Learning Structural Dependencies of Words in the Zipfian Tail

Tejaswini Deoskar  
*School of Informatics, University of Edinburgh,*  
Edinburgh, United Kingdom.  
tdeoskar@inf.ed.ac.uk

Markos Mylonakis  
*Institute for Logic, Language and Computation, University of Amsterdam,*  
Amsterdam, The Netherlands.  
m.mylonakis@uva.nl

Khalil Sima’an  
*Institute for Logic, Language and Computation, University of Amsterdam,*  
Amsterdam, The Netherlands.  
k.simaan@uva.nl

Abstract  

This paper uses semi-supervised Expectation Maximization (EM) to learn lexico-syntactic dependencies, i.e., associations between words and the structures that occur with them. Due to Zipfian distributions in language, such dependencies are extremely sparse in labeled data, and unlabeled data is the only source for learning them. Specifically, we learn sparse lexical parameters of a generative parsing model (a Probabilistic Context Free Grammar) that is initially estimated over the Penn Treebank. Our lexical parameters are similar to *supertags* — they are fine-grained, and encode complex structural information at the pre-terminal level. Our goal is to use unlabeled data to learn these for words that are rare or unseen in the labeled data. We get large error reductions (up to 17.5%) in parsing ambiguous structures associated with unseen verbs, the most important case of learning lexico-structural dependencies, resulting in a statistically significant improvement in labeled bracketing score of the treebank PCFG. Our semi-supervised method incorporates structural and lexical priors from the labeled data to guide estimation from unlabeled data, and is the first successful use of semi-supervised EM to improve a generative structured model already trained over large labeled data. The method scales well to larger amounts of unlabeled data, and also gives substantial error reductions (up to 11.5%) for models trained on smaller amounts of labeled data, making it relevant to low-resource languages with small treebanks.  

*Keywords:* semi-supervised EM, lexical learning, PCFG estimation, subcategorization.
1 Introduction

Statistical 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 (i.e., a treebank) is expensive to create. Although parameter smoothing can alleviate the problem of sparsity to some extent, 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 Probabilistic Context Free Grammar (PCFG) trained over the Penn Treebank, using the Expectation Maximization algorithm (Dempster et al., 1977). Our focus is on learning the structural properties of words, especially for words that are low-frequency or unseen in the labeled data (the Zipfian tail). Words are important determiners of structural information for parsers; for instance, verb subcategorization information improved the Collins’ parser (Collins, 1997). But this data is very sparse in even the largest labeled dataset available today, i.e., the Penn Treebank (Marcus et al., 1993). Thus, this problem is highly relevant even for high-resource languages like English which have a large treebank, let alone for low-resource languages, which may have treebanks of a much smaller size. 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. On the other hand, due to structural ambiguity inherent in language, it is not an easy task to obtain accurate models of structure from 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 Natural Language Processing tasks (Nigam et al., 1998; Blum and Mitchell, 1998; Yarowsky, 1995), but less so for structured tasks. 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 used complex or heuristic objective functions (Deoskar, 2008; Koo et al., 2008), or have depended on the complementarity of multiple views of the data (a discriminative reranking model over a generative model as in McClosky et al. 2006), or have simply incorporated surface counts from unlabeled data (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 contains fine-grained structural information marked on pre-terminal categories, making them similar in spirit to supertags for strongly lexicalised formalisms like Lexicalised Tree Adjoining Grammars (LTAG) (Bangalore and Joshi, 1999) and Combinatory Categorial Grammar (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 $\tau$ 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 result in 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 and low-frequency verbs (up to 17.5% error reduction), resulting in a statistically significant improvement in labeled bracketing score over a smoothed supervised model. We show that the method scales to larger sizes of unlabeled data — more unlabeled data gives better performing models. We also get substantial error reductions (up to 11.5%) for models trained with smaller sizes of labeled data, thus making the method useful for low-resource languages with smaller treebanks, parsers of which face an even more severe problem of sparse and missing lexico-syntactic dependencies.

The rest of this paper is structured as follows: §2 describes the Treebank PCFG model, the nature of representations in it, and its smoothing. §3 describes the semi-supervised method, the constraints derived from labeled data, and their theoretical interpretation. §4 contains our experimental setup, and §5 results and analyses. A discussion of related literature is in §6. §7 concludes.

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). 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. The node-relabelling transformations consist of automatic annotation of a few linguistically motivated features on the nodes of treebank trees, by using information in the rest of the tree. We do not have the space to describe all features in the PCFG; however, the examples in this section give some idea of the nature of the representations in the grammar (for more details, see Deoskar, 2009). For instance, verb phrases (VPs) are annotated as finite, infinite or present-participle (gerund) (VP.fin, VP.base, VP.g), amongst others, based on the part-of-speech of the daughter verb. Other examples of linguistically-motivated features are a feature that marks nouns as temporal or locative, based on functional tag marking on higher noun phrases or prepositional phrases, and a Slash feature that constrains the distribution of empty categories (similar to the Slash feature in Generalised Phrase Structure Grammar (GPSG), Gazdar et al., 1985). In addition to linguistic features, the PCFG also contains features which are tree-geometric but not linguistic in nature (in the style of Johnson (1998)’s parent feature, and Klein and Manning, 2003) – these are relevant to producing a good PCFG but do not necessarily have a linguistic interpretation. Together, the features result in an accurate PCFG.

An important property of the PCFG that is relevant to the present paper is that it has pre-terminal categories that are complex and fine-grained, especially for open-class words. These
complex categories are intended to encode structure selected by/associated with a word onto the pre-terminal tag of the word. The nature of complex pre-terminal categories for verbs is illustrated with some examples below. Fig.1 shows fragments of Penn Treebank (henceforth, PTB) sentences along with their annotation\(^1\). 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. The two 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 Fig. 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*.

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.

The supertags in the grammar are quite fine-grained – there are 81 sub-categories for verbs overall\(^2\). In general, the supertag for a verb can be broken down into three individual features: the first feature depends on the category of the sister/sisters of the verb, and takes 31 values. Examples of these values are intransitive, transitive (NP), predicative (-PRD), ditransitive (NP NP), prepositional (PP), NP-PP, directional (-DIR), and so on. The supertag on the verb *add* from Fig. 1(a) illustrates this feature, since it depends on the sisters of the verb. The second and third features are largely relevant only for clausal complements of the verb (that is, when the sister of the verb is an S or SBAR). An example of these is the supertag *VB.s.e.to* on the verb *want* in Fig. 1(b). The second feature can take three values and indicates the nature of the subject of a clausal complement: a non-empty subject, or the trace of A-movement (as in *want *NP* to communicate* above) or trace of A-bar movement (as in relative clauses). The final feature can take eight possible values and indicates the nature of the S complement of the verb. Its values are *fin* (tensed verb), *base* (base form), *to* (TO), *n* (passive past participle), *h* (non-passive past participle), *g* (gerund), *sc* (small clause), *scclr* (closely related small clause), and - (default). With the three features taken together, the verbal supertag encodes fairly complex information about the nature of the complement of the verb. The full specification with example sentences can be found in Deoskar (2009) (Appendix D).

\(^1\)Empty categories are slightly simplified.

\(^2\)This number holds for the case when lexicalized prepositions are not projected into the supertag. In general, our verbal frames are of comparable complexity to those in e.g. Korhonen (2002).
including empty categories and functional tags, as was seen in the examples before. Most Penn Treebank parsers do not retain all the annotation in the treebank, particularly doing away with empty categories and functional tags (for example, Petrov and Klein (2007); Charniak and Johnson (2005); Charniak (1997) and Collins (1997), except Model 3). This is because retaining this information results in statistical sparsity and lower parser performance. This information, however, may be necessary for parsers that give more sophisticated analyses and are capable of dealing with long-distance dependencies. We chose to use a PCFG whose representations do contain this information, since our goal is exactly to combat sparsity in labeled data by the use of unlabeled data. In this paper, the focus is on the learning method rather than representations in the grammar. We therefore do not experiment with representations, but select one of the best performing grammars of Deoskar (2009). However, it is an open question as to what is the nature of representations in the grammar or the lexicon that will help, or hinder, unsupervised learning from unlabeled data.

The second aspect of the PCFG that is relevant to this work is that it 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). A second important 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), recall that it is trained on PTB trees that retain all functional tags and empty categories originally present in the PTB. 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) and Blaheta (2004) which are the only other published results that use all functional tags in the PTB. 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 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. 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).

---

3 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.

4 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.
The unlabeled corpus is first POS-tagged by an off-the-shelf POS tagger (Treetagger, Schmid, 1994) 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 labeled 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. We therefore keep them fixed at their supervised values and do not re-estimate them from unlabeled data, following Deoskar (2008) who found that re-estimating both grammatical and lexical parameters from unlabeled data decreases performance. Instead, we solely re-estimate lexical parameters of the PCFG: the distributions \( p(w|\tau) \) of words \( w \) covered by each supertag pre-terminal \( \tau \). The supertags contain a lot of structural information localised at the pre-terminal level of the tree (recall the examples in Fig. 1). Limiting re-estimation to the lexical distributions \( p(w|\tau) \) allows us to learn syntactic information, but at the same time, keeps the learning problem adjacent to the lexical surface.

The following subsections contain the exact function that is optimized. We also describe two ways in which we use the labeled data to constrain our latent variable (preterminal supertag sequences), which is important in order to prevent the unlabeled data from corrupting the supervised estimates.

- structural constraints in the form of the structural part of the PCFG (which is not re-estimated). These enforce a preference for grammatical supertag sequences during estimation from the unlabeled part of the corpus.

- constraints over the distributions over lexical parameters \( p(w|\tau) \) themselves. These act as parameter space priors, preferring parameter values that do not deviate much from the supervised treebank estimate.

\footnote{This assumption is justified in the case of an unlexicalised grammar to a large extent; however, grammar rules are also subject to sparsity and may benefit from re-estimation.}
The following subsections show that these constraints are included in a well-founded manner: a structural probabilistic prior over supertag sequences \( p(\tau) \), and Dirichlet priors over conditional distributions \( p(w|\tau) \) (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 constrained learning environment with a clear objective function.

### 3.2 A prior over supertag sequences

**Notation**

- \( w \): terminal (word)
- \( \tau \): pre-terminal (supertag)
- \( w \): sequence of terminals
- \( \tau \): sequence of pre-terminals
- \( p(\tau) \): distribution over \( \tau \)
- \( \hat{p}(\tau) \): relative frequency estimate of \( p(\tau) \)
- \( TB \): labeled corpus
- \( UC \): unlabeled corpus
- \( \tau := \langle T, \iota \rangle \) consists of a POS-tag \( T \) and a sequence of features \( \iota \)

A PCFG, apart from defining a language and distribution over terminal strings, also does so for strings of pre-terminal symbols\(^6\). If we consider derivations down to the level of pre-terminals, the (syntactic part of the) PCFG provides a distribution \( p(\tau) \) over sequences of pre-terminals \( \tau \). \( p(\tau) \) is the sum of probabilities of all trees \( \mathcal{T}(\tau) \) which have \( \tau \) as their leaves (again considering derivations only down to the level of pre-terminals): \( p(\tau) = \sum_{\mathcal{T} \in \mathcal{T}(\tau)} p(\mathcal{T}) \).

We aim to estimate only the conditional lexical probabilities \( p(w|\tau) \), i.e., the parameters of the conditional model \( p(w|\tau) \). For this purpose, we use Maximum-Likelihood Estimation (MLE) on the concatenated corpus consisting of the labeled and unlabeled data taken together. The likelihood of the concatenation of the two corpora can be written as follows, with \( \theta \) the set of lexical parameters \( p(w|\tau) \):

\[
\mathcal{L}(TB, UC; \theta, p(\tau)) = \prod_{(w, \tau) \in TB} p(w|\tau; \theta) p(\tau) \times \prod_{w \in UC} \sum_{\tau} p(w|\tau; \theta) p(\tau)
\]

(1)

For the labeled part of the data \( TB \) (the first term in Equation 1), getting the conditional lexical probabilities \( p(w|\tau) \) boils down to simply getting relative-frequency estimates from the labeled data. For the unlabeled data \( UC \) however (the second term in Equation 1), we need to marginalise over all plausible supertag sequences \( \tau \) proposed for each sentence in \( UC \) by the (syntactic part of the) PCFG.

Note also that \( p(\tau) \) in the first term in Equation (1) is a constant, since for every labeled sentence in \( TB \), the supertag sequence \( \tau \) is known. In the second term however, where the supertag sequence \( \tau \) for a sentence of the unlabeled corpus is not known, \( p(\tau) \) can be considered to be a

---

\(^6\)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.
prior over the hidden variable $\tau$. This prior is estimated from the structural part of the treebank (which is entirely supervised, as noted before).

In general, this approach allows semi-supervised MLE training of a model conditioning on a hidden variable, by introducing a prior over the hidden variable which can be directly estimated from the labeled part of the training data. Since the syntactical preferences present in $TB$ are encoded in the prior $\hat{p}(\tau)$, 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 $\theta$ of the conditional model $p(w|\tau)$ by maximising the likelihood of the concatenation of the labeled and unlabeled corpus. As mentioned before, during the estimation we employ the estimate $\hat{p}(\tau)$ that we retrieve from the labeled corpus $TB$ as a prior over $\tau$, i.e., its parameters are not a subject of the estimation process and remain constant. On the contrary, $\hat{p}(\tau)$ guides the estimation process, applying a strong preference towards supertag sequences which are syntactically justified.

Since $\tau$ is a hidden variable for the unlabeled corpus $UC$, $\arg\max_\theta L(TB, UC; \theta, \hat{p}(\tau))$ cannot be found analytically. Instead, we use the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). We start with an initialisation point $\theta_0$, which, since we have labeled data available, is the (smoothed) relative-frequency estimate of these parameters on $TB$. Figure 2 gives the pseudo-code for the algorithm.

**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 $UC_\tau$) with respect to the missing data (pre-terminal sequences), given the observed data (sentences in $UC$, $\langle w, \tau \rangle$ pairs in $TB$) and the current estimate of the parameters $\theta_{i-1}$. Since the sentences in $TB$ are already labeled with supertag sequences, in practice this step relates only to $UC$.

**M-step** In the Maximization step, the new estimate $\theta_i$ 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 $\theta_{i-1}$. This is equivalent to adding the observed word-supertag counts from the labeled data to the expected counts from the unlabeled part, where the latter can be efficiently computed using the Inside-Outside algorithm (Lari and Young, 1990). The update rule for the parameters of the new estimate $\theta_i$ are:

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

### 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
Initialise $\theta_0$

for $i = 1$ to $N$ iterations do

E-step

{Find expected complete-data log-likelihood, given current estimate}

$$Q(\theta|\theta_{i-1}) = \mathbb{E}[\log(L(TB, \langle UC, UC_T \rangle; \theta, \hat{p}(\tau))|TB, UC, \theta_{i-1}]$$

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 $p(w|\tau)$

Existing analysis. In the new update formula, the scaling factors of the corpora trickle down as scaling factors of the (expected) counts:

$$\theta_i(w|\tau) = \frac{a \cdot c_{i-1}^{UC}(w, \tau) + b \cdot c_{i-1}^{TB}(w, \tau)}{\sum_{w'} a \cdot c_{i-1}^{UC}(w', \tau) + b \cdot c_{i-1}^{TB}(w', \tau)}$$  \hspace{1cm} (3)

Secondly, we might also want to constrain the estimation objective by limiting the number of parameters of the conditional model $p(w|\tau)$ 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 get a similar update formula as before. For each conditional distribution given $\tau$, if $\pi_{\text{fixed}}$ is the sum of the fixed probability values and $W_{\text{fixed}}$ the set of words for which we wish to estimate $p(w|\tau)$, the remaining (i.e., not fixed) probability mass is $(1 - \pi_{\text{fixed}})$. This mass is then distributed to the free parameters in proportion to the related (expected) counts $c(w, \tau)$, as shown in Equation (4) below.

$$\theta_i(w|\tau) = (1 - \pi_{\text{fixed}}) \frac{a \cdot c_{i-1}^{UC}(w, \tau) + b \cdot c_{i-1}^{TB}(w, \tau)}{\sum_{w' \in W_{\text{fixed}}} a \cdot c_{i-1}^{UC}(w', \tau) + b \cdot c_{i-1}^{TB}(w', \tau)}$$  \hspace{1cm} (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 $\theta$. 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 $p(w|\tau)$, let us consider a Dirichlet conjugate prior with hyper-parameters $\alpha$ providing a distribution over the possible multinomial parameter sets.

$$p(w, \tau; \theta) = p(w|\tau; \theta)p(\tau)p(\theta)$$  \hspace{1cm} (5)
The hyper-parameters $\alpha$ of a Dirichlet conjugate distribution can in general be interpreted as prior counts of the events that the multinomial tracks. In this case, each hyper-parameter $\alpha^\tau_w$ corresponds to prior counts registered for the event of a word $w$ emitted by pre-terminal $\tau$. We take advantage of this feature to introduce relevant counts from the labeled corpus in the Dirichlet hyper-parameters. We register a single prior count for every time a word $w$ is covered by a supertag $\tau$ in the treebank $TB_i$, setting each $\alpha^\tau_w = c^{TB_i}(w, \tau) + 1$.

Dempster et al. (1977) show that EM can also be used under MAP to climb towards the posterior mode of the parameter space $\theta$. 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). This establishes 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 $p(\tau)$ 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 Experimental setup

We ran the semi-supervised method on labeled and unlabeled data of different sizes. Inside-outside estimation is implemented in Bitpar (Schmid, 2004). We then parsed various testsets with the semi-supervised models thus obtained, again using Bitpar. Our main evaluation is focused on verbs, since verbs are the most important determiners of structure in a sentence, and the most ambiguous. Our training data is the following:

Labeled: For our main experiments (§5.1, §5.2, §5.3), we use a treebank PCFG trained on approximately 36,000 sentences from sections 0-22 of the Wall Street Journal (WSJ) portion of the PTB, with 1200 sentences held out from these sections to simulate test data containing unseen verbs — see subsection 5.1. In addition to that, we also hold out every tenth sentence from the training sections, to form a test and development set of about 4000 sentences. In §5.4, we describe experiments where the size of labeled treebank data is reduced to 24,000 and 16,000 sentences.

Unlabeled: The unlabeled data consists of WSJ sentences, after limiting sentence length to $< 25$ words. Out of these, we created training sets of different sizes: 4, 8, 12 and 16 million words.

We set the corpus scaling factor for labeled data to 8 (i.e., $a = 1$ and $b = 8$ in Eq. 3). This value makes our labeled data ($\approx 1$ 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 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.
|                | 4M  | 8M  | 12M | 16M |
|----------------|-----|-----|-----|-----|
| \( t_{\text{smooth}} \) | 29.86 | 29.86 | 29.86 | 29.86 |
| \( t_{\text{parse}} \)    | 27.80 | 27.82 | 27.80 | 27.80 |
| It 1           | 28.44 | 28.12 | 27.16 | 27.64 |
| It 2           | 27.72 | 27.08 | 26.13 | 25.73 |
| It 3           | 27.40 | 26.53 | 25.89 | 25.34 |
| It 4           | 27.40 | 26.21 | 25.97 | 25.18 |
| It 5           | 27.24 | 25.89 | 25.66 | 24.7  |
| It 6           | **27.08** | 26.05 | 25.81 | 24.78 |
| It 7           | 27.08 | 26.05 | 25.50 | 24.7  |
| It 8           | -    | -    | 25.42 | 24.62 |
| It 9           | -    | -    | 25.42 | **24.62** |
| It 10          | -    | -    | **25.18** | -     |
| It 11          | -    | -    | 25.42 | -     |
| % Err.reduc    | 9.31 | 12.76 | 15.67 | **17.5** |

Table 1: Supertag error for unseen verbs in Viterbi parses of test sentences, for unlabeled training data of different sizes — 4, 8, 12 and 16 million (M) words. (It=iteration)

5 Results

5.1 Learning lexico-syntactic information

In order to evaluate learning of verbal dependencies, we extracted a special testset of sentences containing a large number of unseen verbs. We first made a list of \( N \) verbs and then extracted all sentences containing them from the training sections (0-22) of the PTB. This effectively made these verbs unseen in the training data, and gave us a testset of 1200 sentences. The verbs in this list were chosen on the basis of occurrence frequency \( f \) in the training sections of the PTB — we selected all verbs that occurred with a frequency of 12 in the training sections. We chose this number as it gave us 110 verb types with 1250 token occurrences in the 1200 sentences, a large enough token count for reliable evaluation, and a good type/token ratio\(^8\). These verbs are mid-frequency verbs (neither very rare, nor very common) and can thus be considered representative of “typical” open-class verbs in the lexicon of the language\(^9\). They thus have a wide variety of subcategorization frames.

We measured the supertag assignment error of these verbs in Viterbi parses (maximum probability parses) of sentences in this testset. This evaluation is a parsing-based evaluation and gave us a focused way of measuring the learning of syntactic structures associated with unseen verbs. Note that each supertag is associated with a local or non-local structure, and hence counting supertag

\( ^7 \)Starting from an uninformed Dirichlet prior \( p(\theta) \) with \( \alpha_w^\tau = 1 \) for all \( w, \tau \), the posterior \( p(\theta|TB) \) after observing the labeled data \( TB \) also takes the form of a Dirichlet distribution with updated hyper-parameters \( \alpha_w^\tau = c^TB(w, \tau) + 1 \).

\( ^8 \)Selecting verbs of frequency 10 or 11 would have been just as good.

\( ^9 \)Some example test verbs are: accomplish acquires admit aims aired arguing asserts auctioned betting boasts buoyed combining completing concede converting cooperate coordinate coupled decides declaring defeated defended demanded demanding denying deserve...
accuracy in effect measures the accuracy of getting this sub-tree structure right. We measure the supertag assignment error of our semi-supervised models against our supervised baseline. These supertags encode empty categories and functional tags and it is therefore 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 sentences in this testset, using semi-supervised models obtained by using unlabeled training data of sizes 4, 8, 12 and 16 million words. The baseline model in each case is the smoothed treebank PCFG $t_{\text{smooth}}$ (§2.1), with an error of 29.86%. This model does not contain lexical information specific to these verbs (being unseen). Thus, in close to 70% of the cases the parser is able to assign a correct supertag to a verb without verb-specific information.

We also created a second baseline by parsing the unlabeled corpus with the model $t_{\text{smooth}}$ and obtaining Viterbi parses. The parsed corpus is merged with the labeled data, keeping corpus scaling factors same as before, and a PCFG $t_{\text{parse}}$ is extracted from it. This model is thus a self-trained model. It reduces the supertag error to 27.8%, and does not change subsequently.

Semi-supervised EM training reduces the error rate below $t_{\text{smooth}}$ in the first iteration, and $t_{\text{parse}}$ in the second. This reduction is already significant ($p < 0.01$, using McNemar’s test, McNemar, 1947). The error rate goes on to further reduce in subsequent iterations. The error rate reduces with increasing sizes of unlabeled data. The lowest error that we obtain is 24.62% with 16M words on unlabeled data ($p < 0.0001$), a substantial error reduction of 17.5% over the smoothed supervised model. Since these verbs have not occurred in the labeled data, the error reduction is solely the result of learning from unlabeled data.

Figure 3 shows the learning curves for different sizes of unlabeled data. A large part of the error reduction occurs in the first iteration. However, subsequent iterations also contribute significantly. The large gap between the 12M and 16M curves suggests that error reduction is not
saturated and further improvements may be obtained by adding more unlabeled data.

We also evaluated *seen but low-frequency* verbs (occurrence frequency of 1 to 5 in the labeled corpus TB). 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).

### 5.2 Labeled bracketing

We report labeled bracketing scores as a measure of the overall quality of our semi-supervised models\(^\text{10}\). However, it should be noted that 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 of the PTB to make a big impact. Despite this, we do see a statistically significant improvement in labeled bracketing scores on Section 23 (Table 2) (statistical significance calculated using a randomised version of the paired-sample t-test).

The improvements are not large. Nevertheless, they are the first improvements to be obtained using semi-supervised EM for a large-scale Penn Treebank grammar. This is solely the result of learning lexical parameters of unseen and 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.

Additionally, the f-score on Section 23 remains stable at the same value, both when the cut-off frequency $f$ is increased, and when unlabeled data size is increased (not shown in table). Thus, adding more unlabeled data benefits unseen words (as shown in the previous evaluation, §5.1) without degrading the overall quality of the models. 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. This makes the method all the more useful for learning about words in the Zipfian tail, where it is important that adding more and more unlabeled data does not degrade the existing model.

\(^{10}\)We do not report the overall supertag accuracy but instead use labeled bracketing f-scores as a measure of the overall quality of the semi-supervised models, since this is a standard evaluation metric for PCFGs. The overall supertag accuracy will include high-frequency, easy cases such as supertags for prepositions and determiners (which contain just the POS-tag with the preposition or determiner itself), which will cause this number to be inflated and not very informative.
Table 2: Labeled bracketing f-scores on Sec. 23 of PTB (4 million words unlabeled training data, \( f < 5 \)). *** \( p < 0.001 \), ** \( p < 0.01 \)

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|}
\hline
 & t_{\text{smooth}} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
\text{Recall} & 86.49 & 86.74 & ***86.83 & 86.79 & 86.79 & 86.80 & 86.79 & 86.79 \\
\text{Precision} & 86.84 & 86.84 & **86.90 & **86.90 & 86.86 & 86.88 & 86.88 & 86.87 \\
f-score & 86.56 & 86.79 & **86.87 & 86.83 & 86.82 & 86.82 & 86.82 & 86.84 & 86.83 \\
\hline
\end{array}
\]

Table 3: Overall verbal supertag error for different values of cut-off frequency (4M words unlabeled data. It=iteration)

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
It & f < 5 & f < 10 & f < 20 & f < 50 & f < 1000 \\
\hline
\text{t}_{\text{smooth}} & 18.13 & 18.13 & 18.13 & 18.13 & 18.13 \\
1 & 17.78 & 17.82 & 17.79 & 17.68 & 17.65 \\
2 & 18.14 & 17.63 & 17.63 & 17.65 & 17.65 \\
3 & 18.43 & 17.65 & 17.70 & 17.67 & 17.65 \\
4 & 18.14 & 17.74 & 17.75 & 17.67 & 17.70 \\
5 & 17.53 & 17.72 & 17.74 & 17.81 & 17.68 \\
6 & 17.65 & 17.81 & 17.84 & 17.79 & 17.75 \\
7 & 17.68 & 17.81 & 17.87 & 17.84 & 17.84 \\
\hline
\end{array}
\]

5.3 Making more parameters free

We also experimented with making more and more lexical parameters free, by changing the cut-off frequency \( f \). Surprisingly, this did not affect the learning process much. The best model is obtained with \( f < 5 \) (as shown in Table 3), in terms of overall supertag accuracy for all verbs (seen and unseen), as well as supertag accuracy for unseen verbs, and labeled bracketing scores. Table 3 shows the overall supertag error for all verbs (seen and unseen). It is clear that it does not change much for different values of \( f \) — even when high-frequency parameters are subject to unsupervised estimation (e.g., \( f < 1000 \)), the error rate degrades by a very small amount. Thus this parameter does not need to be tuned very finely — it seems that 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. This is relevant to future work because it opens up the possibility of learning even mid-to-high frequency lexical items from unlabeled data using this methodology, by setting the cut-off frequency to a high value.

5.4 Changing the size of labeled data

Finally, we wanted to see whether the method is useful in the case of smaller amounts of labeled data. We therefore ran experiments where the initial model was obtained from relative-frequency estimation over smaller sizes of Penn treebank data: 24,000 (24K) sentences and 16,000 (16K) sentences. The size of the unlabeled data is 4 million words in all cases, scaling factors are as before and we evaluate over the same test set of 1200 sentences containing unseen verbs as in subsection 5.1. Supertag error rates for different sizes of labeled data are shown in Table 4, with the results from the original grammar with 36,000 labeled sentences (from Table 1) repeated for

11 \( f \) is the occurrence freq. of words in \( TB \) above which parameters are fixed i.e. estimates from unlabeled data are not used.
Table 4: Supertag error for unseen verbs, for PCFGs obtained with different sizes of labeled data (It=iteration, Unlabeled data size = 4M words)

| Labeled Data Size | 36,000 sentences | 24,000 sentences | 16,000 sentences |
|-------------------|------------------|------------------|------------------|
| \( t_{\text{smooth}} \) | 29.86 | 34.39 | 35.11 |
| It 1 | 28.44 | 34.20 | 34.0 |
| It 2 | 27.72 | 31.77 | 33.28 |
| It 3 | 27.40 | 31.14 | 32.72 |
| It 4 | 27.40 | 30.98 | 31.90 |
| It 5 | 27.24 | 30.50 | 31.53 |
| It 6 | 27.08 | 30.58 | - |
| It 7 | 27.08 | 30.42 | - |
| Err. Reduc | **9.31** | **11.54** | **10.19** |

convenience. The baseline error (\( t_{\text{smooth}} \)) is of course larger when the size of the labeled data is smaller. The baseline error of the 24K model is much higher than that of the 36K model, but there is not much difference in the baseline errors of the 24K and the 16K model. Note that in all cases, the test verbs are unseen — thus, the increase in baseline error must come from the grammatical part of the model getting worse as the size of training data gets smaller, as well as worse lexical entries for other words in the sentence.

Semi-supervised EM benefits both the 24K and the 16K models. After seven iterations of semi-supervised EM, the model trained with 24K labeled sentences is almost able to catch up with the error rate of the larger model of 36K labeled sentences. The total error reduction is 11.54, as opposed to 9.31 for the 36K model, for the same amount of unlabeled data. There is also a substantial error reduction in the 16K model; the error rate is still decreasing with the 5th iteration, and running more iterations might allow it to catch up with the 24K and 36K models. The error reductions in the 24K and 16K models are somewhat larger than that in the 36K model — this could be due to larger relative proportion of unlabeled data as compared to the labeled data for these models. Also note that the unlabeled data in this experiment was limited to 4M words, and adding more unlabeled data might result in further error reduction.

The results of this section show that the method can be used to improve models trained with smaller amounts of labeled data — thus the method could be useful for low-resource languages which do not have a treebank as large as that of English.

### 5.5 Analysis

Finally, in order to give a more qualitative sense of the improvements obtained, we present some examples of incorrect parses by the baseline model \( t_{\text{smooth}} \), 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 improvements 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’ (\( \text{VBG}, n \) supertag in (b) as opposed to \( \text{VBG}, n \text{-p} \) in (a)). Fig. 5 shows the improvement in assigning a more
Figure 4: Improvement in PP attachment.

Figure 5: Detection of an S structure for *aims*
complex supertag. In (a), *to profit* is incorrectly parsed as a directional PP, and the verb *aims* is assigned an incorrect supertag **VBZ**, (directional complement). The EM-trained grammar gives the correct parse – the correct supertag **VBZ** s.e.t.o is assigned to *aims*, with the associated structure of an S with an 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

This work can be compared to prior research along at least three different dimensions. First is the use of semi-supervised EM for NLP tasks in general. Second is the aspect of using labeled data to constrain or guide estimation from unlabeled data, which is getting a lot of attention in recent times. The third is the use of unlabeled data to improve an already accurate, high baseline treebank parser trained over a large treebank like the Penn Treebank, which is distinct from the task of improving over a low-baseline parser trained over a small amount of labeled data. Below we survey the most relevant of related works.

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, 2009) 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 somewhat 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. The intuition that labeled data can be used to constraint estimation was particularly important in that work- for instance they used transformations to ensure that supertag marginals counted over all words from unsupervised counts remained in the same proportion as the treebank marginals over all words. This would ensure for instance, that if transitive supertags were more common than intransitive ones in the treebank (over all words), they would also remain so in the unsupervised estimate. We do not impose such direct constraints on unsupervised estimates but take a more principled semi-supervised approach. The method in Deoskar (2008, 2009) 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 (the main results of the current work were first published in Deoskar et al., 2011).

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.

Roark and Bacchiani (2003) also describe a semi-supervised method to adapt a parser for novel domains, based on self-training and Maximum a Posteriori estimation. They propose to encode the counts of PCFG rule applications in the treebank parses as hyper-parameters of a Dirichlet prior over the parameter space of the adapted model. In section 3.5, we show how such a methodology naturally emanates from applying semi-supervised EM for the model re-estimation task. While Roark and Bacchiani (2003) re-estimate parsing model parameters using solely the n-best parses of each unlabeled sentence, here we employ full semi-supervised EM to iteratively learn a parsing model from a combination of labeled and unlabeled data, exploring the full parse forests of unlabeled sentences in each iteration.

7 Conclusions

We have used a standard and well-understood technique, semi-supervised EM, to learn complex lexico-structural dependencies, obtaining large improvements for the hardest case of unseen verbs.
Increasing sizes of unlabeled data resulted in improved semi-supervised models, with error reductions of up to 17.5% for our largest unlabeled dataset, on supertag assignment of unseen verbs. The technique worked with smaller sizes of labeled data, making it useful for languages with smaller treebanks than English, and also benefitted verbs that were seen but low-frequency in the labeled data.

We used labeled data in a principled manner to derive priors that guided estimation from unlabeled data at both the structural and lexical level. Our structural prior took the form of a PCFG. However the learning method is general enough that the PCFG prior may be replaced by alternative, more complex models that will provide a different syntactic “view” on the labeled data. Additionally, the prior from labeled data may also weakened – for instance, the same methodology could be used for estimating selected sparse syntactic parameters in addition to lexical parameters.

This is the first instance of semi-supervised EM improving a complex structured model. We believe the success is due to tightly constrained estimation from unlabeled data, as well as due to our complex lexical parameters that localized structural information spread across a tree on pre-terminals of the PCFG. Complex lexical parameters allowed us to learn structural information and yet avoid re-estimating higher grammar rules. This may well be an advantage offered by strongly lexicalised grammars (like CCG and LTAG, in which lexical categories are complex but grammatical rules are simple), in learning from unlabeled data. Thus the method has direct applicability to statistical grammars for such formalisms, 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 additional forms of lexico-structural dependencies into the grammar. There are many of these in language that will benefit from semi-supervised estimation, such as noun subcategorization or adverb attachment.

Acknowledgements

We thank Mark Steedman, Alexandra Birch, three anonymous IWPT reviewers, and the JLC 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 the 48th meeting of the Association of Computational Linguistics (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 *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP-2000)*. Seattle, Washington.

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 the 43rd Annual Meeting of the Association for Computational Linguistics. Ann Arbor, Michigan.

Stephen Clark and James R. Curran. 2004. The Importance of Supertagging for Wide-Coverage CCG Parsing. In Proceedings of the 2004 International Conference on Computational Linguistics (COLING-2004), pages 282–288. Geneva, Switzerland.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-1997).

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 International Conference on Computational Linguistics (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, Markos Mylonakis, and Khalil Sima’an. 2011. Learning Structural Dependencies of Words in the Zipfian Tail. In Proceedings of the 12th International Conference on Parsing Technologies (IWPT 2011). Dublin, Ireland.

Tejaswini Deoskar and Mats Rooth. 2008. Induction of Treebank-Aligned Lexical Resources. In Proceedings of 6th Language Resources and Evaluation Conference, 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 Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and IJCNLP (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.

Gerald Gazdar, Ewan Klein, Geoffrey K. Pullum, and Ivan Sag. 1985. Generalized Phrase Structure Grammar. Blackwell, Oxford.

J Graca, K Ganchev, and B Taskar. 2007. Expectation Maximization and posterior constraints. In Proceedings of Advances in Neural Information Processing Systems (NIPS) 2007.
Julia Hockenmaier and Mark Steedman. 2002. Generative Models for Statistical Parsing with Combinatory Categorial Grammar. In Proceedings of the 40th meeting of the Association for Computational Linguistics.

Mark Johnson. 1998. PCFG models of linguistic tree representations. Computational Linguistics, 24(4):613–632.

D. Klein and C. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In Proceedings of 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004).

Dan Klein and Christopher Manning. 2003. Accurate unlexicalized parsing. In Proceedings of the 41st meeting of the Association for Computational Linguistics. Sapporo, Japan.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple Semi-supervised Dependency Parsing. In Proceedings of the 46th meeting of Association for Computational Linguistics/Human Language Technologies (ACL/HLT-2008), pages 595–603. Association for Computational Linguistics, Columbus, Ohio.

Anna Korhonen. 2002. Subcategorization Acquisition. Ph.D. thesis, Univ. of Cambridge.

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 Proceedings of the International Conference on Machine Learning (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 Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT/NAACL-2006).

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika, 12:153–157.

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. In Proceedings of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003).
