Viterbi Training Improves Unsupervised Dependency Parsing

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

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

Daniel Jurafsky and **Christopher D. Manning**
Departments of Linguistics and Computer Science
Stanford University, Stanford, CA, 94305, USA
jurafsky@stanford.edu and manning@cs.stanford.edu

Abstract

We show that Viterbi (or "hard") EM is well-suited to unsupervised grammar induction. It is *more* accurate than standard inside-outside re-estimation (classic EM), significantly faster, and simpler. Our experiments with Klein and Manning's Dependency Model with Valence (DMV) attain state-of-the-art performance — 44.8% accuracy on Section 23 (all sentences) of the Wall Street Journal corpus — without clever initialization; with a good initializer, Viterbi training improves to 47.9%. This generalizes to the Brown corpus, our held-out set, where accuracy reaches 50.8% — a 7.5% gain over previous best results. We find that classic EM learns better from short sentences but cannot cope with longer ones, where Viterbi thrives. However, we explain that both algorithms optimize the wrong objectives and prove that there are fundamental disconnects between the likelihoods of sentences, best parses, and true parses, beyond the well-established discrepancies between likelihood, accuracy and extrinsic performance.

1 Introduction

Unsupervised learning is hard, often involving difficult objective functions. A typical approach is to attempt maximizing the likelihood of unlabeled data, in accordance with a probabilistic model. Sadly, such functions are riddled with local optima (Charniak, 1993, Ch. 7, *inter alia*), since their number of peaks grows exponentially with instances of hidden variables. Furthermore, a higher likelihood does not always translate into superior task-specific accuracy (Elworthy, 1994; Merialdo, 1994). Both complications are real, but we will discuss perhaps more significant shortcomings.

We prove that learning can be error-prone even in cases when likelihood *is* an appropriate measure of extrinsic performance *and* where global optimization is feasible. This is because a key challenge in unsupervised learning is that the *desired* likelihood is unknown. Its absence renders tasks like structure discovery inherently underconstrained. Search-based algorithms adopt surrogate metrics, gambling on convergence to the "right" regularities in data. Their wrong objectives create cases in which *both* efficiency *and* performance improve when expensive exact learning techniques are replaced by cheap approximations.

We propose using Viterbi training (Brown et al., 1993), instead of inside-outside re-estimation (Baker, 1979), to induce hierarchical syntactic structure from natural language text. Our experiments with Klein and Manning's (2004) Dependency Model with Valence (DMV), a popular state-of-the-art model (Headden et al., 2009; Cohen and Smith, 2009; Spitkovsky et al., 2009), beat previous benchmark accuracies by 3.8% (on Section 23 of WSJ) and 7.5% (on parsed Brown).

Since objective functions used in unsupervised grammar induction are provably wrong, advantages of exact inference may not apply. It makes sense to try the Viterbi approximation — it is also wrong, only simpler and cheaper than classic EM. As it turns out, Viterbi EM is not only faster but also more accurate, consistent with hypotheses of de Marcken (1995) and Spitkovsky et al. (2009).

We begin by reviewing the model, standard data sets and metrics, and our experimental results. After relating our contributions to prior work, we delve into proofs by construction, using the DMV.

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

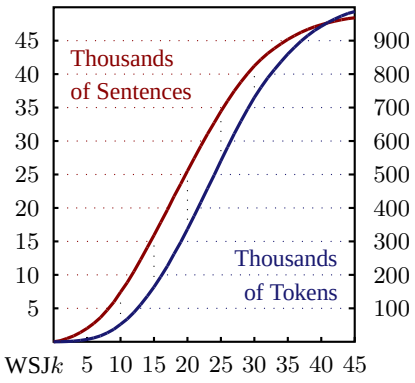


Figure 1: Sizes of WSJ$\{1, \ldots, 45, 100\}$, Section 23 of WSJ$^\infty$ and Brown100 (Spitkovsky et al., 2009).

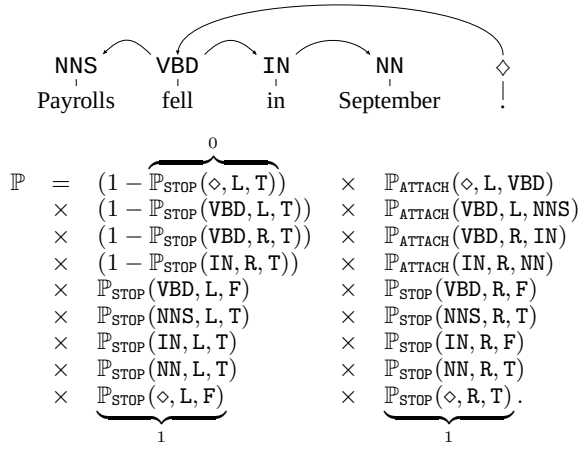


Figure 2: A dependency structure for a short sentence and its probability, as factored by the DMV, after summing out $\mathbb{P}_{\texttt{ORDER}}$ (Spitkovsky et al., 2009).

2 Dependency Model with Valence

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 $dir \in \{\texttt{L}, \texttt{R}\}$ in which to attach children, via probability $\mathbb{P}_{\texttt{ORDER}}(c_h)$; (ii) whether to seal dir, stopping with probability $\mathbb{P}_{\texttt{STOP}}(c_h, dir, adj)$, conditioned on $adj \in \{\texttt{T}, \texttt{F}\}$ (true iff considering dir's first, i.e., *adjacent*, child); and (iii) attachments (of class c_a), according to $\mathbb{P}_{\texttt{ATTACH}}(c_h, dir, c_a)$. This produces only projective trees. A root token $\diamondsuit$ generates the head of a sentence as its left (and only) child. Figure 2 displays a simple example.

The DMV lends itself to unsupervised learning via inside-outside re-estimation (Baker, 1979). Viterbi training (Brown et al., 1993) re-estimates each next model as if supervised by the previous best parse trees. And supervised learning from reference parse trees is straight-forward, since maximum-likelihood estimation reduces to counting: $\hat{\mathbb{P}}_{\texttt{ATTACH}}(c_h, dir, c_a)$ is the fraction of children — those of class c_a — attached on the dir side of a head of class c_h; $\hat{\mathbb{P}}_{\texttt{STOP}}(c_h, dir, adj = \texttt{T})$, the fraction of words of class c_h with no children on the dir side; and $\hat{\mathbb{P}}_{\texttt{STOP}}(c_h, dir, adj = \texttt{F})$, the ratio[1] of the number of words of class c_h having a child on the dir side to their total number of such children.

3 Standard Data Sets and Evaluation

The DMV is traditionally trained and tested on customized subsets of Penn English Treebank's Wall Street Journal portion (Marcus et al., 1993). Following Klein and Manning (2004), we begin with reference constituent parses and compare against deterministically derived dependencies: after pruning out all empty sub-trees, punctuation and terminals (tagged # and $) not pronounced where they appear, we drop all sentences with more than a prescribed number of tokens remaining and use automatic "head-percolation" rules (Collins, 1999) to convert the rest, as is standard practice. We experiment with WSJk (sentences with at most k tokens), for $1 \leq k \leq 45$, and Section 23 of WSJ$^\infty$ (all sentence lengths). We also evaluate on Brown100, similarly derived from the parsed portion of the Brown corpus (Francis and Kucera, 1979), as our held-out set. Figure 1 shows these corpora's sentence and token counts.

Proposed parse trees are judged on accuracy: a *directed score* is simply the overall fraction of correctly guessed dependencies. Let S be a set of sentences, with $|s|$ the number of terminals (to-

[1]The expected number of trials needed to get one Bernoulli(p) success is $n \sim$ Geometric(p), with $n \in \mathbb{Z}^+$, $\mathbb{P}(n) = (1-p)^{n-1}p$ and $\mathbb{E}(n) = p^{-1}$; MoM and MLE agree, $\hat{p} = (\#\text{ of successes})/(\#\text{ of trials})$.

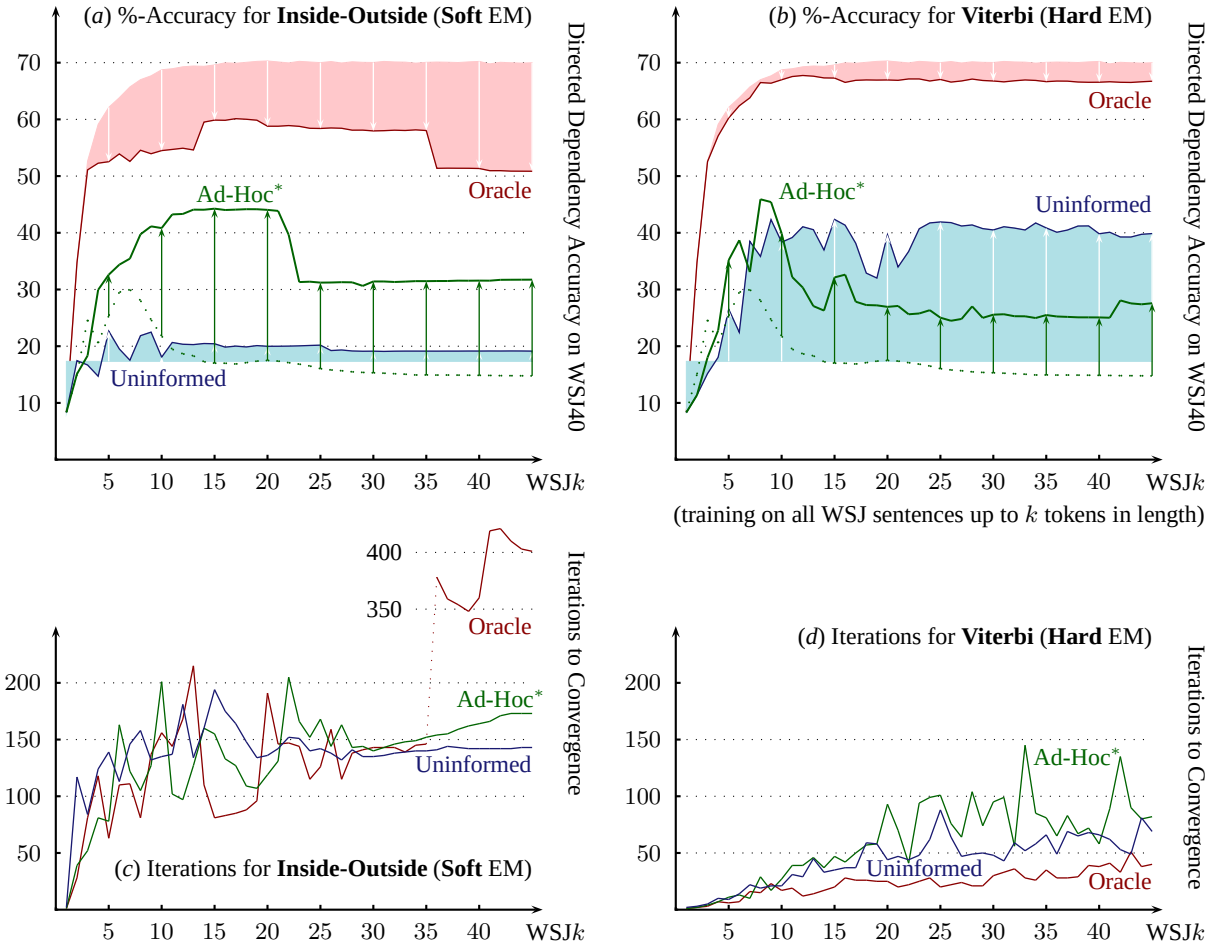


Figure 3: Directed dependency accuracies attained by the DMV, when trained on WSJk, smoothed, then tested against a fixed evaluation set, WSJ40, for three different initialization strategies (Spitkovsky et al., 2009). Red, green and blue graphs represent the supervised (maximum-likelihood oracle) initialization, a linguistically-biased initializer (Ad-Hoc*) and the uninformed (uniform) prior. Panel (b) shows results obtained with Viterbi training instead of classic EM — Panel (a), but is otherwise identical (in both, each of the 45 vertical slices captures five new experimental results and arrows connect starting performance with final accuracy, emphasizing the impact of learning). Panels (c) and (d) show the corresponding numbers of iterations until EM's convergence.

kens) for each $s \in S$. Denote by $T(s)$ the set of all dependency parse trees of s, and let $t_i(s)$ stand for the parent of token i, $1 \leq i \leq |s|$, in $t(s) \in T(s)$. Call the gold reference $t^*(s) \in T(s)$. For a given model of grammar, parameterized by θ, let $\hat{t}^\theta(s) \in T(s)$ be a (not necessarily unique) likeliest (also known as Viterbi) parse of s:

$$\hat{t}^\theta(s) \in \left\{ \arg\max_{t \in T(s)} \mathbb{P}_\theta(t) \right\};$$

then θ's directed accuracy on a reference set R is

$$100\% \cdot \frac{\sum_{s \in R} \sum_{i=1}^{|s|} 1_{\{\hat{t}_i^\theta(s) = t_i^*(s)\}}}{\sum_{s \in R} |s|}.$$

4 Experimental Setup and Results

Following Spitkovsky et al. (2009), we trained the DMV on data sets WSJ$\{1, \ldots, 45\}$ using three initialization strategies: (i) the uninformed uniform prior; (ii) a linguistically-biased initializer, Ad-Hoc*;[2] and (iii) an oracle — the supervised MLE solution. Standard training is without smoothing, iterating each run until successive changes in overall per-token cross-entropy drop below 2^{-20} bits.

We re-trained all models using Viterbi EM instead of inside-outside re-estimation, explored Laplace (add-one) smoothing during training, and experimented with hybrid initialization strategies.

[2] Ad-Hoc* is Spitkovsky et al.'s (2009) variation on Klein and Manning's (2004) "ad-hoc harmonic" completion.

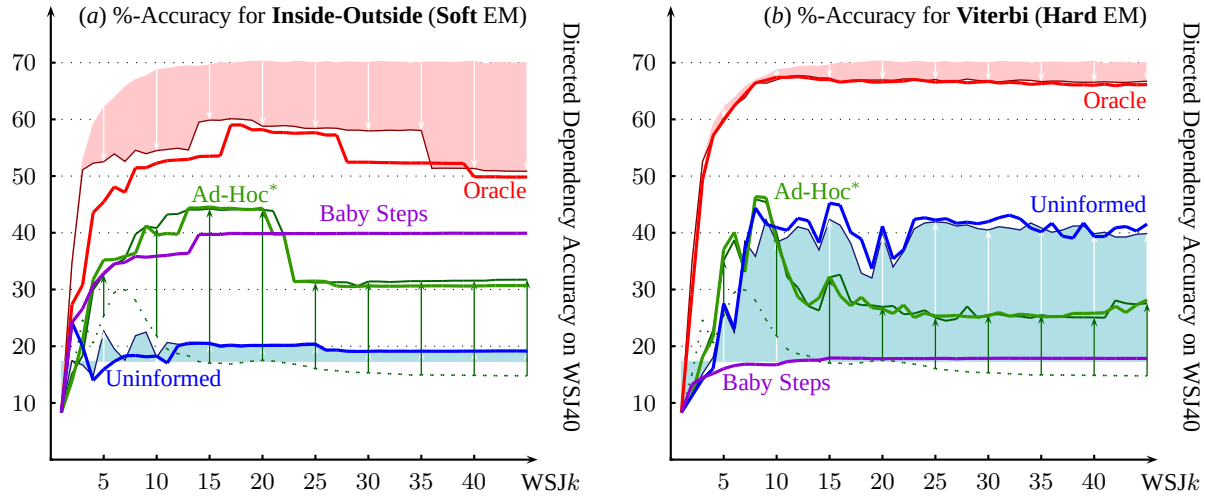


Figure 4: Superimposes directed accuracies attained by DMV models trained *with* Laplace smoothing (brightly-colored curves) over Figure 3(a,b); violet curves represent Baby Steps (Spitkovsky et al., 2009).

4.1 Result #1: Viterbi-Trained Models

The results of Spitkovsky et al. (2009), tested against WSJ40, are re-printed in Figure 3(a); our corresponding Viterbi runs appear in Figure 3(b).

We observe crucial differences between the two training modes for each of the three initialization strategies. Both algorithms walk away from the supervised maximum-likelihood solution; however, Viterbi EM loses at most a few points of accuracy (3.7% at WSJ40), whereas classic EM drops nearly twenty points (19.1% at WSJ45). In both cases, the single best unsupervised result is with good initialization, although Viterbi peaks earlier (45.9% at WSJ8) and in a narrower range (WSJ8-9) than classic EM (44.3% at WSJ15; WSJ13-20). The uniform prior never quite gets off the ground with classic EM but manages quite well under Viterbi training,[3] given sufficient data — it even beats the "clever" initializer everywhere past WSJ10. The "sweet spot" at WSJ15 — a neighborhood where both Ad-Hoc* and the oracle excel under classic EM — disappears with Viterbi. Furthermore, Viterbi does not degrade with more (complex) data, except with a biased initializer.

More than a simple efficiency hack, Viterbi EM actually improves performance. And its benefits to running times are also non-trivial: it not only skips computing the outside charts in every iteration but also converges (sometimes an order of magnitude) faster than classic EM (see Figure 3(c,d)).[4]

4.2 Result #2: Smoothed Models

Smoothing rarely helps classic EM and hurts in the case of oracle training (see Figure 4(a)). With Viterbi, supervised initialization suffers much less, the biased initializer is a wash, and the uninformed uniform prior generally gains a few points of accuracy, e.g., up 2.9% (from 42.4% to 45.2%, evaluated against WSJ40) at WSJ15 (see Figure 4(b)).

Baby Steps (Spitkovsky et al., 2009) — iterative re-training with increasingly more complex data sets, WSJ1, . . . ,WSJ45 — using smoothed Viterbi training fails miserably (see Figure 4(b)), due to Viterbi's poor initial performance at short sentences (possibly because of data sparsity and sensitivity to non-sentences — see examples in §7.3).

4.3 Result #3: State-of-the-Art Models

Simply training up smoothed Viterbi at WSJ15, using the uninformed uniform prior, yields 44.8% accuracy on Section 23 of WSJ$^\infty$, already beating previous state-of-the-art by 0.7% (see Table 1(*A*)).

Since both classic EM and Ad-Hoc* initializers work well with short sentences (see Figure 3(a)), it makes sense to use their pre-trained models to initialize Viterbi training, mixing the two strategies. We judged all Ad-Hoc* initializers against WSJ15 and found that the one for WSJ8 minimizes sentence-level cross-entropy (see Figure 5). This approach does not involve reference parse

[3] In a concurrently published related work, Cohen and Smith (2010) prove that the uniform-at-random initializer is a competitive starting M-step for Viterbi EM; our uninformed prior consists of uniform multinomials, seeding the E-step.

[4] For classic EM, the number of iterations to convergence appears sometimes inversely related to performance, giving credence to the notion of early termination as a regularizer.

Model	*Incarnation*		WSJ10	WSJ20	WSJ$^\infty$	
DMV	Bilingual Log-Normals (tie-verb-noun)	(Cohen and Smith, 2009)	62.0	48.0	42.2	Brown100
	Less is More (Ad-Hoc* @15)	(Spitkovsky et al., 2009)	56.2	48.2	44.1	43.3
A.	Smoothed Viterbi Training (@15), Initialized with the Uniform Prior		59.9	50.0	44.8	48.1
B.	A Good Initializer (Ad-Hoc*'s @8), Classically Pre-Trained (@15)		63.8	52.3	46.2	49.3
C.	Smoothed Viterbi Training (@15), Initialized with *B*		64.4	53.5	47.8	50.5
D.	Smoothed Viterbi Training (@45), Initialized with *C*		65.3	**53.8**	**47.9**	**50.8**
EVG	Smoothed (skip-head), Lexicalized	(Headden et al., 2009)	**68.8**			

Table 1: Accuracies on Section 23 of WSJ$\{10, 20, \infty\}$ and Brown100 for three recent state-of-the-art systems, our initializer, and smoothed Viterbi-trained runs that employ different initialization strategies.

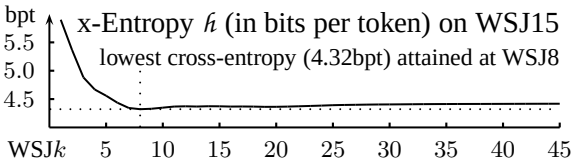


Figure 5: Sentence-level cross-entropy on WSJ15 for Ad-Hoc* initializers of WSJ$\{1, \ldots, 45\}$.

trees and is therefore still unsupervised. Using the Ad-Hoc* initializer based on WSJ8 to seed classic training at WSJ15 yields a further 1.4% gain in accuracy, scoring 46.2% on WSJ$^\infty$ (see Table 1(*B*)).

This good initializer boosts accuracy attained by smoothed Viterbi at WSJ15 to 47.8% (see Table 1(*C*)). Using its solution to re-initialize training at WSJ45 gives a tiny further improvement (0.1%) on Section 23 of WSJ$^\infty$ but bigger gains on WSJ10 (0.9%) and WSJ20 (see Table 1(*D*)).

Our results generalize. Gains due to smoothed Viterbi training and favorable initialization carry over to Brown100 — accuracy improves by 7.5% over previous published numbers (see Table 1).[5]

5 Discussion of Experimental Results

The DMV has no parameters to capture syntactic relationships beyond local trees, e.g., agreement. Spitkovsky et al. (2009) suggest that classic EM breaks down as sentences get longer precisely because the model makes unwarranted independence assumptions. They hypothesize that the DMV reserves too much probability mass for what should be unlikely productions. Since EM faithfully allocates such re-distributions across the possible parse trees, once sentences grow sufficiently long, this process begins to deplete what began as likelier structures. But medium lengths avoid a flood of exponentially-confusing longer sentences (and the sparseness of unrepresentative shorter ones).[6]

Our experiments corroborate this hypothesis. First of all, Viterbi manages to hang on to supervised solutions much better than classic EM. Second, Viterbi does not universally degrade with more (complex) training sets, except with a biased initializer. And third, Viterbi learns poorly from small data sets of short sentences (WSJk, $k < 5$).

Viterbi may be better suited to unsupervised grammar induction compared with classic EM, but neither is sufficient, by itself. Both algorithms abandon good solutions and make no guarantees with respect to extrinsic performance. Unfortunately, these two approaches share a deep flaw.

6 Related Work on Improper Objectives

It is well-known that maximizing likelihood may, in fact, degrade accuracy (Pereira and Schabes, 1992; Elworthy, 1994; Merialdo, 1994). de Marcken (1995) showed that classic EM suffers from a fatal attraction towards deterministic grammars and suggested a Viterbi training scheme as a remedy. Liang and Klein's (2008) analysis of errors in unsupervised learning began with the inappropriateness of the likelihood objective (approximation), explored problems of data sparsity (estimation) and focused on EM-specific issues related to non-convexity (identifiability and optimization).

Previous literature primarily relied on experimental evidence. de Marcken's analytical result is an exception but pertains only to EM-specific local attractors. Our analysis confirms his intuitions and moreover shows that there can be *global* preferences for deterministic grammars — problems that would persist with tractable optimization. We prove that there is a fundamental disconnect between objective functions even when likelihood is a reasonable metric and training data are infinite.

[5] In a sister paper, Spitkovsky et al. (2010) improve performance by incorporating parsing constraints harvested from the web into Viterbi training; nevertheless, results presented in this paper remain the best of models trained purely on WSJ.

[6] Klein and Manning (2004) originally trained the DMV on WSJ10 and Gillenwater et al. (2009) found it useful to discard data from WSJ3, which is mostly incomplete sentences.

7 Proofs (by Construction)

There is a subtle distinction between *three* different probability distributions that arise in parsing, each of which can be legitimately termed "likelihood" — the mass that a particular model assigns to (i) highest-scoring (Viterbi) parse trees; (ii) the correct (gold) reference trees; and (iii) the sentence strings (sums over all derivations). A classic unsupervised parser trains to optimize the third, makes actual parsing decisions according to the first, and is evaluated against the second. There are several potential disconnects here. First of all, the true generative model θ^* may not yield the largest margin separations for discriminating between gold parse trees and next best alternatives; and second, θ^* may assign sub-optimal mass to string probabilities. There is no reason why an optimal estimate $\hat{\theta}$ should make the best parser or coincide with a peak of an unsupervised objective.

7.1 The Three Likelihood Objectives

A supervised parser finds the "best" parameters $\hat{\theta}$ by maximizing the likelihood of all reference *structures* $t^*(s)$ — the product, over all sentences, of the probabilities that it assigns to each such tree:

$$\hat{\theta}_{\text{SUP}} = \arg\max_{\theta} \mathcal{L}(\theta) = \arg\max_{\theta} \prod_{s} \mathbb{P}_{\theta}(t^*(s)).$$

For the DMV, this objective function is convex — its unique peak is easy to find and should match the true distribution θ^* given enough data, barring practical problems caused by numerical instability and inappropriate independence assumptions. It is often easier to work in log-probability space:

$$\begin{aligned} \hat{\theta}_{\text{SUP}} &= \arg\max_{\theta} \log \mathcal{L}(\theta) \\ &= \arg\max_{\theta} \textstyle\sum_s \log \mathbb{P}_{\theta}(t^*(s)). \end{aligned}$$

Cross-entropy, measured in bits per token (bpt), offers an interpretable proxy for a model's quality:

$$\hbar(\theta) = -\frac{\sum_s \lg \mathbb{P}_{\theta}(t^*(s))}{\sum_s |s|}.$$

Clearly, $\arg\max_{\theta} \mathcal{L}(\theta) = \hat{\theta}_{\text{SUP}} = \arg\min_{\theta} \hbar(\theta)$.

Unsupervised parsers cannot rely on references and attempt to jointly maximize the probability of each *sentence* instead, summing over the probabilities of all possible trees, according to a model θ:

$$\hat{\theta}_{\text{UNS}} = \arg\max_{\theta} \sum_{s} \log \underbrace{\sum_{t \in T(s)} \mathbb{P}_{\theta}(t)}_{\mathbb{P}_{\theta}(s)}.$$

This objective function is not convex and in general does not have a unique peak, so in practice one usually settles for $\tilde{\theta}_{\text{UNS}}$ — a fixed point. There is no reason why $\hat{\theta}_{\text{SUP}}$ should agree with $\hat{\theta}_{\text{UNS}}$, which is in turn (often badly) approximated by $\tilde{\theta}_{\text{UNS}}$, in our case using EM. A logical alternative to maximizing the probability of sentences is to maximize the probability of the most likely parse trees instead:[7]

$$\hat{\theta}_{\text{VIT}} = \arg\max_{\theta} \sum_{s} \log \mathbb{P}_{\theta}(\hat{t}^{\theta}(s)).$$

This 1-best approximation similarly arrives at $\tilde{\theta}_{\text{VIT}}$, with no claims of optimality. Each next model is re-estimated as if supervised by reference parses.

7.2 A Warm-Up Case: Accuracy vs. $\hat{\theta}_{\text{SUP}} \neq \theta^*$

A simple way to derail accuracy is to maximize the likelihood of an incorrect model, e.g., one that makes false independence assumptions. Consider fitting the DMV to a contrived distribution — two equiprobable structures over identical three-token sentences from a unary vocabulary $\{ⓐ\}$:

(i) ⓐ ⓐ ⓐ; (ii) ⓐ ⓐ ⓐ.

There are six tokens and only two have children on any given side, so adjacent stopping MLEs are:

$$\hat{\mathbb{P}}_{\texttt{STOP}}(ⓐ, \texttt{L}, \texttt{T}) = \hat{\mathbb{P}}_{\texttt{STOP}}(ⓐ, \texttt{R}, \texttt{T}) = 1 - \frac{2}{6} = \frac{2}{3}.$$

The rest of the estimated model is deterministic:

$$\hat{\mathbb{P}}_{\texttt{ATTACH}}(\diamondsuit, \texttt{L}, ⓐ) = \hat{\mathbb{P}}_{\texttt{ATTACH}}(ⓐ, *, ⓐ) = 1$$

$$\text{and } \hat{\mathbb{P}}_{\texttt{STOP}}(ⓐ, *, \texttt{F}) = 1,$$

since all dependents are ⓐ and every one is an only child. But the DMV generates left- and right-attachments independently, allowing a third parse:

(iii) ⓐ ⓐ ⓐ.

It also cannot capture the fact that all structures are local (or that all dependency arcs point in the same direction), admitting two additional parse trees:

(iv) ⓐ ⓐ ⓐ; (v) ⓐ ⓐ ⓐ.

Each possible structure must make four (out of six) adjacent stops, incurring identical probabilities:

$$\hat{\mathbb{P}}_{\texttt{STOP}}(ⓐ, *, \texttt{T})^4 \times (1 - \hat{\mathbb{P}}_{\texttt{STOP}}(ⓐ, *, \texttt{T}))^2 = \frac{2^4}{3^6}.$$

[7] It is also possible to use k-best Viterbi, with $k > 1$.

Thus, the MLE model does not break symmetry and rates each of the five parse trees as equally likely. Therefore, its expected per-token accuracy is 40%. Average overlaps between structures (i-v) and answers (i,ii) are (i) 100% or 0; (ii) 0 or 100%; and (iii,iv,v) $33.\overline{3}\%$: $(3+3)/(5\times 3) = 2/5 = 0.4$.

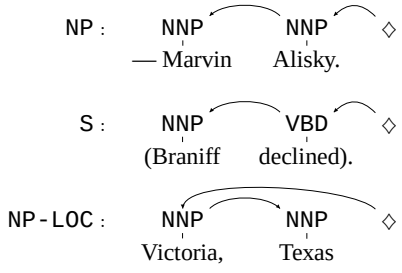
A decoy model without left- or right-branching, i.e., $\tilde{\mathbb{P}}_{\texttt{STOP}}(ⓐ, \texttt{L}, \texttt{T}) = 1$ or $\tilde{\mathbb{P}}_{\texttt{STOP}}(ⓐ, \texttt{R}, \texttt{T}) = 1$, would assign zero probability to some of the training data. It would be forced to parse every instance of ⓐⓐⓐ either as (i) or as (ii), deterministically. Nevertheless, it would attain a higher per-token accuracy of 50%. (Judged on exact matches, at the granularity of whole trees, the decoy's guaranteed 50% accuracy clobbers the MLE's expected 20%.)

Our toy data set could be replicated n-fold without changing the analysis. This confirms that, even in the absence of estimation errors or data sparsity, there can be a fundamental disconnect between likelihood and accuracy, if the model is wrong.[8]

7.3 A Subtler Case: $\theta^* = \hat{\theta}_{\text{SUP}}$ vs. $\hat{\theta}_{\text{UNS}}$ vs. $\hat{\theta}_{\text{VIT}}$

We now prove that, even with the *right* model, mismatches between the different objective likelihoods can also handicap the truth. Our calculations are again exact, so there are no issues with numerical stability. We work with a set of parameters θ^* already factored by the DMV, so that its problems could not be blamed on invalid independence assumptions. Yet we are able to find another impostor distribution $\tilde{\theta}$ that outshines $\hat{\theta}_{\text{SUP}} = \theta^*$ on both unsupervised metrics, which proves that the true models $\hat{\theta}_{\text{SUP}}$ and θ^* are not globally optimal, as judged by the two surrogate objective functions.

This next example is organic. We began with WSJ10 and confirmed that classic EM abandons the supervised solution. We then iteratively discarded large portions of the data set, so long as the remainder maintained the (un)desired effect — EM walking away from its $\hat{\theta}_{\text{SUP}}$. This procedure isolated such behavior, arriving at a minimal set:



This kernel is tiny, but, as before, our analysis is invariant to n-fold replication: the problem cannot be explained away by a small training size — it persists even in infinitely large data sets. And so, we consider three reference parse trees for two-token sentences over a binary vocabulary $\{ⓐ, ⓩ\}$:

(i) ⓐ ⓐ; (ii) ⓐ ⓩ; (iii) ⓐ ⓐ.

One third of the time, ⓩ is the head; only ⓐ can be a child; and only ⓐ has right-dependents. Trees (i)-(iii) are the only two-terminal parses generated by the model and are equiprobable. Thus, these sentences are representative of a length-two restriction of everything generated by the true θ^*:

$$\mathbb{P}_{\texttt{ATTACH}}(\diamondsuit, \texttt{L}, ⓐ) = \frac{2}{3} \text{ and } \mathbb{P}_{\texttt{STOP}}(ⓐ, *, \texttt{T}) = \frac{4}{5},$$

since ⓐ is the head two out of three times, and since only one out of five ⓐ's attaches a child on either side. Elsewhere, the model is deterministic:

$$\mathbb{P}_{\texttt{STOP}}(ⓩ, \texttt{L}, \texttt{T}) = 0;$$

$$\mathbb{P}_{\texttt{STOP}}(*, *, \texttt{F}) = \mathbb{P}_{\texttt{STOP}}(ⓩ, \texttt{R}, \texttt{T}) = 1;$$

$$\mathbb{P}_{\texttt{ATTACH}}(ⓐ, *, ⓐ) = \mathbb{P}_{\texttt{ATTACH}}(ⓩ, \texttt{L}, ⓐ) = 1.$$

Contrast the optimal estimate $\hat{\theta}_{\text{SUP}} = \theta^*$ with the decoy *fixed point*[9] $\tilde{\theta}$ that is identical to θ^*, except

$$\tilde{\mathbb{P}}_{\texttt{STOP}}(ⓐ, \texttt{L}, \texttt{T}) = \frac{3}{5} \text{ and } \tilde{\mathbb{P}}_{\texttt{STOP}}(ⓐ, \texttt{R}, \texttt{T}) = 1.$$

The probability of stopping is now 3/5 on the left and 1 on the right, instead of 4/5 on both sides — $\tilde{\theta}$ disallows ⓐ's right-dependents but preserves its overall fertility. The probabilities of leaves ⓐ (no children), under the models $\hat{\theta}_{\text{SUP}}$ and $\tilde{\theta}$, are:

$$\hat{\mathbb{P}}(ⓐ) = \hat{\mathbb{P}}_{\texttt{STOP}}(ⓐ, \texttt{L}, \texttt{T}) \times \hat{\mathbb{P}}_{\texttt{STOP}}(ⓐ, \texttt{R}, \texttt{T}) = \left(\frac{4}{5}\right)^2$$

$$\text{and } \tilde{\mathbb{P}}(ⓐ) = \tilde{\mathbb{P}}_{\texttt{STOP}}(ⓐ, \texttt{L}, \texttt{T}) \times \tilde{\mathbb{P}}_{\texttt{STOP}}(ⓐ, \texttt{R}, \texttt{T}) = \frac{3}{5}.$$

And the probabilities of, e.g., structure ⓐ ⓩ, are:

$$\hat{\mathbb{P}}_{\texttt{ATTACH}}(\diamondsuit, \texttt{L}, ⓩ) \times \hat{\mathbb{P}}_{\texttt{STOP}}(ⓩ, \texttt{R}, \texttt{T})$$

$$\times (1 - \hat{\mathbb{P}}_{\texttt{STOP}}(ⓩ, \texttt{L}, \texttt{T})) \times \hat{\mathbb{P}}_{\texttt{STOP}}(ⓩ, \texttt{L}, \texttt{F})$$

$$\times \hat{\mathbb{P}}_{\texttt{ATTACH}}(ⓩ, \texttt{L}, ⓐ) \times \hat{\mathbb{P}}(ⓐ)$$

[8] And as George Box quipped, "Essentially, all models are wrong, but some are useful" (Box and Draper, 1987, p. 424).

[9] The model estimated from the parse trees induced by $\tilde{\theta}$ over the three sentences is again $\tilde{\theta}$, for both soft and hard EM.

$$= \hat{\mathbb{P}}_{\texttt{ATTACH}}(\diamondsuit, \texttt{L}, ⓩ) \times \hat{\mathbb{P}}(ⓐ) = \frac{1}{3} \cdot \frac{16}{25}$$

$$\text{and } \tilde{\mathbb{P}}_{\texttt{ATTACH}}(\diamondsuit, \texttt{L}, ⓩ) \times \tilde{\mathbb{P}}(ⓐ) = \frac{1}{3} \cdot \frac{3}{5}.$$

Similarly, the probabilities of all four possible parse trees for the two distinct sentences, ⓐⓐ and ⓐⓩ, under the two models, $\hat{\theta}_{\text{SUP}} = \theta^*$ and $\tilde{\theta}$, are:

	$\hat{\theta}_{\text{SUP}} = \theta^*$	$\tilde{\theta}$
ⓐ ⓩ	$\frac{1}{3}\left(\frac{16}{25}\right) = \frac{16}{75} = \mathbf{0.21\overline{3}}$	$\frac{1}{3}\left(\frac{3}{5}\right) = \frac{1}{5} = \mathbf{0.2}$
ⓐ ⓩ	$\mathbf{0}$	$\mathbf{0}$
ⓐ ⓐ	$\frac{2}{3}\left(\frac{4}{5}\right)\left(1-\frac{4}{5}\right)\left(\frac{16}{25}\right) = \frac{128}{1875} = \mathbf{0.0682\overline{6}}$	$\frac{2}{3}\left(1-\frac{3}{5}\right)\left(\frac{3}{5}\right) = \frac{4}{25} = \mathbf{0.16}$
ⓐ ⓐ	$\mathbf{0.0682\overline{6}}$	$\mathbf{0}$

To the three *true parses*, $\hat{\theta}_{\text{SUP}}$ assigns probability $\left(\frac{16}{75}\right)\left(\frac{128}{1875}\right)^2 \approx 0.0009942$ — about 1.66bpt; $\tilde{\theta}$ leaves zero mass for (iii), corresponding to a larger (infinite) cross-entropy, consistent with theory.

So far so good, but if asked for *best* (Viterbi) *parses*, $\hat{\theta}_{\text{SUP}}$ could still produce the actual trees, whereas $\tilde{\theta}$ would happily parse sentences of (iii) and (i) the same, perceiving a joint probability of $(0.2)(0.16)^2 = 0.00512$ — just 1.27bpt, appearing to outperform $\hat{\theta}_{\text{SUP}} = \theta^*$! Asked for *sentence probabilities*, $\tilde{\theta}$ would remain unchanged (it parses each sentence unambiguously), but $\hat{\theta}_{\text{SUP}}$ would aggregate to $\left(\frac{16}{75}\right)\left(2 \cdot \frac{128}{1875}\right)^2 \approx 0.003977$, improving to 1.33bpt, but still noticeably "worse" than $\tilde{\theta}$.

Despite leaving zero probability to the truth, $\tilde{\theta}$ beats θ^* on both surrogate metrics, globally. This seems like an egregious error. Judged by (extrinsic) accuracy, $\tilde{\theta}$ still holds its own: it gets four directed edges from predicting parse trees (i) and (ii) completely right, but none of (iii) — a solid 66.7%. Subject to tie-breaking, θ^* is equally likely to get (i) and/or (iii) entirely right or totally wrong (they are indistinguishable): it could earn a perfect 100%, tie $\tilde{\theta}$, or score a low 33.3%, at 1:2:1 odds, respectively — same as $\tilde{\theta}$'s deterministic 66.7% accuracy, in expectation, but with higher variance.

8 Discussion of Theoretical Results

Daumé et al. (2009) questioned the benefits of using exact models in approximate inference. In our case, the model already makes strong simplifying assumptions *and* the objective is also incorrect. It makes sense that Viterbi EM sometimes works, since an approximate wrong "solution" *could*, by chance, be better than one that is exactly wrong.

One reason why Viterbi EM may work well is that *its* score is used in selecting actual output parse trees. Wainwright (2006) provided strong theoretical and empirical arguments for using the same approximate inference method in training as in performing predictions for a learned model. He showed that if inference involves an approximation, then using the same approximate method to train the model gives even better performance guarantees than exact training methods. If our task were not parsing but language modeling, where the relevant score is the sum of the probabilities over individual derivations, perhaps classic EM would not be doing as badly, compared to Viterbi.

Viterbi training is not only faster and more accurate but also free of inside-outside's recursion constraints. It therefore invites more flexible modeling techniques, including discriminative, feature-rich approaches that target *conditional* likelihoods, essentially via (unsupervised) self-training (Clark et al., 2003; Ng and Cardie, 2003; McClosky et al., 2006a; McClosky et al., 2006b, *inter alia*).

Such "learning by doing" approaches may be relevant to understanding human language acquisition, as children frequently find themselves forced to interpret a sentence in order to interact with the world. Since most models of *human* probabilistic parsing are massively pruned (Jurafsky, 1996; Chater et al., 1998; Lewis and Vasishth, 2005, *inter alia*), the serial nature of Viterbi EM — or the very limited parallelism of k-best Viterbi — may be more appropriate in modeling this task than the fully-integrated inside-outside solution.

9 Conclusion

Without a known objective, as in unsupervised learning, correct exact optimization becomes impossible. In such cases, approximations, although liable to pass over a true optimum, may achieve faster convergence and still *improve* performance. We showed that this is the case with Viterbi training, a cheap alternative to inside-outside re-estimation, for unsupervised dependency parsing.

We explained why Viterbi EM may be particularly well-suited to learning from longer sentences, in addition to any general benefits to synchronizing approximation methods across learning and inference. Our best algorithm is simpler and an order of magnitude faster than classic EM. It achieves state-of-the-art performance: 3.8% higher accuracy than previous published best

results on Section 23 (all sentences) of the Wall Street Journal corpus. This improvement generalizes to the Brown corpus, our held-out evaluation set, where the same model registers a 7.5% gain.

Unfortunately, approximations alone do not bridge the real gap between objective functions. This deeper issue could be addressed by drawing parsing constraints (Pereira and Schabes, 1992) from specific applications. One example of such an approach, tied to machine translation, is synchronous grammars (Alshawi and Douglas, 2000). An alternative — observing constraints induced by hyper-text mark-up, harvested from the web — is explored in a sister paper (Spitkovsky et al., 2010), published concurrently.

Acknowledgments

Partially funded by NSF award IIS-0811974 and by the Air Force Research Laboratory (AFRL), under prime contract no. FA8750-09-C-0181; first author supported by the Fannie & John Hertz Foundation Fellowship. We thank Angel X. Chang, Mengqiu Wang and the anonymous reviewers for many helpful comments on draft versions of this paper.

References

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*.

G. E. P. Box and N. R. Draper. 1987. *Empirical Model-Building and Response Surfaces*. John Wiley.

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. 1993. *Statistical Language Learning*. MIT Press.

N. Chater, M. J. Crocker, and M. J. Pickering. 1998. The rational analysis of inquiry: The case of parsing. In M. Oaksford and N. Chater, editors, *Rational Models of Cognition*. Oxford University Press.

S. Clark, J. Curran, and M. Osborne. 2003. Bootstrapping POS-taggers using unlabelled data. In *Proc. of CoNLL*.

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

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

H. Daumé, III, J. Langford, and D. Marcu. 2009. Search-based structured prediction. *Machine Learning*, 75(3).

C. de Marcken. 1995. Lexical heads, phrase structure and the induction of grammar. In *WVLC*.

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.

J. Gillenwater, K. Ganchev, J. Graça, B. Taskar, and F. Pereira. 2009. Sparsity in grammar induction. In *NIPS: Grammar Induction, Representation of Language and Language Learning*.

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

D. Jurafsky. 1996. A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science*, 20.

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

R. L. Lewis and S. Vasishth. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29.

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. 2006a. Effective self-training for parsing. In *Proc. of NAACL-HLT*.

D. McClosky, E. Charniak, and M. Johnson. 2006b. Reranking and self-training for parser adaptation. In *Proc. of COLING-ACL*.

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

V. Ng and C. Cardie. 2003. Weakly supervised natural language learning without redundant views. In *Proc. of HLT-NAACL*.

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

V. I. Spitkovsky, H. Alshawi, and D. Jurafsky. 2009. Baby Steps: How "Less is More" in unsupervised dependency parsing. In *NIPS: Grammar Induction, Representation of Language and Language Learning*.

V. I. Spitkovsky, D. Jurafsky, and H. Alshawi. 2010. Profiting from mark-up: Hyper-text annotations for guided parsing. In *Proc. of ACL*.

M. J. Wainwright. 2006. Estimating the "wrong" graphical model: Benefits in the computation-limited setting. *Journal of Machine Learning Research*, 7.