Profiting from Mark-Up: Hyper-Text Annotations for Guided Parsing

Valentin I. Spitkovsky
Computer Science Department
Stanford University and Google Inc.
valentin@google.com

Daniel Jurafsky
Departments of Linguistics and
Computer Science, Stanford University
jurafsky@stanford.edu

Hiyan Alshawi
Google Inc.
hiyan@google.com

Abstract

We show how web mark-up can be used to improve unsupervised dependency parsing. Starting from raw bracketings of four common HTML tags (anchors, bold, italics and underlines), we refine approximate partial phrase boundaries to yield accurate parsing constraints. Conversion procedures fall out of our linguistic analysis of a newly available million-word hyper-text corpus. We demonstrate that derived constraints aid grammar induction by training Klein and Manning's Dependency Model with Valence (DMV) on this data set: parsing accuracy on Section 23 (all sentences) of the Wall Street Journal corpus jumps to 50.4%, beating previous state-of-the-art by more than 5%. Web-scale experiments show that the DMV, perhaps because it is unlexicalized, does not benefit from orders of magnitude more annotated but noisier data. Our model, trained on a single blog, generalizes to 53.3% accuracy out-of-domain, against the Brown corpus — nearly 10% higher than the previous published best. The fact that web mark-up strongly correlates with syntactic structure may have broad applicability in NLP.

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 of this problem that targets dependencies and assumes partial annotation — sentence boundaries and part-of-speech (POS) tagging — has received much attention. Klein and Manning (2004) were 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; Spitkovsky et al., 2010a) are rooted in their Dependency Model with Valence (DMV), still trained using variants of EM.

Pereira and Schabes (1992) outlined three major problems with classic EM, applied to a related problem, constituent parsing. They extended classic inside-outside re-estimation (Baker, 1979) to respect any bracketing constraints included with a training corpus. This conditioning on partial parses addressed all three problems, leading to: (i) linguistically reasonable constituent boundaries and induced grammars more likely to agree with qualitative judgments of sentence structure, which is underdetermined by unannotated text; (ii) fewer iterations needed to reach a good grammar, countering convergence properties that sharply deteriorate with the number of non-terminal symbols, due to a proliferation of local maxima; and (iii) better (in the best case, linear) time complexity per iteration, versus running time that is ordinarily cubic in both sentence length *and* the total number of non-terminals, rendering sufficiently large grammars computationally impractical. Their algorithm sometimes found good solutions from bracketed corpora but not from raw text, supporting the view that purely unsupervised, self-organizing inference methods can miss the trees for the forest of distributional regularities. This was a promising break-through, but the problem of whence to get partial bracketings was left open.

We suggest mining partial bracketings from a cheap and abundant natural language resource: the hyper-text mark-up that annotates web-pages. For example, consider that anchor text can match linguistic constituents, such as verb phrases, exactly:

..., whereas McCain is secure on the topic, Obama <a>[VP worries about winning the pro-Israel vote].

To validate this idea, we created a new data set, novel in combining a real blog's raw HTML with tree-bank-like constituent structure parses, gener-

ated automatically. Our linguistic analysis of the most prevalent tags (anchors, bold, italics and underlines) over its 1M$^+$ words reveals a strong connection between syntax and mark-up (all of our examples draw from this corpus), inspiring several simple techniques for automatically deriving parsing constraints. Experiments with both hard and more flexible constraints, as well as with different styles and quantities of annotated training data — the blog, web news and the web itself, confirm that mark-up-induced constraints consistently improve (otherwise unsupervised) dependency parsing.

2 Intuition and Motivating Examples

It is natural to expect hidden structure to seep through when a person annotates a sentence. As it happens, a non-trivial fraction of the world's population routinely annotates text diligently, if only partially and informally.[1] They inject hyper-links, vary font sizes, and toggle colors and styles, using mark-up technologies such as HTML and XML.

As noted, web annotations can be indicative of phrase boundaries, e.g., in a complicated sentence:

> In 1998, however, as I <a>[VP established in [NP *The New Republic*] and Bill Clinton just <a>[VP confirmed in his memoirs], Netanyahu changed his mind and ...

In doing so, mark-up sometimes offers useful cues even for low-level tokenization decisions:

> [NP [NP Libyan ruler]
> <a>[NP Mu'ammar al-Qaddafi] referred to ...

```
(NP (ADJP (NP (JJ Libyan) (NN ruler))
          (JJ Mu))
    (`` `) (NN ammar) (NNS al-Qaddafi))
```

Above, a backward quote in an Arabic name confuses the Stanford parser.[2] Yet mark-up lines up with the broken noun phrase, signals cohesion, and moreover sheds light on the internal structure of a compound. As Vadas and Curran (2007) point out, such details are frequently omitted even from manually compiled tree-banks that err on the side of flat annotations of base-NPs.

Admittedly, not all boundaries between HTML tags and syntactic constituents match up nicely:

> ..., but [S [NP the <a>*Toronto Star*][VP reports [NP this][PP in the softest possible way],[S stating only that ...]]]

Combining parsing with mark-up may not be straight-forward, but there is hope: even above, one of each nested tag's boundaries aligns; and *Toronto Star*'s neglected determiner could be forgiven, certainly within a dependency formulation.

3 A High-Level Outline of Our Approach

Our idea is to implement the DMV (Klein and Manning, 2004) — a standard unsupervised grammar inducer. But instead of learning the unannotated test set, we train with text that contains web mark-up, using various ways of converting HTML into parsing constraints. We still test on WSJ (Marcus et al., 1993), in the standard way, and also check generalization against a hidden data set — the Brown corpus (Francis and Kucera, 1979). Our parsing constraints come from a blog — a new corpus we created, the web and news (see Table 1 for corpora's sentence and token counts).

To facilitate future work, we make the final models and our manually-constructed blog data publicly available.[3] Although we are unable to share larger-scale resources, our main results should be reproducible, as both linguistic analysis and our best model rely exclusively on the blog.

Corpus	*Sentences*	*POS Tokens*
WSJ$^\infty$	49,208	1,028,347
Section 23	2,353	48,201
WSJ45	48,418	986,830
WSJ15	15,922	163,715
Brown100	24,208	391,796
BLOG$_p$	57,809	1,136,659
BLOG$_t$45	56,191	1,048,404
BLOG$_t$15	23,214	212,872
NEWS45	2,263,563,078	32,119,123,561
NEWS15	1,433,779,438	11,786,164,503
WEB45	8,903,458,234	87,269,385,640
WEB15	7,488,669,239	55,014,582,024

Table 1: Sizes of corpora derived from WSJ and Brown, as well as those we collected from the web.

4 Data Sets for Evaluation and Training

The appeal of unsupervised parsing lies in its ability to learn from surface text alone; but (intrinsic) evaluation still requires parsed sentences. Following Klein and Manning (2004), we begin with reference constituent parses and compare against deterministically derived dependencies: after pruning out all empty subtrees, 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.

[1] Even when (American) grammar schools lived up to their name, they only taught dependencies. This was back in the days before constituent grammars were invented.

[2] http://nlp.stanford.edu:8080/parser/

[3] http://cs.stanford.edu/~valentin/

Length Cutoff	*Marked Sentences*	*POS Tokens*	*Bracketings*	
			All	*Multi-Token*
0	6,047	1,136,659	7,731	6,015
1	*of 57,809*	149,483	7,731	6,015
2	4,934	124,527	6,482	6,015
3	3,295	85,423	4,476	4,212
4	2,103	56,390	2,952	2,789
5	1,402	38,265	1,988	1,874
6	960	27,285	1,365	1,302
7	692	19,894	992	952
8	485	14,528	710	684
9	333	10,484	499	479
10	245	7,887	365	352
15	42	1,519	65	63
20	13	466	20	20
25	6	235	10	10
30	3	136	6	6
40	0	0	0	0

Table 2: Counts of sentences, tokens and (unique) bracketings for BLOG_p, restricted to only those sentences having at least one bracketing no shorter than the length cutoff (but shorter than the sentence).

Our primary reference sets are derived from the Penn English Treebank's Wall Street Journal portion (Marcus et al., 1993): WSJ45 (sentences with fewer than 46 tokens) and Section 23 of WSJ^{∞} (all sentence lengths). We also evaluate on Brown100, similarly derived from the parsed portion of the Brown corpus (Francis and Kucera, 1979). While we use WSJ45 and WSJ15 to train baseline models, the bulk of our experiments is with web data.

4.1 A News-Style Blog: Daniel Pipes

Since there was no corpus overlaying syntactic structure with mark-up, we began constructing a new one by downloading articles[4] from a news-style blog. Although limited to a single genre — political opinion, `danielpipes.org` is clean, consistently formatted, carefully edited and larger than WSJ (see Table 1). Spanning decades, Pipes' editorials are mostly in-domain for POS taggers and tree-bank-trained parsers; his recent (internet-era) entries are thoroughly cross-referenced, conveniently providing just the mark-up we hoped to study via uncluttered (printer-friendly) HTML.[5]

After extracting moderately clean text and mark-up locations, we used MxTerminator (Reynar and Ratnaparkhi, 1997) to detect sentence boundaries. This initial automated pass begot multiple rounds of various semi-automated clean-ups that involved fixing sentence breaking, modifying parser-unfriendly tokens, converting HTML entities and non-ASCII text, correcting typos, and so on. After throwing away annotations of fractional words (e.g., *basmachi*s) and tokens (e.g., *Sesame Street*-like), we broke up all mark-up that crossed sentence boundaries (i.e., loosely speaking, replaced constructs like <u>...]\[s...</u> with <u>...</u>]\[s <u>...</u>) and discarded any tags left covering entire sentences.

We finalized two versions of the data: BLOG_t, tagged with the Stanford tagger (Toutanova and Manning, 2000; Toutanova et al., 2003),[6] and BLOG_p, parsed with Charniak's parser (Charniak, 2001; Charniak and Johnson, 2005).[7] The reason for this dichotomy was to use state-of-the-art parses to analyze the relationship between syntax and mark-up, yet to prevent jointly tagged (and non-standard `AUX[G]`) POS sequences from interfering with our (otherwise unsupervised) training.[8]

4.2 Scaled up *Quantity*: The (English) Web

We built a large (see Table 1) but messy data set, WEB — English-looking web-pages, pre-crawled by a search engine. To avoid machine-generated spam, we excluded low quality sites flagged by the indexing system. We kept only sentence-like runs of words (satisfying punctuation and capitalization constraints), POS-tagged with TnT (Brants, 2000).

4.3 Scaled up *Quality*: (English) Web News

In an effort to trade quantity for quality, we constructed a smaller, potentially cleaner data set, NEWS. We reckoned editorialized content would lead to fewer extracted non-sentences. Perhaps surprisingly, NEWS is less than an order of magnitude smaller than WEB (see Table 1); in part, this is due to less aggressive filtering — we trust sites approved by the human editors at Google News.[9] In all other respects, our pre-processing of NEWS pages was identical to our handling of WEB data.

[4] `http://danielpipes.org/art/year/all`

[5] `http://danielpipes.org/article_print.php?id=...`

[6] `http://nlp.stanford.edu/software/stanford-postagger-2008-09-28.tar.gz`

[7] `ftp://ftp.cs.brown.edu/pub/nlparser/parser05Aug16.tar.gz`

[8] However, since many taggers are themselves trained on manually parsed corpora, such as WSJ, no parser that relies on external POS tags could be considered truly unsupervised; for a *fully* unsupervised example, see Seginer's (2007) CCL parser, available at `http://www.seggu.net/ccl/`

[9] `http://news.google.com/`

5 Linguistic Analysis of Mark-Up

Is there a connection between mark-up and syntactic structure? Previous work (Barr et al., 2008) has only examined search engine queries, showing that they consist predominantly of short noun phrases. If web mark-up shared a similar characteristic, it might not provide sufficiently disambiguating cues to syntactic structure: HTML tags could be too short (e.g., singletons like "click <a>here") or otherwise unhelpful in resolving truly difficult ambiguities (such as PP-attachment). We began simply by counting various basic events in BLOG$_p$.

	Count	*POS Sequence*	*Frac*	*Sum*
1	1,242	NNP NNP	16.1%	
2	643	NNP	8.3	24.4
3	419	NNP NNP NNP	5.4	29.8
4	414	NN	5.4	35.2
5	201	JJ NN	2.6	37.8
6	138	DT NNP NNP	1.8	39.5
7	138	NNS	1.8	41.3
8	112	JJ	1.5	42.8
9	102	VBD	1.3	44.1
10	92	DT NNP NNP NNP	1.2	45.3
11	85	JJ NNS	1.1	46.4
12	79	NNP NN	1.0	47.4
13	76	NN NN	1.0	48.4
14	61	VBN	0.8	49.2
15	60	NNP NNP NNP NNP	0.8	50.0
BLOG$_p$	+3,869	*more with Count* ≤ 49	50.0%	

Table 3: Top 50% of marked POS tag sequences.

	Count	*Non-Terminal*	*Frac*	*Sum*
1	5,759	NP	74.5%	
2	997	VP	12.9	87.4
3	524	S	6.8	94.2
4	120	PP	1.6	95.7
5	72	ADJP	0.9	96.7
6	61	FRAG	0.8	97.4
7	41	ADVP	0.5	98.0
8	39	SBAR	0.5	98.5
9	19	PRN	0.2	98.7
10	18	NX	0.2	99.0
BLOG$_p$	+81	*more with Count* ≤ 16	1.0%	

Table 4: Top 99% of dominating non-terminals.

5.1 Surface Text Statistics

Out of 57,809 sentences, 6,047 (10.5%) are annotated (see Table 2); and 4,934 (8.5%) have multi-token bracketings. We do not distinguish HTML tags and track only unique bracketing end-points within a sentence. Of these, 6,015 are multi-token — an average per-sentence yield of 10.4%.[10]

[10] A non-trivial fraction of our corpus is older (pre-internet) unannotated articles, so this estimate may be conservative.

As expected, many of the annotated words are nouns, but there are adjectives, verbs and other parts of speech too (see Table 3). Mark-up is short, typically under five words, yet (by far) the most frequently marked sequence of POS tags is a pair.

5.2 Common Syntactic Subtrees

For three-quarters of all mark-up, the lowest dominating non-terminal is a noun phrase (see Table 4); there are also non-trace quantities of verb phrases (12.9%) and other phrases, clauses and fragments.

Of the top fifteen — *35.2%* of all — annotated productions, only one is *not* a noun phrase (see Table 5, left). Four of the fifteen lowest dominating non-terminals do *not* match the entire bracketing — all four miss the leading determiner, as we saw earlier. In such cases, we recursively split internal nodes until the bracketing aligned, as follows:

[S [NP the <a>*Toronto Star*][VP reports [NP this]
[PP in the softest possible way],[S stating ...]]]

S → NP VP → DT NNP NNP VBZ NP PP S

We can summarize productions more compactly by using a dependency framework and clipping off any dependents whose subtrees do not cross a bracketing boundary, relative to the parent. Thus,

DT NNP NNP VBZ DT IN DT JJS JJ NN

becomes DT NNP VBZ, "the <a>*Star* reports." Viewed this way, the top fifteen (now collapsed) productions cover *59.4%* of all cases and include four verb heads, in addition to a preposition and an adjective (see Table 5, right). This exposes five cases of inexact matches, three of which involve neglected determiners or adjectives to the left of the head. In fact, the only case that cannot be explained by dropped dependents is #8, where the daughters are marked but the parent is left out. Most instances contributing to this pattern are flat NPs that end with a noun, incorrectly assumed to be the head of *all* other words in the phrase, e.g.,

... [NP a 1994 *New Yorker* article] ...

As this example shows, disagreements (as well as agreements) between mark-up and machine-generated parse trees with automatically percolated heads should be taken with a grain of salt.[11]

[11] In a relatively recent study, Ravi et al. (2008) report that Charniak's re-ranking parser (Charniak and Johnson, 2005) — `reranking-parserAug06.tar.gz`, also available from `ftp://ftp.cs.brown.edu/pub/nlparser/` — attains 86.3% accuracy when trained on WSJ and tested against Brown; its nearly 5% performance loss out-of-domain is consistent with the numbers originally reported by Gildea (2001).

	Count	Constituent Production	Frac	Sum
1	746	NP → NNP NNP	9.6%	
2	357	NP → NNP	4.6	14.3
3	266	NP → NP PP	3.4	17.7
4	183	NP → NNP NNP NNP	2.4	20.1
5	165	NP → DT NNP NNP	2.1	22.2
6	140	NP → NN	1.8	24.0
7	131	NP → DT NNP NNP NNP	1.7	25.7
8	130	NP → DT NN	1.7	27.4
9	127	NP → DT NNP NNP	1.6	29.0
10	109	S → NP VP	1.4	30.4
11	91	NP → DT NNP NNP NNP	1.2	31.6
12	82	NP → DT JJ NN	1.1	32.7
13	79	NP → NNS	1.0	33.7
14	65	NP → JJ NN	0.8	34.5
15	60	NP → NP NP	0.8	35.3
BLOG$_p$	+5,000	*more with Count* $\leq$ 60	64.7%	

	Count	Head-Outward Spawn	Frac	Sum
1	1,889	NNP	24.4%	
2	623	NN	8.1	32.5
3	470	DT NNP	6.1	38.6
4	458	DT NN	5.9	44.5
5	345	NNS	4.5	49.0
6	109	NNPS	1.4	50.4
7	98	VBG	1.3	51.6
8	96	NNP NNP NN	1.2	52.9
9	80	VBD	1.0	53.9
10	77	IN	1.0	54.9
11	74	VBN	1.0	55.9
12	73	DT JJ NN	0.9	56.8
13	71	VBZ	0.9	57.7
14	69	POS NNP	0.9	58.6
15	63	JJ	0.8	59.4
BLOG$_p$	+3,136	*more with Count* $\leq$ 62	40.6%	

Table 5: Top 15 marked productions, viewed as constituents (left) and as dependencies (right), after recursively expanding any internal nodes that did not align with the bracketing (underlined). Tabulated dependencies were collapsed, dropping any dependents that fell entirely in the same region as their parent (i.e., both inside the bracketing, both to its left or both to its right), keeping only crossing attachments.

5.3 Proposed Parsing Constraints

The straight-forward approach — forcing mark-up to correspond to constituents — agrees with Charniak's parse trees only **48.0**% of the time, e.g.,

> ... in [NP <a> [NP an analysis] [PP of perhaps the most astonishing PC item I have yet stumbled upon]].

This number should be higher, as the vast majority of disagreements are due to tree-bank idiosyncrasies (e.g., bare NPs). Earlier examples of incomplete constituents (e.g., legitimately missing determiners) would also be fine in many linguistic theories (e.g., as N-bars). A dependency formulation is less sensitive to such stylistic differences.

We begin with the hardest possible constraint on dependencies, then slowly relax it. Every example used to demonstrate a softer constraint doubles as a counter-example against all previous versions.

• *strict* — seals mark-up into attachments, i.e., inside a bracketing, enforces exactly one external arc — into the overall head. This agrees with head-percolated trees just **35.6**% of the time, e.g.,

> As author of *The Satanic Verses*, I ...

• *loose* — same as *strict*, but allows the bracketing's head word to have external dependents. This relaxation already agrees with head-percolated dependencies **87.5**% of the time, catching many (though far from all) dropped dependents, e.g.,

> ... the *Toronto Star* reports ...

• *sprawl* — same as *loose*, but now allows *all* words inside a bracketing to attach external dependents.[12] This boosts agreement with head-percolated trees to **95.1**%, handling new cases, e.g., where "*Toronto Star*" is embedded in longer mark-up that includes its own parent — a verb:

> ... the <a>*Toronto Star* reports ... ...

• *tear* — allows mark-up to fracture after all, requiring only that the external heads attaching the pieces lie to the same side of the bracketing. This propels agreement with percolated dependencies to **98.9**%, fixing previously broken PP-attachment ambiguities, e.g., a fused phrase like "Fox News in Canada" that detached a preposition from its verb:

> ... concession ... has raised eyebrows among those waiting [PP for <a>Fox News] [PP in Canada] .

Most of the remaining 1.1% of disagreements are due to parser errors. Nevertheless, it *is* possible for mark-up to be torn apart by external heads from *both* sides. We leave this section with a (very rare) true negative example. Below, "CSA" modifies "authority" (to its left), appositively, while "Al-Manar" modifies "television" (to its right):[13]

> The French broadcasting authority, <a>CSA, banned ... Al-Manar satellite television from ...

[12] This view evokes the trapezoids of the $O(n^3)$ recognizer for split head automaton grammars (Eisner and Satta, 1999).

[13] But this is a stretch, since the comma after "CSA" renders the marked phrase ungrammatical even *out* of context.

6 Experimental Methods and Metrics

We implemented the DMV (Klein and Manning, 2004), consulting the details of (Spitkovsky et al., 2010a). Crucially, we swapped out inside-outside re-estimation in favor of Viterbi training. Not only is it better-suited to the general problem (see §7.1), but it also admits a trivial implementation of (most of) the dependency constraints we proposed.[14]

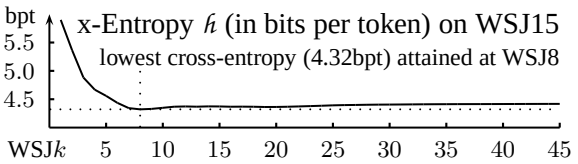

Figure 1: Sentence-level cross-entropy on WSJ15 for Ad-Hoc* initializers of WSJ$\{1, \ldots, 45\}$.

Six settings parameterized each run:

• INIT: 0 — default, uniform initialization; or 1 — a high quality initializer, pre-trained using Ad-Hoc* (Spitkovsky et al., 2010a): we chose the Laplace-smoothed model trained at WSJ15 (the "sweet spot" data gradation) but initialized off WSJ8, since that ad-hoc harmonic initializer has the best cross-entropy on WSJ15 (see Figure 1).

• GENRE: 0 — default, baseline training on WSJ; else, uses 1 — BLOG$_t$; 2 — NEWS; or 3 — WEB.

• SCOPE: 0 — default, uses all sentences up to length 45; if 1, trains using sentences up to length 15; if 2, re-trains on sentences up to length 45, starting from the solution to sentences up to length 15, as recommended by Spitkovsky et al. (2010a).

• CONSTR: if 4, *strict*; if 3, *loose*; and if 2, *sprawl*. We did not implement level 1, *tear*. Over-constrained sentences are re-attempted at successively lower levels until they become possible to parse, if necessary at the lowest (default) level 0.[15]

• TRIM: if 1, discards any sentence without a single multi-token mark-up (shorter than its length).

• ADAPT: if 1, upon convergence, initializes re-training on WSJ45 using the solution to <GENRE>, attempting domain adaptation (Lee et al., 1991).

These make for 294 meaningful combinations. We judged each one by its accuracy on WSJ45, using standard directed scoring — the fraction of correct dependencies over randomized "best" parse trees.

[14] We analyze the benefits of Viterbi training in a companion paper (Spitkovsky et al., 2010b), which dedicates more space to implementation and to the WSJ baselines used here.

[15] At level 4, X<u> Y Z</u> is over-constrained.

7 Discussion of Experimental Results

Evaluation on Section 23 of WSJ and Brown reveals that blog-training beats all published state-of-the-art numbers in every traditionally-reported length cutoff category, with news-training not far behind. Here is a mini-preview of these results, for Section 23 of WSJ10 and WSJ$^\infty$ (from Table 8):

	WSJ10	WSJ$^\infty$
(Cohen and Smith, 2009)	62.0	42.2
(Spitkovsky et al., 2010a)	57.1	45.0
NEWS-best	67.3	*50.1*
BLOG$_t$-best	**69.3**	**50.4**
(Headden et al., 2009)	*68.8*	

Table 6: Directed accuracies on Section 23 of WSJ$\{10, \infty\}$ for three recent state-of-the-art systems and our best runs (as judged against WSJ45) for NEWS and BLOG$_t$ (more details in Table 8).

Since our experimental setup involved testing nearly three hundred models simultaneously, we must take extreme care in analyzing and interpreting these results, to avoid falling prey to any looming "data-snooping" biases.[16] In a sufficiently large pool of models, where each is trained using a randomized and/or chaotic procedure (such as ours), the best may look good due to pure chance. We appealed to three separate diagnostics to convince ourselves that our best results are *not* noise.

The most radical approach would be to write off WSJ as a development set and to focus only on the results from the held-out Brown corpus. It was initially intended as a test of out-of-domain generalization, but since Brown was in no way involved in selecting the best models, it also qualifies as a blind evaluation set. We observe that our best models perform even better (and gain more — see Table 8) on Brown than on WSJ — a strong indication that our selection process has not overfitted.

Our second diagnostic is a closer look at WSJ. Since we cannot graph the full (six-dimensional) set of results, we begin with a simple linear regression, using accuracy on WSJ45 as the dependent variable. We prefer this full factorial design to the more traditional ablation studies because it allows us to account for and to incorporate every single experimental data point incurred along the

[16] In the standard statistical hypothesis testing setting, it is reasonable to expect that $p\%$ of randomly chosen hypotheses will appear significant at the $p\%$ level simply by chance. Consequently, *multiple* hypothesis testing requires re-evaluating significance levels — adjusting raw p-values, e.g., using the Holm-Bonferroni method (Holm, 1979).

Corpus	*Marked Sentences*	*All Sentences*	*POS Tokens*	*All Bracketings*	*Multi-Token Bracketings*
BLOG$_t$45	5,641	56,191	1,048,404	7,021	5,346
BLOG$'_t$45	4,516	4,516	104,267	5,771	5,346
BLOG$_t$15	1,562	23,214	212,872	1,714	1,240
BLOG$'_t$15	1,171	1,171	11,954	1,288	1,240
NEWS45	304,129,910	2,263,563,078	32,119,123,561	611,644,606	477,362,150
NEWS$'$45	205,671,761	205,671,761	2,740,258,972	453,781,081	392,600,070
NEWS15	211,659,549	1,433,779,438	11,786,164,503	365,145,549	274,791,675
NEWS$'$15	147,848,358	147,848,358	1,397,562,474	272,223,918	231,029,921
WEB45	1,577,208,680	8,903,458,234	87,269,385,640	3,309,897,461	2,459,337,571
WEB$'$45	933,115,032	933,115,032	11,552,983,379	2,084,359,555	1,793,238,913
WEB15	1,181,696,194	7,488,669,239	55,014,582,024	2,071,743,595	1,494,675,520
WEB$'$15	681,087,020	681,087,020	5,813,555,341	1,200,980,738	1,072,910,682

Table 7: Counts of sentences, tokens and (unique) bracketings for web-based data sets; trimmed versions, restricted to only those sentences having at least one multi-token bracketing, are indicated by a prime ($'$).

way. Its output is a coarse, high-level summary of our runs, showing which factors significantly contribute to changes in error rate on WSJ45:

Parameter	*(Indicator)*	*Setting*	$\hat{\beta}$	*p-value*
INIT	1	ad-hoc @WSJ8,15	11.8	***
GENRE	1	BLOG$_t$	-3.7	0.06
	2	NEWS	-5.3	**
	3	WEB	-7.7	***
SCOPE	1	@15	-0.5	0.40
	2	@15→45	-0.4	0.53
CONSTR	2	*sprawl*	0.9	0.23
	3	*loose*	1.0	0.15
	4	*strict*	1.8	*
TRIM	1	drop unmarked	-7.4	***
ADAPT	1	WSJ re-training	1.5	**
Intercept		($R^2_{\text{Adjusted}} = 73.6\%$)	39.9	***

We use a standard convention: *** for $p < 0.001$;
** for $p < 0.01$ (very signif.); and * for $p < 0.05$ (signif.).

The default training mode (all parameters zero) is estimated to score 39.9%. A good initializer gives the biggest (double-digit) gain; both domain adaptation and constraints also make a positive impact. Throwing away unannotated data hurts, as does training out-of-domain (the blog is least bad; the web is worst). Of course, this overview should not be taken too seriously. Overly simplistic, a first order model ignores interactions between parameters. Furthermore, a least squares fit aims to capture central tendencies, whereas we are more interested in outliers — the best-performing runs.

A major imperfection of the simple regression model is that helpful factors that require an interaction to "kick in" may not, on their own, appear statistically significant. Our third diagnostic is to examine parameter settings that give rise to the best-performing models, looking out for combinations that consistently deliver superior results.

7.1 WSJ Baselines

Just two parameters apply to learning from WSJ. Five of their six combinations are state-of-the-art, demonstrating the power of Viterbi training; only the default run scores worse than 45.0%, attained by Leapfrog (Spitkovsky et al., 2010a), on WSJ45:

Settings	SCOPE=0	SCOPE=1	SCOPE=2
INIT=0	*41.3*	45.0	45.2
1	46.6	47.5	**47.6**
	@45	@15	@15→45

7.2 Blog

Simply training on BLOG$_t$ instead of WSJ hurts:

GENRE=1	SCOPE=0	SCOPE=1	SCOPE=2
INIT=0	39.6	36.9	36.9
1	46.5	46.3	46.4
	@45	@15	@15→45

The best runs use a good initializer, discard unannotated sentences, enforce the *loose* constraint on the rest, follow up with domain adaptation and benefit from re-training — GENRE=TRIM=ADAPT=1:

INIT=1	SCOPE=0	SCOPE=1	SCOPE=2
CONSTR=0	45.8	48.3	49.6
(*sprawl*) 2	46.3	49.2	49.2
(*loose*) 3	41.3	*50.2*	**50.4**
(*strict*) 4	40.7	49.9	48.7
	@45	@15	@15→45

The contrast between unconstrained learning and annotation-guided parsing is higher for the default initializer, still using trimmed data sets (just over a thousand sentences for BLOG$'_t$15 — see Table 7):

INIT=0	SCOPE=0	SCOPE=1	SCOPE=2
CONSTR=0	25.6	19.4	19.3
(*sprawl*) 2	25.2	22.7	22.5
(*loose*) 3	32.4	26.3	27.3
(*strict*) 4	36.2	38.7	40.1
	@45	@15	@15→45

Above, we see a clearer benefit to our constraints.

7.3 News

Training on WSJ is also better than using NEWS:

GENRE=2	SCOPE=0	SCOPE=1	SCOPE=2
INIT=0	40.2	38.8	38.7
1	43.4	44.0	43.8
	@45	@15	@15→45

As with the blog, the best runs use the good initializer, discard unannotated sentences, enforce the *loose* constraint and follow up with domain adaptation — GENRE=2; INIT=TRIM=ADAPT=1:

Settings	SCOPE=0	SCOPE=1	SCOPE=2
CONSTR=0	46.6	45.4	45.2
(*sprawl*) 2	46.1	44.9	44.9
(*loose*) 3	**49.5**	48.1	48.3
(*strict*) 4	37.7	36.8	37.6
	@45	@15	@15→45

With all the extra training data, the best new score is just 49.5%. On the one hand, we are disappointed by the lack of dividends to orders of magnitude more data. On the other, we are comforted that the system arrives within 1% of its best result — 50.4%, obtained with a manually cleaned up corpus — now using an auto-generated data set.

7.4 Web

The WEB-side story is more discouraging:

GENRE=3	SCOPE=0	SCOPE=1	SCOPE=2
INIT=0	38.3	35.1	35.2
1	42.8	43.6	43.4
	@45	@15	@15→45

Our best run again uses a good initializer, keeps *all* sentences, still enforces the *loose* constraint and follows up with domain adaptation, but performs worse than all well-initialized WSJ baselines, scoring only 45.9% (trained at WEB15).

We suspect that the web is just too messy for us. On top of the challenges of language identification and sentence-breaking, there is a lot of boiler-plate; furthermore, web text can be difficult for news-trained POS taggers. For example, note that the verb "sign" is twice mistagged as a noun and that "YouTube" is classified as a verb, in the top four POS sequences of web sentences:[17]

	POS Sequence	WEB *Count*
	Sample web sentence, chosen uniformly at random.	
1	DT NNS VBN	82,858,487
	All rights reserved.	
2	NNP NNP NNP	65,889,181
	Yuasa et al.	
3	NN IN TO VB RB	31,007,783
	Sign in to YouTube now!	
4	NN IN IN PRP$ JJ NN	31,007,471
	Sign in with your Google Account!	

[17] Further evidence: TnT tags the ubiquitous but ambiguous fragments "click here" and "print post" as noun phrases.

7.5 The State of the Art

Our best model gains more than 5% over previous state-of-the-art accuracy across all sentences of WSJ's Section 23, more than 8% on WSJ20 and rivals the oracle skyline (Spitkovsky et al., 2010a) on WSJ10; these gains generalize to Brown100, where it improves by nearly 10% (see Table 8).

We take solace in the fact that our best models agree in using *loose* constraints. Of these, the models trained with less data perform better, with the best two using trimmed data sets, echoing that "less is more" (Spitkovsky et al., 2010a), pace Halevy et al. (2009). We note that orders of magnitude more data did not improve parsing performance further and suspect a different outcome from lexicalized models: The primary benefit of additional lower-quality data is in improved coverage. But with only 35 unique POS tags, data sparsity is hardly an issue. Extra examples of lexical items help little and hurt when they are mistagged.

8 Related Work

The wealth of new annotations produced in many languages every day already fuels a number of NLP applications. Following their early and wide-spread use by search engines, in service of spam-fighting and retrieval, anchor text and link data enhanced a variety of traditional NLP techniques: cross-lingual information retrieval (Nie and Chen, 2002), translation (Lu et al., 2004), both named-entity recognition (Mihalcea and Csomai, 2007) and categorization (Watanabe et al., 2007), query segmentation (Tan and Peng, 2008), plus semantic relatedness and word-sense disambiguation (Gabrilovich and Markovitch, 2007; Yeh et al., 2009). Yet several, seemingly natural, candidate core NLP tasks — tokenization, CJK segmentation, noun-phrase chunking, and (until now) parsing — remained conspicuously uninvolved.

Approaches related to ours arise in applications that combine parsing with named-entity recognition (NER). For example, constraining a parser to respect the boundaries of known entities is standard practice not only in joint modeling of (constituent) parsing and NER (Finkel and Manning, 2009), but also in higher-level NLP tasks, such as relation extraction (Mintz et al., 2009), that couple chunking with (dependency) parsing. Although restricted to proper noun phrases, dates, times and quantities, we suspect that constituents identified by trained (supervised) NER systems would also

Model	*Incarnation*		WSJ10	WSJ20	WSJ$^\infty$	
DMV	Bilingual Log-Normals (tie-verb-noun)	(Cohen and Smith, 2009)	62.0	48.0	42.2	Brown100
	Leapfrog	(Spitkovsky et al., 2010a)	57.1	48.7	45.0	43.6
	default	INIT=0,GENRE=0,SCOPE=0,CONSTR=0,TRIM=0,ADAPT=0	55.9	45.8	41.6	40.5
	WSJ-best	INIT=1,GENRE=0,SCOPE=2,CONSTR=0,TRIM=0,ADAPT=0	65.3	53.8	47.9	50.8
	BLOG$_t$-best	INIT=1,GENRE=1,SCOPE=2,CONSTR=3,TRIM=1,ADAPT=1	**69.3**	**56.8**	**50.4**	**53.3**
	NEWS-best	INIT=1,GENRE=2,SCOPE=0,CONSTR=3,TRIM=1,ADAPT=1	67.3	*56.2*	*50.1*	*51.6*
	WEB-best	INIT=1,GENRE=3,SCOPE=1,CONSTR=3,TRIM=0,ADAPT=1	64.1	52.7	46.3	46.9
EVG	Smoothed (skip-head), Lexicalized	(Headden et al., 2009)	*68.8*			

Table 8: Accuracies on Section 23 of WSJ$\{10, 20, \infty\}$ and Brown100 for three recent state-of-the-art systems, our default run, and our best runs (judged by accuracy on WSJ45) for each of four training sets.

be helpful in constraining grammar induction.

Following Pereira and Schabes' (1992) success with partial annotations in training a model of (English) constituents generatively, their idea has been extended to discriminative estimation (Riezler et al., 2002) and also proved useful in modeling (Japanese) dependencies (Sassano, 2005). There was demand for partially bracketed corpora. Chen and Lee (1995) constructed one such corpus by learning to partition (English) POS sequences into chunks (Abney, 1991); Inui and Kotani (2001) used n-gram statistics to split (Japanese) clauses.

We combine the two intuitions, using the web to build a partially parsed corpus. Our approach could be called *lightly-supervised*, since it does not require manual annotation of a single complete parse tree. In contrast, traditional semi-supervised methods rely on fully-annotated seed corpora.[18]

9 Conclusion

We explored novel ways of training dependency parsing models, the best of which attains 50.4% accuracy on Section 23 (all sentences) of WSJ, beating all previous unsupervised state-of-the-art by more than 5%. Extra gains stem from guiding Viterbi training with web mark-up, the *loose* constraint consistently delivering best results. Our linguistic analysis of a blog reveals that web annotations can be converted into accurate parsing constraints (*loose*: 88%; *sprawl*: 95%; *tear*: 99%) that could be helpful to supervised methods, e.g., by boosting an initial parser via self-training (McClosky et al., 2006) on sentences with mark-up. Similar techniques may apply to standard word-processing annotations, such as font changes, and to certain (balanced) punctuation (Briscoe, 1994).

We make our blog data set, overlaying mark-up and syntax, publicly available. Its annotations are 75% noun phrases, 13% verb phrases, 7% simple declarative clauses and 2% prepositional phrases, with traces of other phrases, clauses and fragments. The type of mark-up, combined with POS tags, could make for valuable features in discriminative models of parsing (Ratnaparkhi, 1999).

A logical next step would be to explore the connection between syntax and mark-up for genres other than a news-style blog and for languages other than English. We are excited by the possibilities, as unsupervised parsers are on the cusp of becoming useful in their own right — recently, Davidov et al. (2009) successfully applied Seginer's (2007) fully unsupervised grammar inducer to the problems of pattern-acquisition and extraction of semantic data. If the strength of the connection between web mark-up and syntactic structure is universal across languages and genres, this fact could have broad implications for NLP, with applications extending well beyond parsing.

[18] A significant effort expended in building a tree-bank comes with the first batch of sentences (Druck et al., 2009).

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, Spence Green, Christopher D. Manning, Richard Socher, Mihai Surdeanu and the anonymous reviewers for many helpful suggestions, and we are especially grateful to Andy Golding, for pointing us to his sample Map-Reduce over the Google News crawl, and to Daniel Pipes, for allowing us to distribute the data set derived from his blog entries.

References

S. Abney. 1991. Parsing by chunks. *Principle-Based Parsing: Computation and Psycholinguistics.*

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

C. Barr, R. Jones, and M. Regelson. 2008. The linguistic structure of English web-search queries. In *EMNLP.*

T. Brants. 2000. TnT — a statistical part-of-speech tagger. In *ANLP.*

T. Briscoe. 1994. Parsing (with) punctuation, etc. Technical report, Xerox European Research Laboratory.

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

E. Charniak. 2001. Immediate-head parsing for language models. In *ACL*.

H.-H. Chen and Y.-S. Lee. 1995. Development of a partially bracketed corpus with part-of-speech information only. In *WVLC*.

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

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

D. Davidov, R. Reichart, and A. Rappoport. 2009. Superior and efficient fully unsupervised pattern-based concept acquisition using an unsupervised parser. In *CoNLL*.

G. Druck, G. Mann, and A. McCallum. 2009. Semi-supervised learning of dependency parsers using generalized expectation criteria. In *ACL-IJCNLP*.

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

J. R. Finkel and C. D. Manning. 2009. Joint parsing and named entity recognition. In *NAACL-HLT*.

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.

E. Gabrilovich and S. Markovitch. 2007. Computing semantic relatedness using Wikipedia-based Explicit Semantic Analysis. In *IJCAI*.

D. Gildea. 2001. Corpus variation and parser performance. In *EMNLP*.

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

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

S. Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6.

N. Inui and Y. Kotani. 2001. Robust N-gram based syntactic analysis using segmentation words. In *PACLIC*.

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

C.-H. Lee, C.-H. Lin, and B.-H. Juang. 1991. A study on speaker adaptation of the parameters of continuous density Hidden Markov Models. *IEEE Trans. on Signal Processing*, 39.

W.-H. Lu, L.-F. Chien, and H.-J. Lee. 2004. Anchor text mining for translation of Web queries: A transitive translation approach. *ACM Trans. on Information Systems*, 22.

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

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

R. Mihalcea and A. Csomai. 2007. Wikify!: Linking documents to encyclopedic knowledge. In *CIKM*.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL-IJCNLP*.

J.-Y. Nie and J. Chen. 2002. Exploiting the Web as parallel corpora for cross-language information retrieval. *Web Intelligence*.

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

A. Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34.

S. Ravi, K. Knight, and R. Soricut. 2008. Automatic prediction of parser accuracy. In *EMNLP*.

J. C. Reynar and A. Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *ANLP*.

S. Riezler, T. H. King, R. M. Kaplan, R. Crouch, J. T. Maxwell, III, and M. Johnson. 2002. Parsing the Wall Street Journal using a lexical-functional grammar and discriminative estimation techniques. In *ACL*.

M. Sassano. 2005. Using a partially annotated corpus to build a dependency parser for Japanese. In *IJCNLP*.

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

V. I. Spitkovsky, H. Alshawi, and D. Jurafsky. 2010a. From Baby Steps to Leapfrog: How "Less is More" in unsupervised dependency parsing. In *NAACL-HLT*.

V. I. Spitkovsky, H. Alshawi, D. Jurafsky, and C. D. Manning. 2010b. Viterbi training improves unsupervised dependency parsing. In *CoNLL*.

B. Tan and F. Peng. 2008. Unsupervised query segmentation using generative language models and Wikipedia. In *WWW*.

K. Toutanova and C. D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *EMNLP-VLC*.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL*.

D. Vadas and J. R. Curran. 2007. Adding noun phrase structure to the Penn Treebank. In *ACL*.

Y. Watanabe, M. Asahara, and Y. Matsumoto. 2007. A graph-based approach to named entity categorization in Wikipedia using conditional random fields. In *EMNLP-CoNLL*.

E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa. 2009. WikiWalk: Random walks on Wikipedia for semantic relatedness. In *TextGraphs*.