Learning Structural Dependencies of Words in the Zipfian Tail

Tejaswini Deoskar
School of Informatics
University of Edinburgh
tdeoskar@inf.ed.ac.uk

Markos Mylonakis
ILLC
University of Amsterdam
m.mylonakis@uva.nl

Khalil Sima'an
ILLC
University of Amsterdam
k.simaan@uva.nl

Abstract

Using semi-supervised EM, we learn fine-grained but sparse lexical parameters of a generative parsing model (a PCFG) initially estimated over the Penn Treebank. Our lexical parameters employ *supertags*, which encode complex structural information at the pre-terminal level, and are particularly sparse in labeled data – our goal is to learn these for words that are unseen or rare in the labeled data. In order to guide estimation from unlabeled data, we incorporate both structural and lexical priors from the labeled data. We get a large error reduction in parsing ambiguous structures associated with unseen verbs, the most important case of learning lexico-structural dependencies. We also obtain a statistically significant improvement in labeled bracketing score of the treebank PCFG, the first successful improvement via semi-supervised EM of a generative structured model already trained over large labeled data.

1 Introduction

Computational models of natural language trained on labeled data contain many parameters that are not estimated accurately, due to the data sparsity inherent in labeled data. This is especially true of complex structured models like parsers, which contain a large number of parameters, and where labeled training data is expensive to create.These models employ various forms of parameter smoothing to deal with overfitting and with unknown or low-frequency words. However, it is desirable, and in many cases necessary, to augment supervised models using readily available unlabeled data, such as raw news-wire or from the web. Semi-supervised methods have therefore received a lot of attention in recent years.

In this paper, we present a method for semi-supervised training of a large-scale structured model (a Penn Treebank PCFG) using the Expectation Maximization algorithm (Dempster et al., 1977). We focus on learning only those parameters of the model that are particularly difficult or impossible to obtain from labeled data, namely parameters related to *low-frequency* and *unseen* words (the Zipfian tail). Words are important determiners of structural information for parsers; for instance, verb subcategorization information improved the Collins' parser (Collins, 1997). However, this data is very sparse in even the largest labeled dataset available today, i.e., the Penn Treebank (Marcus et al., 1993). To illustrate the severity of the problem, consider the fact that close to 40% of verb types in the training sections of the Penn Treebank have occurred only *once* therein. Thus, modelling the structural properties of these verbs that may be useful for disambiguation in a parser (such as subcategorization properties) is simply not possible from labeled data, and one has to look to unlabeled data.

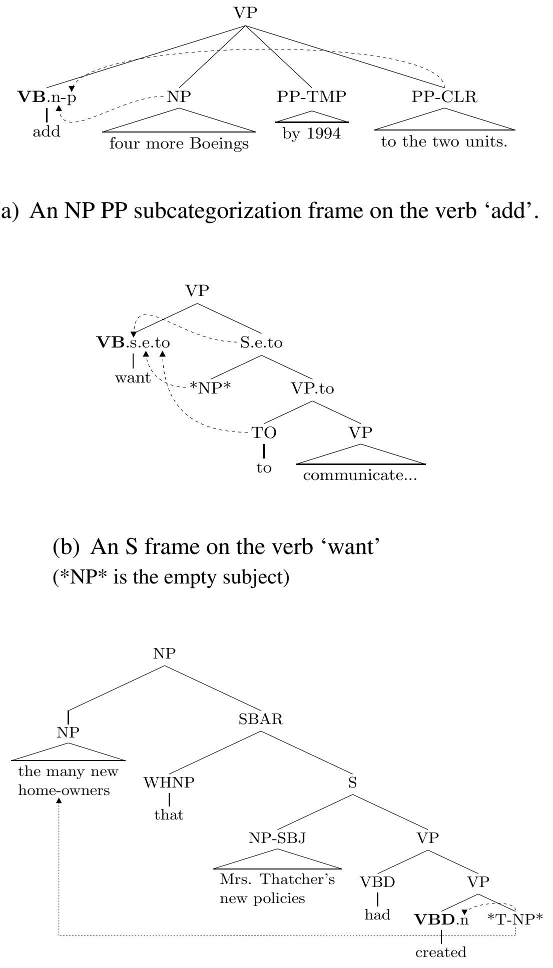
From the machine learning point of view, semi-supervised learning in general, and semi-supervised EM in particular, has been successful for classification-based NLP tasks (e.g. Nigam et al. (1998), Blum and Mitchell (1998), Yarowsky (1995)). For more structured tasks such as part-of-speech tagging and grammar learning, semi-supervised learning has worked largely in the case where the labeled data is small in size (Klein and Manning, 2004; Steedman et al., 2003; Druck et al., 2009a; Ganchev et al., 2010; Reichart and Rappoport, 2007). There have been some instances of successful large-scale semi-supervised learning for structured models (McClosky et al., 2006; Deoskar, 2008; Koo et al., 2008; Bansal and Klein, 2011), where a grammar model trained on a large amount of labeled data such as the full Penn Treebank has shown further improvement from unlabeled data. These methods have typically depended on the complementarity of multiple views

Proceedings of the 12th International Conference on Parsing Technologies, pages 80–91,
October 5-7, 2011, Dublin City University.

of the data (a discriminative reranking model over a generative model as in (McClosky et al., 2006)), and/or complex or heuristic objective functions (as in Deoskar (2008); Koo et al. (2008)) or simply by incorporating surface counts from unlabeled data (as in a recent paper by Bansal and Klein (2011)). A contribution of this paper is that we show that using EM in a semi-supervised manner with a simple objective function can improve a parser, contrary to common belief in the field.

The PCFG model used in this paper is trained on the Penn Treebank. It contains fine-grained structural information marked on pre-terminal categories, making them similar in spirit to *supertags* for strongly lexicalised formalisms like LTAG (Bangalore and Joshi, 1999) and CCG (Steedman, 2000). A supertag encodes structure that is distributed over the tree and localises it onto a single parameter of the model. Our learning problem is cast very simply as estimating the parameters $p(w|\tau)$ (where w is a word and τ a supertag) from labeled and unlabeled data. The problem is, however, more complex than a sequence labeling task because these supertags are highly ambiguous and encode argument-adjunct distinctions as well as long-distance dependencies (illustrated later in examples). Semi-supervised EM is known to often give models that are worse than the supervised model (Merialdo, 1994; Charniak, 1993; Ng and Cardie, 2003). To address this, we incorporate probabilistic constraints on unsupervised estimation by using labeled data to derive prior knowledge at two levels: (a) structural constraints in the form of higher PCFG rules (b) preferences over the distributions $p(w|\tau)$ themselves. We obtain large improvements in assigning correct structures to unseen verbs, and also a statistically significant improvement in labeled bracketing over a smoothed supervised model.

The rest of the paper is structured as follows: a description of the Treebank PCFG model and its smoothing is in §2. §3 describes the semi-supervised method, the constraints derived from labeled data, and their theoretical interpretation. §4 contains experiments and §5 evaluations. A discussion of related literature is in §6. §7 concludes.



(a) An NP PP subcategorization frame on the verb 'add'.

(b) An S frame on the verb 'want'
(*NP* is the empty subject)

(c) Long-distance. (*T-NP* is the trace of NP)

Figure 1: Some verbal supertags.

2 The PCFG Model

We work with a probabilistic context-free grammar (PCFG) model, since it is easy to analyse and most other more sophisticated parsing models can be understood as refinements of it (Charniak, 1997). The Penn Treebank PCFG used in this work is based on Deoskar and Rooth (2008) and Deoskar (2009) . It has pre-terminal categories that are complex and fine-grained, especially for open-class words. The PCFG is obtained by a process that effectively results in node-relabelling transformations of Penn Treebank II trees (Johnson, 1998), and counting relative frequencies of context-free rules in the transformed trees. We illustrate the nature of complex pre-terminal categories in the grammar with some examples below. These complex categories are intended to encode structure selected by/associated with a word onto the preterminal-tag of the word. Fig. 1 shows fragments of Penn Treebank (henceforth, PTB) sentences along with their annotation (empty categories are slightly simplified). In (a), the verb *add* has two arguments – an NP *four more Boeings* and a PP-CLR *to the two units*. The -CLR label indicates that the PP is an argument.

These arguments are encoded in the supertag on the verb as n-p giving the new pre-terminal category'**VB**.n-p', made of the original PTB POS-tag VB, followed after a dot by its refinement n-p indicating the NP and PP-CLR arguments. The temporal PP (PP-TMP) is considered an adjunct and is not included in the supertag. Fig. 1(b) shows a more complex supertag on the verb *want* – this supertag encodes the complement S as s, the empty subject of the S as e and the TO further down the tree as to, together forming s.e.to. The e serves to distinguish this structure from others like *expect them to communicate*, while the to distinguishes it from finite subordinate clauses like *set the economy moving* or *help meet increasing demand*. The final example in 1(c) shows an object relative clause. The verb of interest is 'created', which has a transitive supertag n indicating an NP complement. Notice that this verb is assigned the transitive supertag even though the complement NP is quite far removed from its original position (indicated by *T-NP*), thus capturing a long-distance dependency between the verb *created* and the NP *the many new home-owners*.

Our supertags are quite fine-grained – there are 81 sub-categories for verbs[1]. The additional marking on the original PTB POS tag is determined automatically and unambiguously by (solely) using information available in the treebank tree, such as the structure of the tree and functional tag marking. As seen above, these supertags distinguish arguments from adjuncts and localise onto a single parameter, long distance information that may be spread across different levels of the tree.

For space reasons, we do not describe aspects of the PCFG that are not directly relevant to this work (but see Deoskar (2009)). Importantly, the PCFG does not contain lexicalisation at higher levels of the tree, except for function words such as prepositions and determiners (as in (Klein and Manning, 2003)). As far as content-words (non-functional words) are concerned, word or head-word information is not part of any parameter of the PCFG except pre-terminal rules. Thus the unlexicalised PCFG has a clean division between complex lexical parameters (pre-terminal rules) and non-lexical ones (the rest). We exploit this in our semi-supervised method to constrain unsupervised estimation (§3). Another consideration in using an unlexicalised PCFG for this work is that it would be significantly more computationally expensive to use a lexicalized one, due to the larger number of parameters.

The (smoothed) PCFG performs close to the best reported results for a simple unlexicalised Treebank PCFG (without splitting and merging of categories as in Petrov and Klein (2007)), with a labeled bracketing f-score of 87.4% (< 40 words) and 86.5% (all sentences) on Section 23 of the PTB. While this is not the highest-performing grammar trained on the Penn Treebank (Petrov and Klein, 2007; Charniak and Johnson, 2005), note that it is trained on PTB trees that retain all functional categories as well as empty categories originally present in the PTB. Most treebank parsers remove functional tags and empty categories, thereby reducing sparsity and improving scores. Including functional categories and traces enables our PCFG to make finer distinctions and recover traces, but makes our training data much sparser than usual. Empty category recovery of the PCFG is 84%, at par with the state-of-the-art (Schmid, 2006). Functional tag recovery is comparable to Blaheta and Charniak (2000); Blaheta (2004) (the only other reported results that use all functional tags in the PTB [2]). Our non-null f-scores for the categories described in Blaheta's work are as follows (with the best scores from Blaheta and Charniak (2000) or Blaheta (2004) in brackets) – Grammatical: 94.78 (95.55), Semantic: 77.96 (78.63), Topicalization: 96.26 (95.28), Miscellaneous: 61.97 (58.99).

2.1 Smoothing the treebank PCFG based on POS tagging: creating a baseline.

Most treebank parsers are required to smooth their estimates to deal with over-fitting and with unknown words. This is usually done by backing off from a more articulated level (such as words) to a less articulated one (such as POS-tags), or by interpolating between the two. In the case of fine-grained lexical categories (supertags), the problem of smoothing becomes more severe. In some other generative models containing fine-grained lexical categories, such as CCG, smoothing is done by replacing unseen words and words below a cut-off

[1] This number holds for the case when lexicalized prepositions are not projected into the supertag. The complete list is available in Deoskar (2009) (Appendix D).

[2] Merlo and Musillo (2005)'s work uses a subset of the functional tags in the PTB, and hence their results are not comparable to ours.

frequency with POS tags. This cut-off frequency is in fact very high – for instance, Hockenmaier and Steedman (2002) find that the optimal cut-off is 30 for their generative parser. In our work, such a method is not an option: we are interested precisely in learning supertags for low frequency and unseen words from the unlabeled corpus. Secondly, POS tags are not a parameter of the PCFG, only supertags are.

We adopt a smoothing method first described in Deoskar (2008), that specifically aims at introducing parameters for unseen words from the unlabeled corpus into the PCFG[3]. In this method, every word from the unlabeled corpus is assigned with all those supertags that have been seen in the labeled corpus with the POS tag of the word. Thus, each verb is assigned all supertags that are associated with verbs in the labeled corpus. This applies both to words that are seen and unseen in the labeled data, thus taking care of the case where a word may have been seen in the labeled data, but may not have been seen with all relevant categories (an issue when dealing with fine-grained categories). A small probability mass is taken from the supervised distribution and redistributed amongst the newly introduced parameters. Equations and more details are in Deoskar (2008).

The unlabeled corpus is first POS-tagged by an off-the-shelf POS tagger to give counts of words and POS-tags. The count of a (word, POS-tag) pair from the unlabeled corpus is divided amongst all supertags (for that POS-tag) based on the ratio of supertags in the *labeled* data. For unseen words, this gives an initial estimate that is informed by marginal counts, counted over all words (with the given POS tag) in the labeled data. For instance, in the case of an unseen verb, the method will result, say, in the transitive supertag being more common than a ditransitive one, since transitive supertags are overall more common than ditransitive ones across all verbs in the labeled data. This model thus gives us an informed baseline to evaluate models learnt from the semi-supervised process, a baseline that is more informed than backing off to the part-of-speech of the word. This smoothed model also forms the initial model for the EM estimation described in the section below.

3 Semi-Supervised Learning of Lexical Parameters

3.1 The Learning Problem

EM is notoriously fickle for learning structured models in semi-supervised settings, needing tricky initialisation and careful constraining (Mann and McCallum, 2010) (e.g. Charniak (1993) for parsing, Merialdo (1994) for POS-tagging). In our case, the initial model is a highly-accurate, smoothed model obtained from labeled data (§2.1). Our task is to retrieve an estimate from the joint corpus of labelled and unlabeled data that performs better than a smoothed estimate from labeled data alone. In our unlexicalised PCFG, grammatical parameters (i.e., non-lexical rules) from the labeled data are fairly accurate [4]. We do not re-estimate these from unlabeled data (following Deoskar (2008)). Instead, we solely re-estimate lexical parameters, which are complex and contain a lot of structural information localised onto the pre-terminal level of the tree (recall the examples in Fig. 1). This allows us to learn syntactic information, while keeping the learning problem adjacent to the lexical surface.

In the following sections, we describe two ways in which we use the labeled data to constrain our latent variable (preterminal supertag sequences): structural constraints in the form of non-lexical rules, and distributions over lexical parameters $p(w|\tau)$ themselves. These constraints are included in a well-founded manner: a structural probabilistic prior over supertag sequences, and Dirichlet priors over conditional distributions $p(w|\tau)$ (as seen later in §3.5, by interpreting the learning process as a maximum a posteriori unsupervised estimator). These priors direct the estimator towards more promising parameter spaces, creating a strong learning environment with a clear objective function.

3.2 A Prior Over Supertag Sequences

Notation:

w : terminal (word)	TB : labeled corpus
τ : pre-terminal	UC : unlabeled corpus

$\mathbf{w}$: sequence of terminals
$\boldsymbol{\tau}$: sequence of pre-terminals
$p(\boldsymbol{\tau})$: distribution over $\boldsymbol{\tau}$
$\hat{p}(\boldsymbol{\tau})$: relative frequency estimate of $p(\boldsymbol{\tau})$

[3] It is important for unsupervised estimation that the PCFG contain non-zero lexical parameters for all words in the unlabeled corpus. If not, sentences with unseen words will not get an analysis and parameters for those words will never be induced.

[4] This assumption is justified to a large extent in the case of an unlexicalised grammar; however, grammar rules are also subject to sparsity and may benefit from re-estimation.

$\tau := \langle T, \iota \rangle$ consists of a POS-tag T and a sequence of features ι.

A PCFG, apart from defining a language and distribution over terminal strings, *also* does so for strings of pre-terminal symbols [5]. If we consider derivations down to the level of pre-terminals, the (syntactic part of the) PCFG provides a distribution $p(\boldsymbol{\tau})$ over sequences of pre-terminals $\boldsymbol{\tau}$. If $\mathcal{T}(\boldsymbol{\tau})$ is the set of trees with $\boldsymbol{\tau}$ as their leaves, then $p(\boldsymbol{\tau})$ is the sum of probabilities of all such trees $p(\boldsymbol{\tau}) = \sum_{\mathcal{T} \in \mathcal{T}(\boldsymbol{\tau})} p(\mathcal{T})$.

We are concerned with estimating the conditional probabilities $p(w|\tau)$, i.e., the parameters of the conditional model $p(\mathbf{w}|\boldsymbol{\tau})$. We use Maximum-Likelihood Estimation (MLE) for this purpose. For the labeled part of the data TB, MLE boils down to simply getting the relative-frequency estimate. For the unlabeled data UC however, we need to marginalise over all plausible pre-terminal sequences $\boldsymbol{\tau}$. As a consequence, $p(\boldsymbol{\tau})$ directly emerges as a *prior* over the latent variable $\boldsymbol{\tau}$. The likelihood of the concatenation of the two corpora can then be written as follows, with θ the set of lexical parameters $p(w|\tau)$:

$$\mathcal{L}(TB, UC;\ \theta, p(\boldsymbol{\tau})) = \prod_{\langle \mathbf{w}, \boldsymbol{\tau} \rangle \in TB} p(\mathbf{w}|\boldsymbol{\tau};\theta)p(\boldsymbol{\tau}) \ * \prod_{\mathbf{w} \in UC} \sum_{\boldsymbol{\tau}} p(\mathbf{w}|\boldsymbol{\tau};\theta)p(\boldsymbol{\tau}) \qquad (1)$$

In general, this approach allows semi-supervised MLE training of a model conditioning on a latent variable, by introducing a prior over the latent variable which can be directly estimated from the labeled part of the training data. For our model, after computing a PCFG relative-frequency estimate for the parameters $\hat{p}(\boldsymbol{\tau})$ on TB, we can shift our focus away from the syntactical analyses in TB and effectively treat this part of the data as a corpus of sentences labeled with pre-terminal sequences.

3.3 MLE with Semi-Supervised EM

We estimate the parameter set θ of the conditional model $p(\mathbf{w}|\boldsymbol{\tau})$ by maximising the likelihood of the concatenation of the labeled and unlabeled corpus. During the estimation we employ the estimate $\hat{p}(\boldsymbol{\tau})$ that we retrieve from the labeled corpus TB as a prior over $\boldsymbol{\tau}$, i.e., its parameters are not a subject of the estimation process and remain constant. On the contrary, $\hat{p}(\boldsymbol{\tau})$ *guides* the estimation process, showing a strong preference towards supertag sequences which are syntactically justified.

```
Initialise $\theta_0$
for $i = 1$ to $N$ iterations do
  E-step {Find expected complete-data log-likelihood, given current estimate}
    $Q(\theta|\theta_{i-1}) =$
    $E[log(\mathcal{L}(TB, \langle UC, UC_{\boldsymbol{\tau}} \rangle; \theta, \hat{p}(\boldsymbol{\tau})))|TB, UC, \theta_{i-1}]$
  M-step {Maximise $Q$ in respect to $\theta$}
    $\theta_i = \arg\max_{\theta} Q(\theta|\theta_{i-1})$
end for
```

Figure 2: The EM algorithm for the semi-supervised learning of $p(w|\tau)$

Since $\boldsymbol{\tau}$ is a latent variable for the unlabeled corpus UC, $\arg\max_{\theta} \mathcal{L}(TB, UC; \theta, \hat{p}(\boldsymbol{\tau}))$ cannot be found analytically. Instead, we use the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). We start with an initialisation point θ_0, which, since we have labeled data available, is the (smoothed) relative-frequency estimate of these parameters on TB.

E-step In the *Expectation* step, we find the expected value $Q(\theta|\theta_{i-1})$ of the complete data log-likelihood (with UC completed with missing pre-terminal sequences $UC_{\boldsymbol{\tau}}$) with respect to the missing data (pre-terminal sequences), given the observed data (sentences in UC, $\langle \mathbf{w}, \boldsymbol{\tau} \rangle$ pairs in TB) and the current estimate of the parameters θ_{i-1}. Since the sentences in TB already have supertags, in practice this step relates only to UC.

M-step In the *Maximization* step, the new estimate θ_i is retrieved by maximising the expectation of the E-step. The M-step under the constraints $\sum_w p(w|\tau) = 1$ can be performed analytically. This involves computing the expected counts of word-supertag pairs $c_{i-1}(w, \tau)$ over the combined corpus of labeled and unlabeled data, given θ_{i-1}. This is equivalent to adding the observed word-supertag counts from the labeled data to the expected counts from the unlabeled part, which can be efficiently computed using the Inside-Outside algorithm (Lari and Young, 1990). The update rule for the parameters of the new estimate θ_i are:

$$\theta_i(w|\tau) = \frac{c_{i-1}^{UC}(w, \tau) + c_{i-1}^{TB}(w, \tau)}{\sum_{w'} c_{i-1}^{UC}(w', \tau) + c_{i-1}^{TB}(w', \tau)} \qquad (2)$$

[5] Most PCFGs used in parsing employ pre-terminals, i.e., non-terminals which are the only ones which expand to terminal symbols and only terminal symbols. Even if a PCFG does not satisfy this requirement, it can be converted to an equivalent Chomsky Normal Form grammar which does so. Without loss of generality, we will here confine ourselves to a grammar making use of pre-terminals.

3.4 Corpora Scaling Factors and Additional Constraints

The impact of the labeled part of the data can be fine-tuned as follows: since the training data is seen as a concatenation of the labeled and unlabeled part, we can scale them before concatenating them, i.e., take a 'copies' of the unlabeled data together with b 'copies' of the labeled data. This operation can be understood as merely altering the input training corpus and has no effect on the existing analysis. In the new update formula, the scaling factors of the corpora trickle down as scaling factors of the (expected) counts:

$$\theta_i(w|\tau) = \frac{a * c^{UC}_{i-1}(w,\tau) + b * c^{TB}_{i-1}(w,\tau)}{\sum_{w'} a * c^{UC}_{i-1}(w',\tau) + b * c^{TB}_{i-1}(w',\tau)} \tag{3}$$

Secondly, we might also want to constrain the estimation objective by limiting the number of parameters of the conditional model $p(\mathbf{w}|\boldsymbol{\tau})$ to be estimated. Many lexical parameters are estimated accurately from the treebank (for example, those related to function words and other high-frequency words), and estimation from unlabeled data might hurt them. For each distribution $p(w|\tau)$, we choose to retain values from TB for some of the parameters which we assume are less affected by sparsity issues (i.e., we keep these parameters fixed) while estimating the rest. Under the same analysis as above, we end up with a similar update formula as before. For each conditional distribution given τ, if π_{fixed} is the sum of the fixed probability values and W^{τ}_{free} the set of words for which we wish to estimate $p(w|\tau)$, the remaining (i.e., not fixed) probability mass is $(1 - \pi_{\text{fixed}})$ and is distributed to the free parameters in proportion to the related (expected) counts $c(w,\tau)$. We skip the proof due to space limitations.

$$\theta_i(w|\tau) = \\ (1-\pi_{\text{fixed}}) \frac{a * c^{UC}_{i-1}(w,\tau) + b * c^{TB}_{i-1}(w,\tau)}{\sum\limits_{w' \in W^{\tau}_{\text{free}}} a * c^{UC}_{i-1}(w',\tau) + b * c^{TB}_{i-1}(w',\tau)} \tag{4}$$

3.5 Semi-Supervised Learning as Maximum A Posteriori Estimation

In this subsection, we discuss an interpretation of our learning method (i.e. maximum-likelihood of the concatenated labeled and unlabeled corpora) as *Maximum a Posteriori* (MAP) estimation solely on the unlabeled corpus employing a prior $p(\theta)$ over the parameter set θ. This is useful in order to understand the role that the labeled data plays in guiding estimation from unlabeled data. For each of the multinomials $p(\mathbf{w}|\boldsymbol{\tau})$, consider a Dirichlet conjugate prior with hyper-parameters α providing a distribution over the possible multinomial parameter sets.

$$p(\mathbf{w},\boldsymbol{\tau};\theta) = p(\mathbf{w}|\boldsymbol{\tau};\theta)p(\boldsymbol{\tau})p(\theta) \tag{5}$$

The hyper-parameters α of the Dirichlet can be interpreted as prior counts of the events that the multinomial tracks, with each α^{τ}_{w} corresponding to word w emitted by pre-terminal τ. We take advantage of this feature[6] to introduce relevant counts from the labeled corpus in the Dirichlet hyper-parameters, setting each $\alpha^{\tau}_{w} = c^{TB}(w,\tau) + 1$.

Dempster et al. (1977) show that EM can also be used under MAP to climb towards the posterior mode of the parameter space θ. Due to the Dirichlet being conjugate to the multinomial distribution, it is easy to show that the new quantity that we wish to maximise has the same functional form as $Q(\theta|\theta_{i-1})$. Interestingly, for the Dirichlet priors in Eq. (5), MAP estimation boils down to the same update formula as in (2), establishing an equivalent interpretation of the estimation process which clarifies how the labeled training data 'guide' EM estimation on the unlabeled part of the corpus at two distinct levels: (a) a structural prior $p(\boldsymbol{\tau})$ preferring syntactically correct pre-terminal sequences, considering the interdependencies between pre-terminals in a sentence and (b) priors over the parameter space itself $p(\theta)$, considering lexical choice for each pre-terminal separately.

4 Experiments

We report experiments using a treebank PCFG trained on approximately 36,000 sentences from sections 0-22 of the Wall Street Journal (WSJ) portion of the PTB, with about 5000 sentences held-out for testing and development. Semi-supervised training is carried out using 4, 8, 12 and 16 million words of unlabeled WSJ data, after limiting sentence length to <25 words. Inside-outside estimation is implemented in Bitpar (Schmid, 2004). The corpus scaling factor for labeled data is set to 8 (i.e., $a = 1$ and $b = 8$ in Eq. 3; this value makes our labeled data (1 million words) weigh about twice as much as our smallest unlabeled corpus of 4 million words. We experimented with setting the scaling factor to 4, making the labeled corpus of 1

[6] Starting from an uninformed Dirichlet prior $p(\theta)$ with $\alpha^{\tau}_{w} = 1$ for all w, τ, the posterior $p(\theta|TB)$ after observing the labeled data TB also takes the form of a Dirichlet distribution with updated hyper-parameters $\alpha^{\tau}_{w} = c^{TB}(w,\tau) + 1$.

	4M	8M	12M	16M
t_{smooth}	29.86	29.86	29.86	29.86
t_{parse}	27.80	27.82	27.80	27.80
It 1	28.44	28.12	27.16	27.64
2	27.72	27.08	26.13	25.73
3	27.40	26.53	25.89	25.34
4	27.40	26.21	25.97	25.18
5	27.24	**25.89**	25.66	24.7
6	**27.08**	26.05	25.81	24.78
7	27.08	26.05	25.50	24.7
8	-	-	25.42	24.62
9	-	-	25.42	**24.62**
10	-	-	**25.18**	-
11	-	-	25.42	-
% Err.reduc	9.31	12.76	15.67	**17.5**

Table 1: Supertag error for *unseen* verbs in test Viterbi parses, for different sizes of unlabeled training data

million words effectively equal in size to the unlabeled corpus of 4 million words; however, a value of 8 gives better results, and we report only these.

5 Evaluations

Learning lexico-syntactic information

We evaluate the learning of lexico-syntactic dependencies by measuring the accuracy of supertag assignment in Viterbi (maximum-probability) parses of test sentences. We report this number for verbs, since they are the most important lexical determiners of structure in a sentence, as well as the most ambiguous. To evaluate unseen verbs, we created a separate testset of 1200 sentences with about 1250 token occurrences of unseen verbs (about 110 types), by holding out all sentences with occurrences of these verbs from the labeled data. These verbs have a wide variety of ambiguous subcategorization frames and are thus representative of typical verbs in the lexicon of a language. This evaluation is a parsing-based evaluation and gives us a focused way of measuring the learning of syntactic structures associated with unseen words (verbs in this case). Note that each supertag is associated with a local or non-local structure, and hence counting supertag accuracy in effect measures the accuracy of getting this subtree-structure right. Since these supertags encode empty categories and functional tags, it is not possible to compare other standard state-of-the-art parsers on this metric, since they do not contain either in their output. Table 1 shows the error in identifying the correct supertag for these unseen verbs in Viterbi parses of test sentences, for unlabeled training data of sizes 4, 8, 12 and 16 million words. The baseline model is the smoothed treebank PCFG t_{smooth} (§2.1), with an error of 29.86%. This model does not contain lexical information specific to these verbs (being unseen). Thus, in about 70% of the cases the parser assigns a correct supertag without verb-specific information. We create a second baseline by parsing the unlabeled corpus with the model t_{smooth} and obtaining Viterbi parses – this parsed corpus is merged with the labeled data, keeping corpus scaling factors same as before, and a PCFG t_{parse} extracted from it. This model is thus a self-trained model – it improves the supertag error over t_{smooth} to 27.8%, and does not change subsequently.

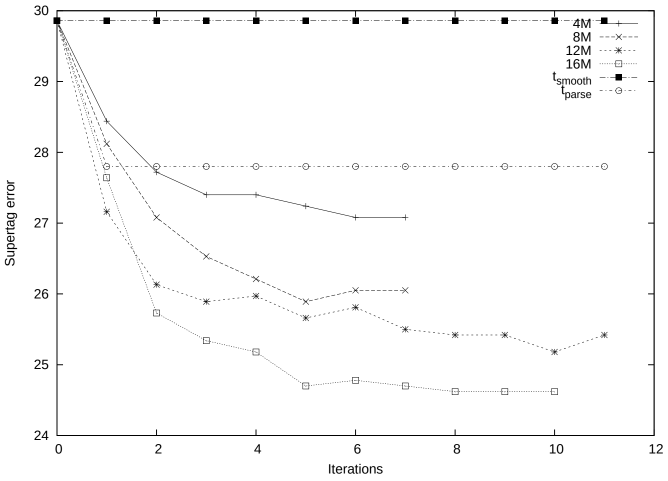


Figure 3: Supertag error for unseen verbs in Viterbi parses, for different sizes of unlabeled training data.

Semi-supervised EM training improves the error rate over t_{smooth} in the first iteration, and t_{parse} in the second. This improvement is already significant ($p < 0.01$, using McNemar's test). The error rate goes on to further improve in subsequent iterations. The error rate also improves with increasing sizes of unlabeled data. The best obtained error is 24.62% with 16M words ($p < 0.0001$), a substantial error reduction of 17.5% over the smoothed supervised model. Since these verbs have not occurred in the labeled data, the improvements are solely the result of learning from unlabeled data. We also evaluated seen but low-frequency verbs (frequency 1 to 5 in the training corpus). We see a benefit for these as well, with an error reduction of 8.97% (from 23.51 for the baseline t_{smooth} to 21.40 for 16M words of unlabeled data).

Figure 3 shows the learning curves for different sizes of unlabeled data. The distance between the 12M and 16M curves suggests that further improvements may be obtained by adding even more

	t_{smooth}	It 1	2	3	4	5	6	7	8
Recall	86.49	86.74	***86.83	86.79	86.79	86.78	86.80	86.79	86.79
Precision	86.84	86.84	**86.90	**86.90	86.83	86.86	86.88	86.88	86.87
f-score	86.56	86.79	**86.87	86.83	86.82	86.82	86.82	86.84	86.83

Table 2: Labeled bracketing f-scores on Sec. 23 of PTB (4M words, $f < 5$). *** p<0.001, ** p<0.01

It	$f < 5$	$f < 10$	$f < 20$	$f < 50$	$f < 1000$
t_{smooth}	18.13	18.13	18.13	18.13	18.13
1	17.78	17.82	17.79	17.68	17.65
2	18.14	17.63	17.63	17.65	17.65
3	18.43	17.65	17.70	17.67	17.65
4	18.14	17.74	17.75	17.67	17.70
5	**17.53**	17.72	17.74	17.81	17.68
6	17.65	17.81	17.84	17.79	17.75
7	17.68	17.81	17.87	17.84	17.84

Table 3: Overall verbal supertag error, 4M words unlabeled data. (It=iteration)

unlabeled data.

Labeled bracketing

The PARSEVAL metric is not the best metric for evaluating the lexico-syntactic learning that is the focus of this paper, for two reasons. Firstly, it is a coarse metric, known to be insensitive to lexico-syntactic (i.e. subcategorization) information (Briscoe et al., 1998), in addition to not counting argument/adjunct distinctions, functional tags or empty categories. Secondly, and more importantly, our method is targeted towards the learning of rare/low-frequency events, which do not have enough of a token count in Section 23 to make a big impact. However, we do see a statistically significant improvement in labeled bracketing scores on Section 23 of the PTB (Table 2) (statistical significance calculated using a randomised version of the paired-sample t-test).

The improvements are not large, however they are the first improvements to be obtained using semi-supervised EM for a large-scale Penn Treebank grammar. This is the result solely of learning lexical parameters of low-frequency words ($f < 5$). It is not surprising that the improvements are small – the total token count of words that our method impacts (i.e., words with a frequency less than 5 in the training data) constitute only 6.1% tokens in Section 23 (excluding numbers, but including proper nouns, for which it is not useful to learn structural dependencies). However, they correspond to about 34.2% types, relevant for a obtaining a broad-coverage lexicon, but not relevant for a token-based evaluation like labelled bracketing. It should be noted that while models in later iterations are not better than the baseline, nor are they significantly *worse*.

Another important point is that the f-score on Section 23 remains stable when the value of the cut-off frequency f is increased, and when unlabeled data size is increased to 16M words (not shown in table). Thus, although we obtain large improvements in learning about unseen words (as shown in the previous evaluation), the overall quality of the models, as measured by labeled bracketing does not degrade. This is an important consideration for semi-supervised learning, since adding large amounts of unlabeled data tends to have a negative impact on the supervised model.

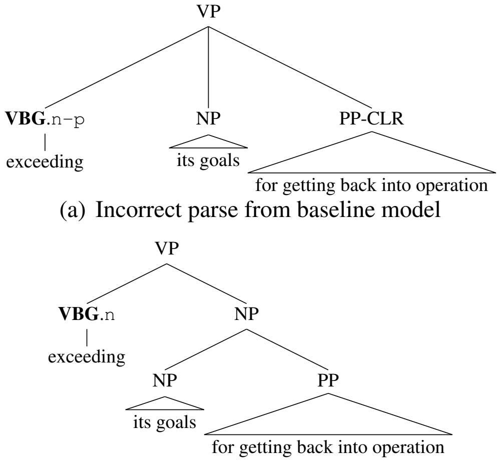
Making more parameters free

We experimented with making more and more lexical parameters free, by changing the cut-off frequency f[7]. Surprisingly, this does not affect the learning process much. The best model is obtained with $f < 5$, in terms of labeled bracketing scores, supertag accuracy for unseen verbs, as well as overall supertag accuracy for verbs (seen and unseen). Table 3 shows the overall supertag error for all verbs (seen and unseen) for different values of f. When high-frequency parameters are subject to unsupervised estimation, the error rate degrades by a small amount, but not much, even for $f < 1000$. Thus, the structural constraints plus the current corpus scaling factor (of eight) that scales up the size of labeled data are together sufficient to keep these estimates in the right ballpark, with the cut-off frequency not playing much of a role. This will be relevant to future work since it opens up the possibility of learning even mid-to-high frequency lexical items using this methodology.

5.1 Analysis

We present some examples of incorrect parses by the baseline model t_{smooth}, and corresponding improved parses by a semi-supervised EM-trained model (10 iterations, 12M words unlabeled data). These examples also serve to illustrate exactly what is captured by measuring supertag accuracy (our main evaluation). Fig. 4 shows improve-

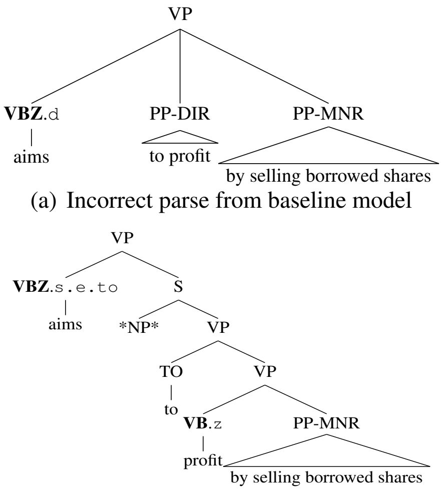
[7] f is the occurrence freq. of words in TB above which parameters are *fixed* i.e. estimates from unlabeled data are not used.



(a) Incorrect parse from baseline model

(b) Correct parse from EM-estimated model

Figure 4: Improvement in PP attachment.



(a) Incorrect parse from baseline model

(b) Correct parse from EM-estimated model

Figure 5: Detection of an S structure for *aims*

ments in a common PP attachment case (some categories are simplified for clarity). This improvement is due to learning a distribution for the unseen verb *exceeding* that represents its subcategorization preference for 'NP' rather than 'NP PP' (**VBG**.n supertag in (b) as opposed to **VBG**.n-p in (a)). Fig. 5 shows the improvement in assigning a more complex supertag. In (a), *to profit* is incorrectly parsed as a directional PP, and the verb *aims* is assigned an incorrect supertag **VBZ**.d (*d*irectional complement). The EM-trained grammar gives the correct parse – the correct supertag **VBZ**.s.e.to is assigned to *aims*, with the associated structure of an S with a empty subject *NP* and an infinitival (to) VP. Additionally, *profit* is now correctly detected as a verb and assigned an intransitive supertag (**VB**.z in our notation).

6 Related Work

We compare our work to prior research along several dimensions: for instance, the use of semi-supervised EM for a complex structured model, the aspect of using labeled data to constrain or guide estimation from unlabeled data, and the use of unlabeled data to improve an already accurate, high baseline treebank parser.

Semi-supervised learning for a generative model employing the EM-algorithm was already introduced in (Miller and Uyar, 1996). It has been applied to text classification before (Nigam et al., 1998, 2006) (we derive our inspiration from this work), but has not been successful with more complex NLP tasks such as parsing. In contrast to text classification, where the latent variable is the document class (amongst a few tens of classes), our latent variable (pre-terminal supertag sequences) is much richer in nature and takes an unbounded number of values. While in Nigam et al. (2006) a simple multinomial prior over document classes is part of the joint model and is itself trained, we have a rich structural prior obtained from labeled data which is kept fixed. In addition, Nigam et al. (2006) make use of a uniform Dirichlet prior over the model parameters. Instead, we utilise the labeled corpus to impose an informed Dirichlet prior over model parameters with a preference for configurations closer to the relative-frequency estimate of the labeled data.

Recently, there has been a lot of focus on semi-supervised methods that can incorporate constraints on latent variables based on prior knowledge, either in the form of labeled data or by other forms of indirect supervision. Ganchev et al. (2010); Graca et al. (2007) present the Posterior Regularization framework, which incorporates data-dependent constraints encoded as model posteriors on the observed data. The Generalized Expectation criteria (Mann and McCallum, 2010, 2007) incorporates weakly labeled data or 'side-information' such as marginal label distributions to inform estimation from unlabeled data. These methods have been shown to work for some structured tasks but have not been applied to a large scale grammar yet, and whether they can be used to improve a high baseline model is an open question.

There is also a substantial body of work on supertagging (Bangalore and Joshi (1999); Clark and Curran (2004), amongst several others), but their focus has been on improving parsing *efficiency*. Some other work focuses on unsupervised learning, but not for high-baseline supervised models (for instance, Dridan and Baldwin (2010); Ravi et al. (2010)).

The current work is most similar to Deoskar (2008) who used a treebank PCFG with Inside-outside to obtain ML estimates from an unlabeled corpus with an intention similar to ours: to learn lexico-syntactic dependencies. Their method gave improved results, with error reductions of up to 31.6% on the supertag detection task (we are not able to compare absolute numbers, since their treebank model is different from ours). Their approach was based on frequency transformations of inside-outside counts at each iteration: these transformations ensured that unsupervised estimates did not diverge far from the original treebank estimates, playing the same role as our priors. Their method did not have an interpretation in terms of a well-understood objective function; it is therefore not clear whether it has general applicability, or will extend to larger unlabeled data. The current work, although it shows somewhat more modest improvements, overcomes these shortcomings.

McClosky et al. (2006) enhance the performance of a state-of-the-art parser-reranker combination by self-training on large amounts of unlabeled data. Much of the improvement in their case comes from the ability of an external maximum-entropy Parse Reranker (Charniak and Johnson, 2005) to select parses from the parser's output for the unannotated sentences. Our work differs from McClosky et al. (2006) in that, firstly, they employ a fully lexicalized parser, whereas our parser is unlexicalised with supertags as pre-terminals. We are thus isolating lexico-syntactic dependencies, rather than word-word dependencies. All our improvements come from enhancing the lexical component of the PCFG. They find in their analysis that lexical learning does not play a large role in the improvements they obtain. Secondly, in contrast with their somewhat complex self-training objective, we retrain the parser under a well known and simple Maximum-likelihood objective. Koo et al. (2008) improved a dependency parser by using word clusters learnt from unlabeled data (an idea similar in some ways to learning supertag-word dependencies, since supertags form finer classes of words that POS tags do, but coarser than words), showing the utility of learning such statistics from unlabeled data. Most recently, Bansal and Klein (2011) improved the Berkeley parser (Petrov and Klein, 2007) by using surface counts from Google n-grams. The method proved very useful for some cases of parser disambiguation, but it is unlikely that surface counts alone can be used to learn long-distance or complex structural properties.

7 Conclusions

We have used semi-supervised EM to learn complex, ambiguous lexico-structural dependencies, obtaining large improvements for the hardest case of unseen verbs, as well as low-frequency verbs. We used a parser that uses all the information in the Penn Treebank, viz functional tags and empty categories. Learning such information is crucial for semantic analysis, besides being useful for syntactic disambiguation, but falls in the long Zipfian tail of linguistic events for which unlabeled data is the only learning source. We used labeled data to derive priors that guided estimation from unlabeled data to both the structural and lexical level in a principled manner. Our structural prior took the form of a PCFG; however it may be replaced by alternative, more complex models employing a different view on the labeled data.

This is the first instance of semi-supervised EM improving a complex structured model, and we believe the success is due to tightly constraining estimation from unlabeled data, as well as due to our complex lexical parameters that isolate structural information spread across a tree onto localised parameters of the model. The method has direct applicability to statistical grammars for strongly lexicalised formalisms like CCG and LTAG, of which statistical models suffer from severe sparsity and have not been successfully trained using semi-supervised methods. Another area of future work will be to incorporate supertags that encode other forms of lexico-structural dependencies, such as noun subcategorization or adverb attachment.

Acknowledgements

We thank Alexandra Birch, Mark Steedman, and three anonymous reviewers for detailed comments and suggestions. This research was supported by the VIDI grant 639.022.604 from The Netherlands Organisation for Scientific Research (NWO). The first author was further supported by the ERC Advanced Fellowship 249520 GRAMPLUS.

References

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An Approach to Almost Parsing. *Computational Linguistics*, 25:237–265.

Mohit Bansal and Dan Klein. 2011. Web-Scale Features for Full-Scale Parsing. In *Proceedings of ACL 2011*.

Don Blaheta. 2004. *Functional Tagging*. Ph.D. thesis, Department of Computer Science, Brown University.

Don Blaheta and Eugene Charniak. 2000. Assigning Function Tags to Parsed Text. In *ANLP'00*.

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of Conference on Computational Learning Theory (COLT) 1998*.

Ted Briscoe, John Carroll, and Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *1st Language Resources and Evaluation Conference*. Granada, Spain.

E. Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the 14th National Conference on Artificial Intelligence*, pages 598–603. AAAI Press/MIT Press, Menlo Park.

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

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of 43rd ACL*. Ann Arbor, Michigan.

Stephen Clark and James R. Curran. 2004. The Importance of Supertagging for Wide-Coverage CCG Parsing. In *Proceedings of COLING-04*, pages 282–288. Geneva, Switzerland.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th ACL*.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood estimation from incomplete data via the EM algorithm. *J. Royal Statistical Society*, 39(B):1–38.

Tejaswini Deoskar. 2008. Re-estimation of Lexical Parameters for Treebank PCFGs. In *Proceedings of COLING 2008*.

Tejaswini Deoskar. 2009. *Induction of fine-grained lexical parameters of treebank PCFGs with inside-outside estimation and frequency transformations*. Ph.D. thesis, Cornell University.

Tejaswini Deoskar and Mats Rooth. 2008. Induction of Treebank-Aligned Lexical Resources. In *Proceedings of 6th LREC, Marrakech, Morocco*.

Rebecca Dridan and Timothy Baldwin. 2010. Unsupervised Parse Selection for HPSG. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 694–704. Association for Computational Linguistics, Cambridge, MA.

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

Kuzman Ganchev, Joao Graca, Jennifer Gillenwater, and Bill Taskar. 2010. Posterior Regularization for Structured Latent Variable Models. *Journal of Machine Learning Research*, 11:2001–2049.

J Graca, K Ganchev, and B Taskar. 2007. Expectation Maximization and posterior constraints. In *NIPS 2007*.

Julia Hockenmaier and Mark Steedman. 2002. Generative Models for Statistical Parsing with Combinatory Categorial Grammar. In *ACL40*.

Mark Johnson. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4).

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

Dan Klein and Christopher Manning. 2003. Accurate unlexicalized parsing. In *ACL 41*. Sapporo, Japan.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple Semi-supervised Dependency Parsing. In *Proceedings of ACL-08: HLT*, pages 595–603. Association for Computational Linguistics, Columbus, Ohio.

K. Lari and S. J. Young. 1990. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 4:35–56.

G. Mann and A. McCallum. 2007. Simple, robust, scalable semi-supervised learning via expectation regularization. In *ICML 2007*.

Gideon Mann and Andrew McCallum. 2010. Generalized Expectation Criteria for Semi-Supervised LEarning with Weakly Labeled Data. *Journal of Machine Learning Research*, 11:955–984.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

D. McClosky, E. Charniak, and M. Johnson. 2006. Effective Self-Training for Parsing. In *Proceedings of HLT-NAACL 2006*.

Bernard Merialdo. 1994. Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2):155–171.

Paola Merlo and Gabriele Musillo. 2005. Accurate Function Parsing. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

David J. Miller and Hasan S. Uyar. 1996. A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 571–577.

Vincent Ng and CLaire Cardie. 2003. Weakly supervised natural language learning without redundant views.

Kamal Nigam, Andrew Mccallum, and Tom Mitchell. 1998. Learning to Classify Text from Labeled and Unlabeled Documents. In *Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 792–799. AAAI Press.

Kamal Nigam, Andrew Mccallum, and Tom Mitchell. 2006. *Semi-Supervised Learning*, chap. 3. MIT Press, Cambridge, Massachusetts.

Slav Petrov and Dan Klein. 2007. Improved Inference for Unlexicalized Parsing. In *HLT- NAACL 07*.

Sujith Ravi, Jason Baldridge, and Kevin Knight. 2010. Minimized Models and Grammar-Informed Initialization for Supertagging with Highly Ambiguous Lexicons. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 495–503. Association for Computational Linguistics, Uppsala, Sweden.

Roi Reichart and Ari Rappoport. 2007. Self-Training for Enhancement and Domain Adaptation of Statistical Parsers Trained on Small Datasets. In *ACL2007*.

Helmut Schmid. 2004. Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. In *COLING 2004*.

Helmut Schmid. 2006. Trace Prediction and Recovery with Unlexicalised PCFGs and Slash Features. In *21st COLING and 44th Annual Meeting of the ACL*, pages 177–184. Sydney, Australia.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press/Bradford Books.

Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proceedings of EACL03*.

D. Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *ACL95*.