{ATTACH}} \equiv (d_a + \delta_a)^{-1} = (d_a + 2)^{-1},\ d_a \geq 1.$$

Integers $d_{\{s,a\}}$ are distances from heads to stopping boundaries and dependents.[8] We initialize training by producing best-scoring parses of all input sentences and converting them into proper probability distributions $\mathbb{P}_{\text{STOP}}$ and $\mathbb{P}_{\text{ATTACH}}$ via maximum-likelihood estimation (a single step of Viterbi training (Brown et al., 1993)). Since left and right children are independent, we drop $\mathbb{P}_{\text{ORDER}}$ altogether, mak-

[7] Unlike spanning tree algorithms (McDonald et al., 2005), DMV's chart-based method disallows crossing dependencies.

[8] Constants $\delta_{\{s,a\}}$ come from personal communication. Note that δ_s is one higher than is strictly necessary to avoid both division by zero and determinism; δ_a could have been safely zeroed out, since we never compute $1 - \mathbb{P}_{\text{ATTACH}}$ (see Figure 2).

ing "headedness" deterministic. Our parser carefully randomizes tie-breaking, so that all parse trees having the same score get an equal shot at being selected (both during initialization and evaluation). We terminate EM when a successive change in overall per-token cross-entropy drops below 2^{-20} bits.

5.2 Algorithm #1: Baby Steps — An Initialization-Independent Scaffolding

We eliminate the need for initialization by first training on a trivial subset of the data — WSJ1; this works, since there is only one (the correct) way to parse a single-token sentence. We plug the resulting model into training on WSJ2 (sentences of up to two tokens), and so forth, building up to WSJ45.[9] This algorithm is otherwise identical to Ad-Hoc*, with the exception that it re-estimates each model using Laplace smoothing, so that earlier solutions could be passed to next levels, which sometimes contain previously unseen dependent and head POS tags.

5.3 Algorithm #2: Less is More — Ad-Hoc* where Baby Steps Flatlines

We jettison long, complex sentences and deploy Ad-Hoc*'s initializer and batch training at WSJ$\hat{k}^*$ — an estimate of the sweet spot data gradation. To find it, we track Baby Steps' successive models' cross-entropies on the complete data set, WSJ45. An initial segment of rapid improvement is separated from the final region of convergence by a *knee* — points of maximum curvature (see Figure 3). We use an improved[10] L method (Salvador and Chan, 2004) to automatically locate this area of diminishing returns. Specifically, we determine its end-points $[k_0, k^*]$ by minimizing squared error, estimating $\hat{k}_0 = 7$ and $\hat{k}^* = 15$. Training at WSJ15 just misses the plateau.

5.4 Algorithm #3: Leapfrog — A Practical and Efficient Hybrid Mixture

Cherry-picking the best features of "Less is More" and Baby Steps, we begin by combining their models at WSJ$\hat{k}^*$. Using one best parse from each, for every sentence in WSJ$\hat{k}^*$, the base case re-estimates a new model from a *mixture* of twice the normal number of trees; inductive steps leap over $\hat{k}^*$ lengths, conveniently ending at WSJ45, and estimate their initial models by applying a previous solution to a new input set. Both follow up the single step of Viterbi training with at most five iterations of EM.

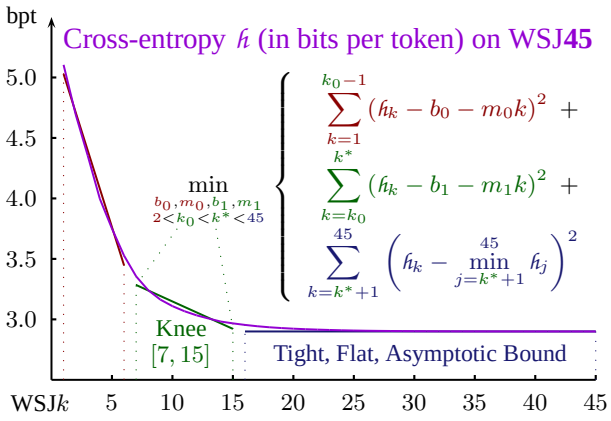


Figure 3: Cross-entropy on WSJ45 after each baby step, a piece-wise linear fit, and an estimated region for the knee.

Our hybrid makes use of two good (conditionally) independent initialization strategies and executes many iterations of EM where that is cheap — at shorter sentences (WSJ15 and below). It then increases the step size, training just three more times (at WSJ$\{15, 30, 45\}$) and allowing only a few (more expensive) iterations of EM. Early termination improves efficiency and regularizes these final models.

5.5 Reference Algorithms — Baselines, a Skyline and Published Art

We carve out the problem space using two extreme initialization strategies: (i) the uninformed uniform prior, which serves as a fair "zero-knowledge" baseline for comparing uninitialized models; and (ii) the maximum-likelihood "oracle" prior, computed from reference parses, which yields a *skyline* (a reverse baseline) — how well any algorithm that stumbled on the true solution would fare at EM's convergence.

In addition to citing Klein and Manning's (2004) results, we compare our accuracies on Section 23 of WSJ$^\infty$ to two state-of-the-art systems and past baselines (see Table 2). Headden et al.'s (2009) lexicalized EVG is the best on short sentences, but

[9] Its 48,418 sentences (see Figure 1) cover 94.4% of all sentences in WSJ; the longest of the missing 790 has length 171.

[10] Instead of iteratively fitting a two-segment form and adaptively discarding its tail, we use *three* line segments, applying ordinary least squares to the first two, but requiring the third to be horizontal and tangent to a minimum. The result is a *batch* optimization routine that returns an *interval* for the knee, rather than a point estimate (see Figure 3 for details).

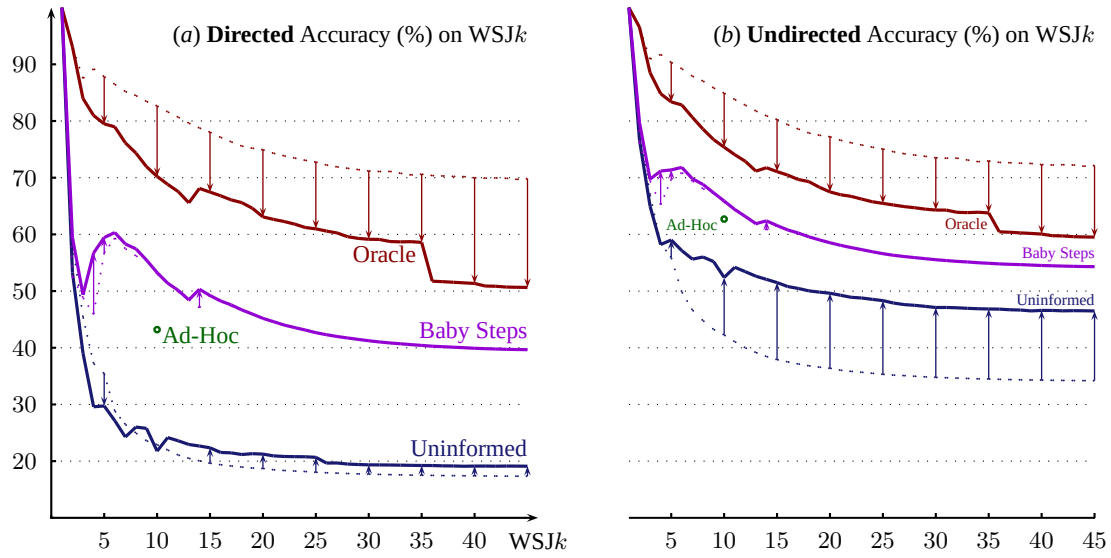


Figure 4: Directed and undirected accuracy scores attained by the DMV, when trained and tested on the same gradation of WSJ, for several different initialization strategies. Green circles mark Klein and Manning's (2004) published scores; red, violet and blue curves represent the supervised (maximum-likelihood oracle) initialization, Baby Steps, and the uninformed uniform prior. Dotted curves reflect starting performance, solid curves register performance at EM's convergence, and the arrows connecting them emphasize the impact of learning.

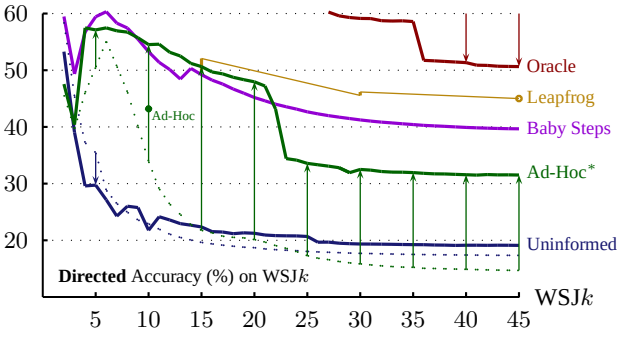


Figure 5: Directed accuracies for Ad-Hoc* (shown in green) and Leapfrog (in gold); all else as in Figure 4(*a*).

its performance is unreported for longer sentences, for which Cohen and Smith's (2009) seem to be the highest published scores; we include their intermediate results that preceded parameter-tying — Bayesian models with Dirichlet and log-normal priors, coupled with both Viterbi and minimum Bayes-risk (MBR) decoding (Cohen et al., 2008).

6 Experimental Results

We packed thousands of empirical outcomes into the space of several graphs (Figures 4, 5 and 6). The colors (also in Tables 1 and 2) correspond to different initialization strategies — to a first approximation, the learning algorithm was held constant (see §5).

Figures 4 and 5 tell one part of our story. As data sets increase in size, training algorithms gain access to more information; however, since in this unsupervised setting training and test sets are the same, additional longer sentences make for substantially more challenging evaluation. To control for these dynamics, we applied Laplace smoothing to all (otherwise unsmoothed) models and re-plotted their performance, holding several test sets fixed, in Figure 6.

We report undirected accuracies parenthetically.

6.1 Result #1: Baby Steps

Figure 4 traces out performance on the training set. Klein and Manning's (2004) published scores appear as dots (Ad-Hoc) at WSJ10: 43.2% (63.7%). Baby Steps achieves 53.0% (65.7%) by WSJ10; trained and tested on WSJ45, it gets 39.7% (54.3%). Uninformed, classic EM learns little about directed dependencies: it improves only slightly, e.g., from 17.3% (34.2%) to 19.1% (46.5%) on WSJ45 (learning some of the structure, as evidenced by its undirected scores), but degrades with shorter sentences, where its initial guessing rate is high. In the case of oracle training, we expected EM to walk away from supervised solutions (Elworthy, 1994; Meri-

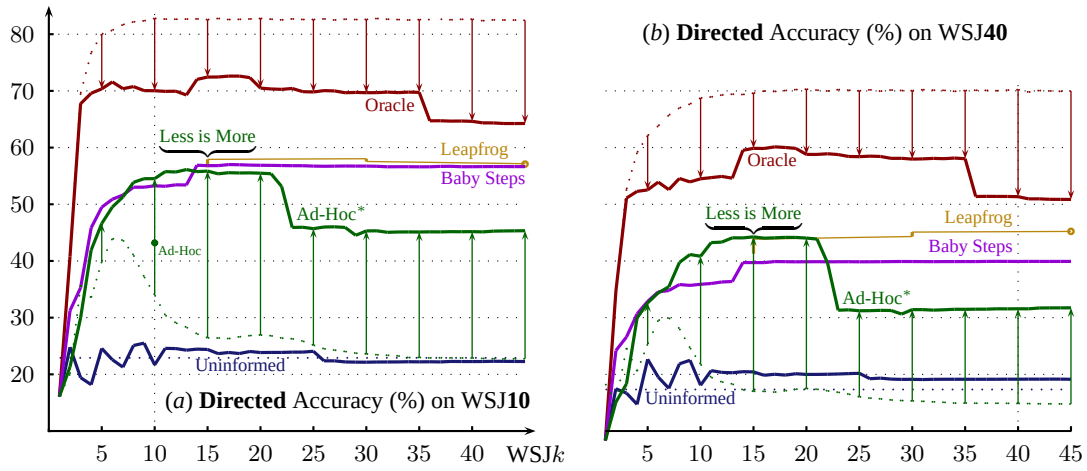


Figure 6: Directed accuracies attained by the DMV, when trained at various gradations of WSJ, smoothed, then tested against fixed evaluation sets — WSJ$\{10, 40\}$; graphs for WSJ$\{20, 30\}$, not shown, are qualitatively similar to WSJ40.

aldo, 1994; Liang and Klein, 2008), but the extent of its drops is alarming, e.g., from the supervised 69.8% (72.2%) to the skyline's 50.6% (59.5%) on WSJ45. In contrast, Baby Steps' scores usually do not change much from one step to the next, and where its impact of learning is big (at WSJ$\{4, 5, 14\}$), it is invariably positive.

6.2 Result #2: Less is More

Ad-Hoc*'s curve (see Figure 5) suggests how Klein and Manning's Ad-Hoc initializer may have scaled with different gradations of WSJ. Strangely, our implementation performs significantly above their reported numbers at WSJ10: 54.5% (68.3%) is even slightly higher than Baby Steps; nevertheless, given enough data (from WSJ22 onwards), Baby Steps overtakes Ad-Hoc*, whose ability to learn takes a serious dive once the inputs become sufficiently complex (at WSJ23), and never recovers. Note that Ad-Hoc*'s biased prior peaks early (at WSJ6), eventually falls below the guessing rate (by WSJ24), yet still remains well-positioned to climb, outperforming uninformed learning.

Figure 6 shows that Baby Steps scales better with more (complex) data — its curves do not trend downwards. However, a good initializer induces a sweet spot at WSJ15, where the DMV is learned best using Ad-Hoc*. This mode *is* "Less is More," scoring 44.1% (58.9%) on WSJ45. Curiously, even oracle training exhibits a bump at WSJ15: once sentences get long enough (at WSJ36), its performance degrades below that of oracle training with virtually no supervision (at the hardly representative WSJ3).

6.3 Result #3: Leapfrog

Mixing Ad-Hoc* with Baby Steps at WSJ15 yields a model whose performance initially falls between its two parents but surpasses both with a little training (see Figure 5). Leaping to WSJ45, via WSJ30, results in our strongest model: its 45.0% (58.4%) accuracy bridges half of the gap between Baby Steps and the skyline, and at a tiny fraction of the cost.

6.4 Result #4: Generalization

Our models carry over to the larger WSJ100, Section 23 of WSJ$^\infty$, and the independent Brown100 (see Table 1). Baby Steps improves out of domain, confirming that shaping generalizes well (Krueger and Dayan, 2009; Bengio et al., 2009). Leapfrog does best across the board but dips on Brown100, despite its safe-guards against over-fitting.

Section 23 (see Table 2) reveals, unexpectedly, that Baby Steps would have been state-of-the-art in 2008, whereas "Less is More" outperforms all prior work on longer sentences. Baby Steps is competitive with log-normal families (Cohen et al., 2008), scoring slightly better on longer sentences against Viterbi decoding, though worse against MBR. "Less is More" beats state-of-the-art on longer sentences by close to 2%; Leapfrog gains another 1%.

	*Ad-Hoc**	*Baby Steps*	*Leapfrog*		*Ad-Hoc**	*Baby Steps*	*Leapfrog*	
Section 23	44.1 (58.8)	39.2 (53.8)	43.3 (55.7)	@15	31.5 (51.6)	39.4 (54.0)	**45.0** (58.4)	@45
WSJ100	43.8 (58.6)	39.2 (53.8)	43.3 (55.6)		31.3 (51.5)	39.4 (54.1)	**44.7** (58.1)	
Brown100	43.3 (59.2)	42.3 (55.1)	42.8 (56.5)		32.0 (52.4)	42.5 (55.5)	**43.6** (59.1)	

Table 1: Directed and undirected accuracies on Section 23 of WSJ$^{\infty}$, WSJ100 and Brown100 for Ad-Hoc*, Baby Steps and Leapfrog, trained at WSJ15 and WSJ45.

			Decoding	WSJ10	WSJ20	WSJ$^{\infty}$
	Attach-Right	(Klein and Manning, 2004)	—	38.4	33.4	31.7
DMV	Ad-Hoc	(Klein and Manning, 2004)	Viterbi	45.8	39.1	34.2
	Dirichlet	(Cohen et al., 2008)	Viterbi	45.9	39.4	34.9
	Ad-Hoc	(Cohen et al., 2008)	MBR	46.1	39.9	35.9
	Dirichlet	(Cohen et al., 2008)	MBR	46.1	40.6	36.9
	Log-Normal Families	(Cohen et al., 2008)	Viterbi	59.3	45.1	39.0
	Baby Steps (@15)		*Viterbi*	*55.5*	*44.3*	*39.2*
	Baby Steps (@45)		*Viterbi*	*55.1*	*44.4*	*39.4*
	Log-Normal Families	(Cohen et al., 2008)	MBR	59.4	45.9	40.5
	Shared Log-Normals (tie-verb-noun)	(Cohen and Smith, 2009)	MBR	61.3	47.4	41.4
	Bilingual Log-Normals (tie-verb-noun)	(Cohen and Smith, 2009)	MBR	62.0	48.0	42.2
	Less is More (Ad-Hoc* @15)		*Viterbi*	*56.2*	*48.2*	*44.1*
	Leapfrog (Hybrid @45)		*Viterbi*	*57.1*	**48.7**	**45.0**
EVG	Smoothed (skip-val)	(Headden et al., 2009)	Viterbi	62.1		
	Smoothed (skip-head)	(Headden et al., 2009)	Viterbi	65.0		
	Smoothed (skip-head), Lexicalized	(Headden et al., 2009)	Viterbi	**68.8**		

Table 2: Directed accuracies on Section 23 of WSJ$\{10, 20, \infty\}$ for several baselines and recent state-of-the-art systems.

7 Conclusion

We explored three simple ideas for unsupervised dependency parsing. Pace Halevy et al. (2009), we find "Less is More" — the paradoxical result that better performance can be attained by training with less data, even when removing samples from the true (test) distribution. Our small tweaks to Klein and Manning's approach of 2004 break through the 2009 state-of-the-art on longer sentences, when trained at WSJ15 (the auto-detected sweet spot gradation).

The second, Baby Steps, is an elegant meta-heuristic for optimizing non-convex training criteria. It eliminates the need for linguistically-biased manually-tuned initializers, particularly if the location of the sweet spot is not known. This technique scales gracefully with more (complex) data and should easily carry over to more powerful parsing models and learning algorithms.

Finally, Leapfrog forgoes the elegance and meticulousness of Baby Steps in favor of pragmatism. Employing both good initialization strategies at its disposal, and spending CPU cycles wisely, it achieves better performance than both "Less is More" and Baby Steps.

Future work could explore unifying these techniques with other state-of-the-art approaches. It may be useful to scaffold on both data and model complexity, e.g., by increasing head automata's number of states (Alshawi and Douglas, 2000). We see many opportunities for improvement, considering the poor performance of oracle training relative to the supervised state-of-the-art, and in turn the poor performance of unsupervised state-of-the-art relative to the oracle models.[11] To this end, it would be instructive to understand both the linguistic and statistical nature of the sweet spot, and to test its universality.

Acknowledgments

We thank Angel X. Chang, Pi-Chuan Chang, David L.W. Hall, Christopher D. Manning, David McClosky, Daniel Ramage and the anonymous reviewers for many helpful comments on draft versions of this paper.

References

E. L. Allgower and K. Georg. 1990. *Numerical Continuation Methods: An Introduction.* Springer-Verlag.

[11] To facilitate future work, all of our models are publicly available at `http://cs.stanford.edu/~valentin/`.

H. Alshawi and S. Douglas. 2000. Learning dependency transduction models from unannotated examples. In *Royal Society of London Philosophical Transactions Series A*, volume 358.

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

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*.

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.

E. Charniak and M. Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proc. of ACL*.

E. Charniak, M. Johnson, M. Elsner, J. Austerweil, D. Ellis, I. Haxton, C. Hill, R. Shrivaths, J. Moore, M. Pozar, and T. Vu. 2006. Multilevel coarse-to-fine PCFG parsing. In *HLT-NAACL*.

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

S. B. Cohen, K. Gimpel, and N. A. Smith. 2008. Logistic normal priors for unsupervised probabilistic grammar induction. In *NIPS*.

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

M. Dorigo and M. Colombetti. 1998. *Robot Shaping: An Experiment in Behavior Engineering*. MIT Press/Bradford Books.

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

D. Elworthy. 1994. Does Baum-Welch re-estimation help taggers? In *Proc. of ANLP*.

W. N. Francis and H. Kucera, 1979. *Manual of Information to Accompany a Standard Corpus of Present-Day Edited American English, for use with Digital Computers*. Department of Linguistic, Brown University.

Y. Freund and R. E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1).

A. Halevy, P. Norvig, and F. Pereira. 2009. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2).

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

R. Kail. 1984. *The development of memory in children*. W. H. Freeman and Company, 2nd edition.

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

D. Klein. 2005. *The Unsupervised Learning of Natural Language Structure*. Ph.D. thesis, Stanford University.

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

P. Liang and D. Klein. 2008. Analyzing the errors of unsupervised learning. In *Proc. of HLT-ACL*.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2).

D. McClosky, E. Charniak, and M. Johnson. 2006. Effective self-training for parsing. In *Proc. of NAACL-HLT*.

R. McDonald, F. Pereira, K. Ribarov, and J. Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proc. of HLT-EMNLP*.

B. Merialdo. 1994. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):155–172.

E. L. Newport. 1988. Constraints on learning and their role in language acquisition: Studies of the acquisition of American Sign Language. *Language Sciences*, 10(1).

E. L. Newport. 1990. Maturational constraints on language learning. *Cognitive Science*, 14(1).

S. Petrov, A. Haghighi, and D. Klein. 2008. Coarse-to-fine syntactic machine translation using language projections. In *Proc. of EMNLP*.

S. O. Petrov. 2009. *Coarse-to-Fine Natural Language Processing*. Ph.D. thesis, University of California, Berkeley.

S. Ravi and K. Knight. 2009. Minimized models for unsupervised part-of-speech tagging. In *Proc. of ACL-IJCNLP*.

D. L. T. Rohde and D. C. Plaut. 1999. Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, 72(1).

L. M. Saksida, S. M. Raymond, and D. S. Touretzky. 1997. Shaping robot behavior using principles from instrumental conditioning. *Robotics and Autonomous Systems*, 22(3).

S. Salvador and P. Chan. 2004. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Proc. of ICTAI*.

T. D. Sanger. 1994. Neural network learning control of robot manipulators using gradually increasing task difficulty. *IEEE Trans. on Robotics and Automation*, 10.

T. Savage. 1998. Shaping: The link between rats and robots. *Connection Science*, 10(3).

T. Savage. 2001. Shaping: A multiple contingencies analysis and its relevance to behaviour-based robotics. *Connection Science*, 13(3).

Y. Seginer. 2007. Fast unsupervised incremental parsing. In *Proc. of ACL*.

S. P. Singh. 1992. Transfer of learning by composing solutions of elemental squential tasks. *Machine Learning*, 8.

B. F. Skinner. 1938. *The behavior of organisms: An experimental analysis*. Appleton-Century-Crofts.

N. A. Smith and J. Eisner. 2004. Annealing techniques for unsupervised statistical language learning. In *Proc. of ACL*.

N. A. Smith and J. Eisner. 2005. Guiding unsupervised grammar induction using contrastive estimation. In *Proc. of the IJCAI Workshop on Grammatical Inference Applications*.

N. A. Smith and J. Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *Proc. of COLING-ACL*.

J. Suzuki, H. Isozaki, X. Carreras, and M. Collins. 2009. An empirical study of semi-supervised structured conditional models for dependency parsing. In *Proc. of EMNLP*.