Unsupervised Dependency Parsing without Gold Part-of-Speech Tags

Valentin I. Spitkovsky[†*]
valentin@cs.stanford.edu

Hiyan Alshawi[*]
hiyan@google.com

Angel X. Chang[†*]
angelx@cs.stanford.edu

Daniel Jurafsky[‡†]
jurafsky@stanford.edu

[†]Computer Science Department
Stanford University
Stanford, CA, 94305

[*]Google Research
Google Inc.
Mountain View, CA, 94043

[‡]Department of Linguistics
Stanford University
Stanford, CA, 94305

Abstract

We show that categories induced by unsupervised word clustering can surpass the performance of gold part-of-speech tags in dependency grammar induction. Unlike classic clustering algorithms, our method allows a word to have different tags in different contexts. In an ablative analysis, we first demonstrate that this context-dependence is crucial to the superior performance of gold tags — requiring a word to always have the same part-of-speech significantly degrades the performance of manual tags in grammar induction, eliminating the advantage that human annotation has over unsupervised tags. We then introduce a sequence modeling technique that combines the output of a word clustering algorithm with context-colored noise, to allow words to be tagged differently in different contexts. With these new induced tags as input, our state-of-the-art dependency grammar inducer achieves 59.1% directed accuracy on Section 23 (all sentences) of the Wall Street Journal (WSJ) corpus — 0.7% higher than using gold tags.

1 Introduction

Unsupervised learning — machine learning without manually-labeled training examples — is an active area of scientific research. In natural language processing, unsupervised techniques have been successfully applied to tasks such as word alignment for machine translation. And since the advent of the web, algorithms that induce structure from unlabeled data have continued to steadily gain importance. In this paper we focus on unsupervised part-of-speech tagging and dependency parsing — two related problems of syntax discovery. Our methods are applicable to vast quantities of unlabeled monolingual text.

Not all research on these problems has been fully unsupervised. For example, to the best of our knowledge, every new state-of-the-art dependency grammar inducer since Klein and Manning (2004) relied on gold part-of-speech tags. For some time, multi-point performance degradations caused by switching to automatically induced word categories have been interpreted as indications that "good enough" parts-of-speech induction methods exist, justifying the focus on grammar induction with supervised part-of-speech tags (Bod, 2006), pace (Cramer, 2007). One of several drawbacks of this practice is that it weakens any conclusions that could be drawn about how computers (and possibly humans) learn in the absence of explicit feedback (McDonald et al., 2011).

In turn, not all unsupervised taggers actually induce word categories: Many systems — known as part-of-speech *disambiguators* (Merialdo, 1994) — rely on external dictionaries of possible tags. Our work builds on two older part-of-speech *inducers* — word clustering algorithms of Clark (2000) and Brown et al. (1992) — that were recently shown to be more robust than other well-known fully unsupervised techniques (Christodoulopoulos et al., 2010).

We investigate which properties of gold part-of-speech tags are useful in grammar induction and parsing, and how these properties could be introduced into induced tags. We also explore the number of word classes that is good for grammar induction: in particular, whether categorization is needed at all. By removing the "unrealistic simplification" of using gold tags (Petrov et al., 2011, §3.2, Footnote 4), we will go on to demonstrate why grammar induction from plain text is no longer "still too difficult."

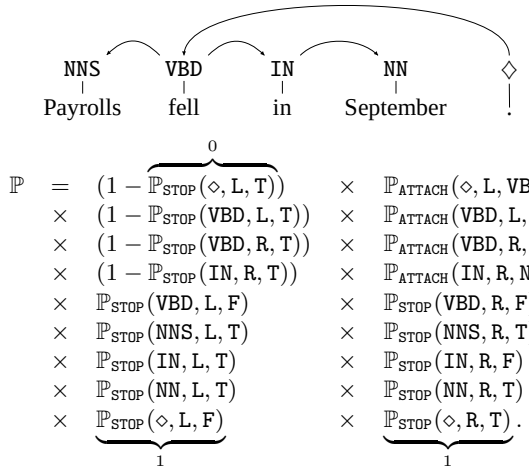

Figure 1: A dependency structure for a short WSJ sentence and its probability, factored by the DMV, using gold tags, after summing out $\mathbb{P}_{\text{ORDER}}$ (Spitkovsky et al., 2009).

2 Methodology

In all experiments, we model the English grammar via Klein and Manning's (2004) Dependency Model with Valence (DMV), induced from subsets of not-too-long sentences of the Wall Street Journal (WSJ).

2.1 The Model

The original DMV is a single-state head automata model (Alshawi, 1996) over lexical word classes $\{c_w\}$ — gold part-of-speech 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 recursive process produces only projective trees. A root token $\diamond$ generates the head of the sentence as its left (and only) child (see Figure 1 for a simple, concrete example).

2.2 Learning Algorithms

The DMV lends itself to unsupervised learning via inside-outside re-estimation (Baker, 1979). Klein and Manning (2004) initialized their system using an "ad-hoc harmonic" completion, followed by training using 40 steps of EM (Klein, 2005). We reproduce this set-up, iterating without actually verifying convergence, in most of our experiments (#1–4, §3–4).

Experiments #5–6 (§5) employ our new state-of-the-art grammar inducer (Spitkovsky et al., 2011), which uses constrained Viterbi EM (details in §5).

2.3 Training Data

The DMV is usually trained on a customized subset of Penn English Treebank's Wall Street Journal portion (Marcus et al., 1993). Following Klein and Manning (2004), we begin with reference constituent parses, prune out all empty sub-trees and remove punctuation and terminals (tagged # and $) that are not pronounced where they appear. We then train only on the remaining sentence *yields* consisting of no more than fifteen tokens (WSJ15), in most of our experiments (#1–4, §3–4); by contrast, Klein and Manning's (2004) original system was trained using less data: sentences up to length ten (WSJ10).[1]

Our final experiments (#5–6, §5) employ a simple scaffolding strategy (Spitkovsky et al., 2010a) that follows up initial training at WSJ15 ("less is more") with an additional training run ("leapfrog") that incorporates most sentences of the data set, at WSJ45.

2.4 Evaluation Methods

Evaluation is against the training set, as is standard practice in unsupervised learning, in part because Klein and Manning (2004, §3) did not smooth the DMV (Klein, 2005, §6.2). For most of our experiments (#1–4, §3–4), this entails starting with the reference trees from WSJ15 (as modified in §2.3), automatically converting their labeled constituents into unlabeled dependencies using deterministic "head-percolation" rules (Collins, 1999), and then computing (directed) dependency accuracy scores of the corresponding induced trees. We report overall percentages of correctly guessed arcs, including the arcs from sentence root symbols, as is standard practice (Paskin, 2001; Klein and Manning, 2004).

For a meaningful comparison with previous work, we also test some of the models from our earlier experiments (#1,3) — and both models from final experiments (#5,6) — against Section 23 of WSJ$^\infty$, after applying Laplace (a.k.a. "add one") smoothing.

[1] WSJ15 contains 15,922 sentences up to length fifteen (a total of 163,715 tokens, not counting punctuation) — versus 7,422 sentences of at most ten words (only 52,248 tokens) comprising WSJ10 — and is a better trade-off between the quantity and complexity of training data in WSJ (Spitkovsky et al., 2009).

1. **manual** tags	Accuracy *Unsupervised*	Accuracy *Sky*	*Viable Groups*
gold	50.7	*78.0*	36
mfc	**47.2**	74.5	34
mfp	40.4	76.4	160
ua	44.3	**78.4**	328
2. tagless **lexicalized** models			
full	25.8	**97.3**	49,180
partial	29.3	60.5	176
none	**30.7**	*24.5*	1
3. tags from a **flat** (Clark, 2000) clustering			
	47.8	**83.8**	197
4. prefixes of a **hierarchical** (Brown et al., 1992) clustering			
first 7 bits	46.4	73.9	96
8 bits	**48.0**	77.8	165
9 bits	46.8	**82.3**	262

Table 1: Directed accuracies for the "less is more" DMV, trained on WSJ15 (after 40 steps of EM) and evaluated also against WSJ15, using various lexical categories in place of gold part-of-speech tags. For each tag-set, we include its effective number of (non-empty) categories in WSJ15 and the oracle skylines (supervised performance).

3 Motivation and Ablative Analyses

The concepts of polysemy and synonymy are of fundamental importance in linguistics. For words that can take on multiple parts of speech, knowing the gold tag can reduce ambiguity, improving parsing by limiting the search space. Furthermore, pooling the statistics of words that play similar syntactic roles, as signaled by shared gold part-of-speech tags, can simplify the learning task, improving generalization by reducing sparsity. We begin with two sets of experiments that explore the impact that each of these factors has on grammar induction with the DMV.

3.1 Experiment #1: Human-Annotated Tags

Our first set of experiments attempts to isolate the effect that replacing gold part-of-speech tags with deterministic *one class per word* mappings has on performance, quantifying the cost of switching to a monosemous clustering (see Table 1: manual; and Table 4). Grammar induction with gold tags scores 50.7%, while the oracle skyline (an ideal, supervised instance of the DMV) could attain 78.0% accuracy.

It may be worth noting that only 6,620 (13.5%) of 49,180 unique tokens in WSJ appear with multiple part-of-speech tags. Most words, like *it*, are always tagged the same way (5,768 times PRP). Some words, like *gains*, usually serve as one part of speech (227 times NNS, as in *the gains*) but are occasionally used differently (5 times VBZ, as in *he gains*). Only 1,322 tokens (2.7%) appear with three or more different gold tags. However, this minority includes the most frequent word — *the* (50,959 times DT, 7 times JJ, 6 times NNP and once as each of CD, NN and VBP).[2]

token	*mfc*	*mfp*	*ua*
it	{PRP}	{PRP}	{PRP}
gains	{NNS}	{VBZ, NNS}	{VBZ, NNS}
the	{DT}	{JJ, DT}	{VBP, NNP, NN, JJ, DT, CD}

Table 2: Example most frequent class, most frequent pair and union all reassignments for tokens *it*, *the* and *gains*.

We experimented with three natural reassignments of part-of-speech categories (see Table 2). The first, *most frequent class* (mfc), simply maps each token to its most common gold tag in the entire WSJ (with ties resolved lexicographically). This approach discards two gold tags (types PDT and RBR are not most common for any of the tokens in WSJ15) and costs about three-and-a-half points of accuracy, in both supervised and unsupervised regimes.

Another reassignment, *union all* (ua), maps each token to the set of all of its observed gold tags, again in the entire WSJ. This inflates the number of groupings by nearly a factor of ten (effectively lexicalizing the most ambiguous words),[3] yet improves the oracle skyline by half-a-point over actual gold tags; however, learning is harder with this tag-set, losing more than six points in unsupervised training.

Our last reassignment, *most frequent pair* (mfp), allows up to two of the most common tags into a token's label set (with ties, once again, resolved lexicographically). This intermediate approach performs strictly worse than *union all*, in both regimes.

3.2 Experiment #2: Lexicalization Baselines

Our next set of experiments assesses the benefits of categorization, turning to lexicalized baselines that avoid grouping words altogether. All three models discussed below estimated the DMV *without* using the gold tags in any way (see Table 1: lexicalized).

[2] Some of these are annotation errors in the treebank (Banko and Moore, 2004, Figure 2): such (mis)taggings can severely degrade the accuracy of part-of-speech disambiguators, without additional supervision (Banko and Moore, 2004, §5, Table 1).

[3] Kupiec (1992) found that the 50,000-word vocabulary of the Brown corpus similarly reduces to ∼400 ambiguity classes.

First, not surprisingly, a fully-lexicalized model over nearly 50,000 unique words is able to essentially memorize the training set, supervised. (Without smoothing, it is possible to deterministically attach most rare words in a dependency tree correctly, etc.) Of course, local search is unlikely to find good instantiations for so many parameters, causing unsupervised accuracy for this model to drop in half.

For our next experiment, we tried an intermediate, partially-lexicalized approach. We mapped frequent words — those seen at least 100 times in the training corpus (Headden et al., 2009) — to their own individual categories, lumping the rest into a single "unknown" cluster, for a total of under 200 groups. This model is significantly worse for supervised learning, compared even with the monosemous clusters derived from gold tags; yet it is only slightly more learnable than the broken fully-lexicalized variant.

Finally, for completeness, we trained a model that maps every token to the same one "unknown" category. As expected, such a trivial "clustering" is ineffective in supervised training; however, it outperforms both lexicalized variants unsupervised,[4] strongly suggesting that lexicalization alone may be insufficient for the DMV and hinting that some degree of categorization is essential to its learnability.

Cluster #173		*Cluster #188*	
1.	open	1.	get
2.	free	2.	make
3.	further	3.	take
4.	higher	4.	find
5.	lower	5.	give
6.	similar	6.	keep
7.	leading	7.	pay
8.	present	8.	buy
9.	growing	9.	win
10.	increased	10.	sell
⋮	⋮	⋮	⋮
37.	cool	42.	improve
⋮	⋮	⋮	⋮
1,688.	up-wind	2,105.	zero-out

Table 3: Representative members for two of the flat word groupings: cluster #173 (left) contains adjectives, especially ones that take comparative (or other) complements; cluster #188 comprises bare-stem verbs (infinitive stems). (Of course, many of the words have other syntactic uses.)

[4]Note that it also beats supervised training. That isn't a bug: Spitkovsky et al. (2010b, §7.2) explain this paradox in the DMV.

4 Grammars over Induced Word Clusters

We have demonstrated the need for grouping similar words, estimated a bound on performance losses due to monosemous clusterings and are now ready to experiment with induced part-of-speech tags. We use two sets of established, publicly-available hard clustering assignments, each computed from a much larger data set than WSJ (approximately a million words). The first is a flat mapping (200 clusters) constructed by training Clark's (2000) distributional similarity model over several hundred million words from the British National and the English Gigaword corpora.[5] The second is a hierarchical clustering — binary strings up to eighteen bits long — constructed by running Brown et al.'s (1992) algorithm over 43 million words from the BLLIP corpus, minus WSJ.[6]

4.1 Experiment #3: A Flat Word Clustering

Our main purely unsupervised results are with a flat clustering (Clark, 2000) that groups words having similar context distributions, according to Kullback-Leibler divergence. (A word's context is an ordered pair: its left- and right-adjacent neighboring words.)

To avoid overfitting, we employed an implementation from previous literature (Finkel and Manning, 2009). The number of clusters (200) and the sufficient amount of training data (several hundred-million words) were tuned to a task (NER) that is not directly related to dependency parsing. (Table 3 shows representative entries for two of the clusters.)

We added one more category (#0) for unknown words. Now every token in WSJ could again be replaced by a coarse identifier (one of at most 201, instead of just 36), in both supervised and unsupervised training. (Our training code did not change.)

The resulting supervised model, though not as good as the fully-lexicalized DMV, was more than five points more accurate than with gold part-of-speech tags (see Table 1: flat). Unsupervised accuracy was lower than with gold tags (see also Table 4) but higher than with *all* three derived hard assignments. This suggests that polysemy (i.e., ability to

[5]`http://nlp.stanford.edu/software/stanford-postagger-2008-09-28.tar.gz:models/egw.bnc.200`

[6]`http://people.csail.mit.edu/maestro/papers/bllip-clusters.gz`

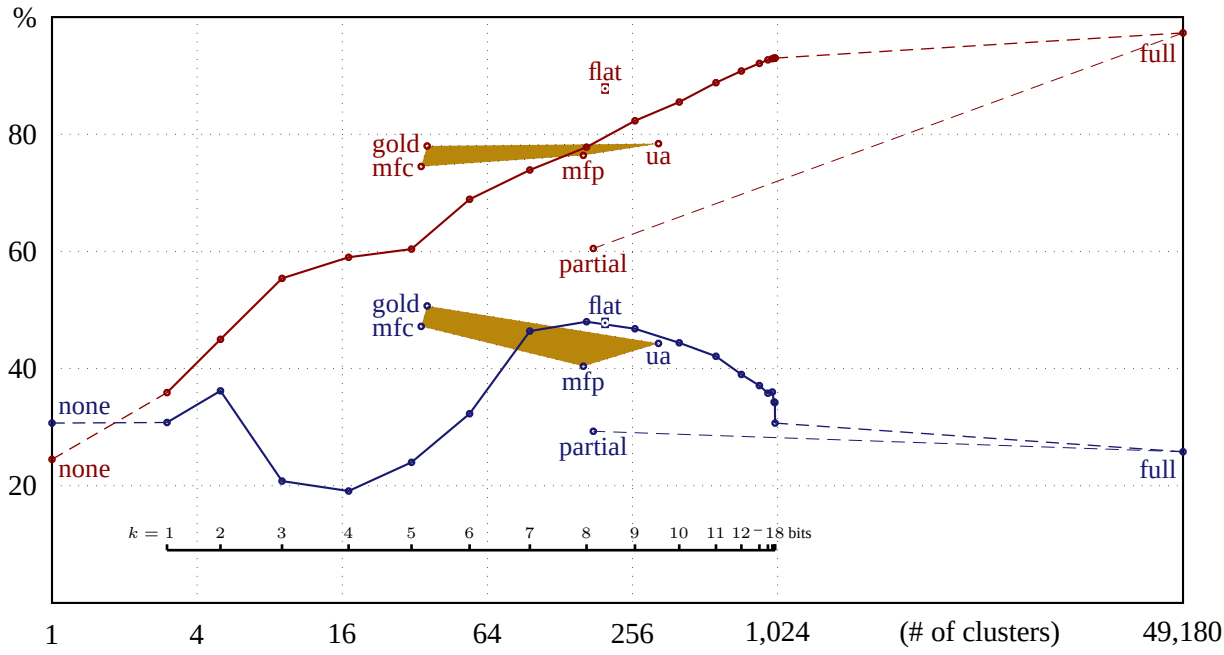

Figure 2: Parsing performance (accuracy on WSJ15) as a "function" of the number of syntactic categories, for all prefix lengths — $k \in \{1, \ldots, 18\}$ — of a hierarchical (Brown et al., 1992) clustering, connected by solid lines (dependency grammar induction in blue; supervised oracle skylines in red, above). Tagless lexicalized models (*full*, *partial* and *none*) connected by dashed lines. Models based on *gold* part-of-speech tags, and derived monosemous clusters (*mfc*, *mfp* and *ua*), shown as vertices of gold polygons. Models based on a *flat* (Clark, 2000) clustering indicated by squares.

tag a word differently in context) may be the primary advantage of manually constructed categorizations.

4.2 Experiment #4: A Hierarchical Clustering

The purpose of this batch of experiments is to show that Clark's (2000) algorithm isn't unique in its suitability for grammar induction. We found that Brown et al.'s (1992) older information-theoretic approach, which does not explicitly address the problems of rare and ambiguous words (Clark, 2000) and was designed to induce large numbers of plausible syntactic *and* semantic clusters, can perform just as well.

Once again, the sufficient amount of data (43 million words) was tuned in earlier work (Koo, 2010). His task of interest was, in fact, dependency parsing. But since this algorithm is hierarchical (i.e., there isn't a parameter for the number of categories), we doubt that there was a strong enough risk of overfitting to question the clustering's unsupervised nature.

As there isn't a set number of categories, we used binary prefixes of length k from each word's address in the computed hierarchy as cluster labels. Results for $7 \leq k \leq 9$ bits (approximately 100–250 non-empty clusters, close to the 200 we used before) are similar to those of flat clusters (see Table 1: hierarchical). Outside of this range, however, performance can be substantially worse (see Figure 2), consistent with earlier findings: Headden et al. (2008) demonstrated that (constituent) grammar induction, using the singular-value decomposition (SVD-based) tagger of Schütze (1995), also works best with 100–200 clusters. Important future research directions may include learning to automatically select a good number of word categories (in the case of flat clusterings) and ways of using multiple clustering assignments, perhaps of different granularities/resolutions, in tandem (e.g., in the case of a hierarchical clustering).

4.3 Further Evaluation

It is important to enable easy comparison with previous and future work. Since WSJ15 is not a standard test set, we evaluated two key experiments — "less is more" with gold part-of-speech tags (#1, Table 1: gold) and with Clark's (2000) clusters (#3, Table 1: flat) — on all sentences (not just length fifteen and shorter), in Section 23 of WSJ (see Table 4). This required smoothing both final models (§2.4).

We showed that two classic unsupervised word

	System Description		*Accuracy*
#1 (§3.1)	"less is more"	(Spitkovsky et al., 2009)	44.0
#3 (§4.1)	"less is more" with monosemous induced tags		41.4 (-2.6)

Table 4: Directed accuracies on Section 23 of WSJ (all sentences) for two experiments with the base system.

clusterings — one flat and one hierarchical — can be better for dependency grammar induction than monosemous syntactic categories derived from gold part-of-speech tags. And we confirmed that the unsupervised tags are worse than the actual gold tags, in a simple dependency grammar induction system.

5 State-of-the-Art without Gold Tags

Until now, we have deliberately kept our experimental methods simple and nearly identical to Klein and Manning's (2004), for clarity. Next, we will explore how our main findings generalize beyond this toy setting. A preliminary test will simply quantify the effect of replacing gold part-of-speech tags with the monosemous flat clustering (as in experiment #3, §4.1) on a modern grammar inducer. And our last experiment will gauge the impact of using a polysemous (but still unsupervised) clustering instead, obtained by executing standard sequence labeling techniques to introduce context-sensitivity into the original (independent) assignment or words to categories.

These final experiments are with our latest state-of-the-art system (Spitkovsky et al., 2011) — a partially lexicalized extension of the DMV that uses constrained Viterbi EM to train on nearly all of the data available in WSJ, at WSJ45 (48,418 sentences; 986,830 non-punctuation tokens). The key contribution that differentiates this model from its predecessors is that it incorporates punctuation into grammar induction (by turning it into parsing constraints, instead of ignoring punctuation marks altogether). In training, the model makes a simplifying assumption — that sentences can be split at punctuation and that the resulting fragments of text could be parsed independently of one another (these parsed fragments are then reassembled into full sentence trees, by parsing the sequence of their own head words). Furthermore, the model continues to take punctuation marks into account in inference (using weaker, more accurate constraints, than in training). This system scores 58.4% on Section 23 of WSJ$^{\infty}$ (see Table 5).

5.1 Experiment #5: A Monosemous Clustering

As in experiment #3 (§4.1), we modified the base system in exactly one way: we swapped out gold part-of-speech tags and replaced them with a flat distributional similarity clustering. In contrast to simpler models, which suffer multi-point drops in accuracy from switching to unsupervised tags (e.g., 2.6%), our new system's performance degrades only slightly, by 0.2% (see Tables 4 and 5). This result improves over substantial performance degradations previously observed for unsupervised dependency parsing with induced word categories (Klein and Manning, 2004; Headden et al., 2008, *inter alia*).[7]

One risk that arises from using gold tags is that newer systems could be finding cleverer ways to exploit manual labels (i.e., developing an over-reliance on gold tags) instead of actually learning to acquire language. Part-of-speech tags are *known* to contain significant amounts of information for unlabeled dependency parsing (McDonald et al., 2011, §3.1), so we find it reassuring that our latest grammar inducer is *less* dependent on gold tags than its predecessors.

5.2 Experiment #6: A Polysemous Clustering

Results of experiments #1 and 3 (§3.1, 4.1) suggest that grammar induction stands to gain from relaxing the *one class per word* assumption. We next test this conjecture by inducing a polysemous unsupervised word clustering, then using it to induce a grammar.

Previous work (Headden et al., 2008, §4) found that simple bitag hidden Markov models, classically trained using the Baum-Welch (Baum, 1972) variant of EM (HMM-EM), perform quite well,[8] on average, across different grammar induction tasks. Such sequence models incorporate a sensitivity to context via state transition probabilities $\mathbb{P}_{\texttt{TRAN}}(t_i \mid t_{i-1})$, capturing the likelihood that a tag t_i immediately follows the tag t_{i-1}; emission probabilities $\mathbb{P}_{\texttt{EMIT}}(w_i \mid t_i)$ capture the likelihood that a word of type t_i is w_i.

[7] We also briefly comment on this result in the "punctuation" paper (Spitkovsky et al., 2011, §7), published concurrently.

[8] They are also competitive with Bayesian estimators, on larger data sets, with cross-validation (Gao and Johnson, 2008).

	System Description		*Accuracy*
(§5)	"punctuation"	(Spitkovsky et al., 2011)	58.4
#5 (§5.1)	"punctuation" with monosemous induced tags		58.2 (-0.2)
#6 (§5.2)	"punctuation" with **context-sensitive** induced tags		**59.1** (+**0.7**)

Table 5: Directed accuracies on Section 23 of WSJ (all sentences) for experiments with the state-of-the-art system.

We need a context-sensitive tagger, and HMM models are good — relative to other tag-inducers. However, they are not better than gold tags, at least when trained using a modest amount of data.[9] For this reason, we decided to relax the monosemous flat clustering, plugging it in as an initializer for the HMM. The main problem with this approach is that, at least without smoothing, every monosemous labeling is trivially at a local optimum, since $\mathbb{P}(t_i \mid w_i)$ is deterministic. To escape the initial assignment, we used a "noise injection" technique (Selman et al., 1994), inspired by the contexts of Clark (2000). First, we collected the MLE statistics for $\mathbb{P}_{\text{R}}(t_{i+1} \mid t_i)$ and $\mathbb{P}_{\text{L}}(t_i \mid t_{i+1})$ in WSJ, using the flat monosemous tags. Next, we replicated the text of WSJ 100-fold. Finally, we retagged this larger data set, as follows: with probability 80%, a word kept its monosemous tag; with probability 10%, we sampled a new tag from the left context ($\mathbb{P}_{\text{L}}$) associated with the original (monosemous) tag of its rightmost neighbor; and with probability 10%, we drew a tag from the right context ($\mathbb{P}_{\text{R}}$) of its leftmost neighbor.[10] Given that our initializer — and later the input to the grammar inducer — are hard assignments of tags to words, we opted for (the faster and simpler) Viterbi training.

In the spirit of reproducibility, we again used an off-the-shelf component for tagging-related work.[11] Viterbi training converged after just 17 steps, replacing the original monosemous tags for 22,280 (of 1,028,348 non-punctuation) tokens in WSJ. For example, the first changed sentence is #3 (of 49,208):

> *Some "circuit breakers" installed after the October 1987 crash failed their first test, traders say, unable to **cool** the selling panic in both stocks and futures.*

Above, the word *cool* gets relabeled as #188 (from #173 — see Table 3), since its context is more suggestive of an infinitive verb than of its usual grouping with adjectives. (A proper analysis of all changes, however, is beyond the scope of this work.)

Using this new context-sensitive hard assignment of tokens to unsupervised categories our grammar inducer attained a directed accuracy of 59.1%, nearly a full point better than with the monosemous hard assignment (see Table 5). To the best of our knowledge it is also the first state-of-the-art unsupervised dependency parser to perform better with induced categories than with gold part-of-speech tags.

6 Related Work

Early work in dependency grammar induction already relied on gold part-of-speech tags (Carroll and Charniak, 1992). Some later models (Yuret, 1998; Paskin, 2001, *inter alia*) attempted full lexicalization. However, Klein and Manning (2004) demonstrated that effort to be worse at recovering dependency arcs than choosing parse structures at random, leading them to incorporate gold tags into the DMV.

Klein and Manning (2004, §5, Figure 6) had also tested their own models with induced word classes, constructed using a distributional similarity clustering method (Schütze, 1995). Without gold part-of-speech tags, their combined DMV+CCM model was about five points worse, both in (directed) unlabeled dependency accuracy (42.3% vs. 47.5%)[12] and unlabeled bracketing F_1 (72.9% vs. 77.6%), on WSJ10.

In constituent parsing, earlier Seginer (2007a, §6, Table 1) built a fully-lexicalized grammar inducer

[9] All of Headden et al.'s (2008) grammar induction experiments with induced parts-of-speech were worse than their best results using gold part-of-speech tags, most likely because they used a very small corpus (half of WSJ10) to cluster words.

[10] We chose the sampling split (80:10:10) and replication parameter (100) somewhat arbitrarily, so better results could likely be obtained with tuning. However, we suspect that the real gains would come from using soft clustering techniques (Hinton and Roweis, 2003; Pereira et al., 1993, *inter alia*) and propagating (joint) estimates of tag distributions into a parser. Our ad-hoc approach is intended to serve solely as a proof of concept.

[11] David Elworthy's `C+ tagger`, with options `-i t -G -l`, available from `http://friendly-moose.appspot.com/code/NewCpTag.zip`.

[12] On the same evaluation set (WSJ10), our context-sensitive system without gold tags (Experiment #6, §5.2) scores 66.8%.

that was competitive with DMV+CCM despite not using gold tags. His CCL parser has since been improved via a "zoomed learning" technique (Reichart and Rappoport, 2010). Moreover, Abend et al. (2010) reused CCL's internal distributional representation of words in a cognitively-motivated part-of-speech inducer. Unfortunately their tagger did not make it into Christodoulopoulos et al.'s (2010) excellent and otherwise comprehensive evaluation.

Outside monolingual grammar induction, fully-lexicalized statistical dependency transduction models have been trained from unannotated parallel bitexts for machine translation (Alshawi et al., 2000). More recently, McDonald et al. (2011) demonstrated an impressive alternative to grammar induction by projecting reference parse trees from languages that have annotations to ones that are resource-poor.[13] It uses graph-based label propagation over a bilingual similarity graph for a sentence-aligned parallel corpus (Das and Petrov, 2011), inducing part-of-speech tags from a universal tag-set (Petrov et al., 2011).

Even in supervised parsing we are starting to see a shift away from using gold tags. For example, Alshawi et al. (2011) demonstrated good results for mapping text to underspecified semantics via dependencies without resorting to gold tags. And Petrov et al. (2010, §4.4, Table 4) observed only a small performance loss "going POS-less" in question parsing.

We are not aware of any systems that induce both syntactic trees and their part-of-speech categories. However, aside from the many systems that induce trees from gold tags, there are also unsupervised methods for inducing syntactic categories from gold trees (Finkel et al., 2007; Pereira et al., 1993), as well as for inducing dependencies from gold constituent annotations (Sangati and Zuidema, 2009; Chiang and Bikel, 2002). Considering that Headden et al.'s (2008) study of part-of-speech taggers found no correlation between standard tagging metrics and the quality of induced grammars, it may be time for a unified treatment of these very related syntax tasks.

[13] When the target language is English, however, their best accuracy (projected from Greek) is low: 45.7% (McDonald et al., 2011, §4, Table 2); tested on the same CoNLL 2007 evaluation set (Nivre et al., 2007), our "punctuation" system with context-sensitive induced tags (trained on WSJ45, without gold tags) performs substantially better, scoring 51.6%. Note that this is also an improvement over our system trained on the CoNLL set using gold tags: 50.3% (Spitkovsky et al., 2011, §8, Table 6).

7 Discussion and Conclusions

Unsupervised word clustering techniques of Brown et al. (1992) and Clark (2000) are well-suited to dependency parsing with the DMV. Both methods outperform gold parts-of-speech in supervised modes. And both can do better than monosemous clusters derived from gold tags in unsupervised training. We showed how Clark's (2000) flat tags can be relaxed, using context, with the resulting polysemous clustering outperforming gold part-of-speech tags for the English dependency grammar induction task.

Monolingual evaluation is a significant flaw in our methodology, however. One (of many) take-home points made in Christodoulopoulos et al.'s (2010) study is that results on one language do not necessarily correlate with other languages.[14] Assuming that our results do generalize, it will still remain to remove the present reliance on gold tokenization and sentence boundary labels. Nevertheless, we feel that eliminating gold tags is an important step towards the goal of fully-unsupervised dependency parsing.

We have cast the utility of a categorization scheme as a combination of two effects on parsing accuracy: a synonymy effect and a polysemy effect. Results of our experiments with both full and partial lexicalization suggest that grouping similar words (i.e., synonymy) is vital to grammar induction with the DMV. This is consistent with an established viewpoint, that simple tabulation of frequencies of words participating in certain configurations cannot be reliably used for comparing their likelihoods (Pereira et al., 1993, §4.2): "The statistics of natural languages is inherently ill defined. Because of Zipf's law, there is never enough data for a reasonable estimation of joint object distributions." Seginer's (2007b, §1.4.4) argument, however, is that the Zipfian distribution — a property of words, not parts-of-speech — should allow frequent words to successfully guide

[14] Furthermore, it would be interesting to know how sensitive different head-percolation schemes (Yamada and Matsumoto, 2003; Johansson and Nugues, 2007) would be to gold versus unsupervised tags, since the Magerman-Collins rules (Magerman, 1995; Collins, 1999) agree with gold dependency annotations only 85% of the time, even for WSJ (Sangati and Zuidema, 2009). Proper intrinsic evaluation of dependency grammar inducers is not yet a solved problem (Schwartz et al., 2011).

parsing and learning: "A relatively small number of frequent words appears almost everywhere and most words are never too far from such a frequent word (this is also the principle behind successful part-of-speech induction)." We believe that it is important to thoroughly understand how to reconcile these only seemingly conflicting insights, balancing them both in theory and in practice. A useful starting point may be to incorporate frequency information in the parsing models directly — in particular, capturing the relationships between words of various frequencies.

The polysemy effect appears smaller but is less controversial: Our experiments suggest that the primary drawback of the classic clustering schemes stems from their *one class per word* nature — and not a lack of supervision, as may be widely believed. Monosemous groupings, even if they are themselves derived from human-annotated syntactic categories, simply cannot disambiguate words the way gold tags can. By relaxing Clark's (2000) flat clustering, using contextual cues, we improved dependency grammar induction: directed accuracy on Section 23 (all sentences) of the WSJ benchmark increased from 58.2% to 59.1% — from slightly worse to better than with gold tags (58.4%, previous state-of-the-art).

Since Clark's (2000) word clustering algorithm is already context-sensitive in training, we suspect that one could do better simply by preserving the polysemous nature of its internal representation. Importing the relevant distributions into a sequence tagger directly would make more sense than going through an intermediate monosemous summary. And exploring other uses of *soft* clustering algorithms — perhaps as inputs to part-of-speech disambiguators — may be another fruitful research direction. We believe that a *joint* treatment of grammar and parts-of-speech induction could fuel major advances in both tasks.

Acknowledgments

Partially funded by the Air Force Research Laboratory (AFRL), under prime contract no. FA8750-09-C-0181, and by NSF, via award #IIS-0811974. We thank Omri Abend, Spence Green, David McClosky and the anonymous reviewers for many helpful comments on draft versions of this paper.

References

O. Abend, R. Reichart, and A. Rappoport. 2010. Improved unsupervised POS induction through prototype discovery. In *ACL*.

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

H. Alshawi, P.-C. Chang, and M. Ringgaard. 2011. Deterministic statistical mapping of sentences to underspecied semantics. In *IWCS*.

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

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

M. Banko and R. C. Moore. 2004. Part of speech tagging in context. In *COLING*.

L. E. Baum. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In *Inequalities*.

R. Bod. 2006. An all-subtrees approach to unsupervised parsing. In *COLING-ACL*.

P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18.

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

D. Chiang and D. M. Bikel. 2002. Recovering latent information in treebanks. In *COLING*.

C. Christodoulopoulos, S. Goldwater, and M. Steedman. 2010. Two decades of unsupervised POS induction: How far have we come? In *EMNLP*.

A. Clark. 2000. Inducing syntactic categories by context distribution clustering. In *CoNLL-LLL*.

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

B. Cramer. 2007. Limitations of current grammar induction algorithms. In *ACL: Student Research*.

D. Das and S. Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *ACL*.

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

J. R. Finkel, T. Grenager, and C. D. Manning. 2007. The infinite tree. In *ACL*.

J. Gao and M. Johnson. 2008. A comparison of Bayesian estimators for unsupervised Hidden Markov Model POS taggers. In *EMNLP*.

W. P. Headden, III, D. McClosky, and E. Charniak. 2008. Evaluating unsupervised part-of-speech tagging for grammar induction. In *COLING*.

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

G. Hinton and S. Roweis. 2003. Stochastic neighbor embedding. In *NIPS*.

R. Johansson and P. Nugues. 2007. Extended constituent-to-dependency conversion for English. In *NODALIDA*.

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

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

T. Koo. 2010. *Advances in Discriminative Dependency Parsing*. Ph.D. thesis, MIT.

J. Kupiec. 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language*, 6.

D. M. Magerman. 1995. Statistical decision-tree models for parsing. In *ACL*.

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

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

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

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

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

F. Pereira, N. Tishby, and L. Lee. 1993. Distributional clustering of English words. In *ACL*.

S. Petrov, P.-C. Chang, M. Ringgaard, and H. Alshawi. 2010. Uptraining for accurate deterministic question parsing. In *EMNLP*.

S. Petrov, D. Das, and R. McDonald. 2011. A universal part-of-speech tagset. In *ArXiv*.

R. Reichart and A. Rappoport. 2010. Improved fully unsupervised parsing with zoomed learning. In *EMNLP*.

F. Sangati and W. Zuidema. 2009. Unsupervised methods for head assignments. In *EACL*.

H. Schütze. 1995. Distributional part-of-speech tagging. In *EACL*.

R. Schwartz, O. Abend, R. Reichart, and A. Rappoport. 2011. Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In *ACL*.

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

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

B. Selman, H. A. Kautz, and B. Cohen. 1994. Noise strategies for improving local search. In *AAAI*.

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, 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*.

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

H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *IWPT*.

D. Yuret. 1998. *Discovery of Linguistic Relations Using Lexical Attraction*. Ph.D. thesis, MIT.