Unsupervised Structure Prediction with Non-Parallel Multilingual Guidance

Shay B. Cohen **Dipanjan Das** **Noah A. Smith**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{scohen,dipanjan,nasmith}@cs.cmu.edu

Abstract

We describe a method for prediction of linguistic structure in a language for which only unlabeled data is available, using annotated data from a set of one or more helper languages. Our approach is based on a model that locally mixes between supervised models from the helper languages. Parallel data is not used, allowing the technique to be applied even in domains where human-translated texts are unavailable. We obtain state-of-the-art performance for two tasks of structure prediction: unsupervised part-of-speech tagging and unsupervised dependency parsing.

1 Introduction

A major focus of recent NLP research has involved unsupervised learning of structure such as POS tag sequences and parse trees (Klein and Manning, 2004; Johnson et al., 2007; Berg-Kirkpatrick et al., 2010; Cohen and Smith, 2010, *inter alia*). In its purest form, such research has improved our understanding of unsupervised learning practically and formally, and has led to a wide range of new algorithmic ideas. Another strain of research has sought to exploit resources and tools in some languages (especially English) to construct similar resources and tools for other languages, through heuristic "projection" (Yarowsky and Ngai, 2001; Xi and Hwa, 2005) or constraints in learning (Burkett and Klein, 2008; Smith and Eisner, 2009; Das and Petrov, 2011; McDonald et al., 2011) or inference (Smith and Smith, 2004). Joint unsupervised learning (Snyder and Barzilay, 2008; Naseem et al., 2009; Snyder et al., 2009) is yet another research direction that seeks to learn models for many languages at once, exploiting linguistic universals and language similarity. The driving force behind all of this work has been the hope of building NLP tools for languages that lack annotated resources.[1]

In this paper, we present an approach to using annotated data from one or more languages (*helper* languages) to learn models for another language that lacks annotated data (the *target* language). Unlike the previous work mentioned above, our framework does not rely on parallel data in any form. This is advantageous because parallel text exists only in a few text domains (e.g., religious texts, parliamentary proceedings, and news).

We focus on generative probabilistic models parameterized by multinomial distributions. We begin with supervised maximum likelihood estimates for models of the helper languages. In the second stage, we learn a model for the target language using unannotated data, maximizing likelihood over *interpolations* of the helper language models' distributions. The tying is performed at the parameter level, through coarse, nearly-universal syntactic categories (POS tags). The resulting model is then used to *initialize* learning of the target language's model using standard unsupervised parameter estimation.

Some previous multilingual research, such as Bayesian parameter tying across languages (Cohen and Smith, 2009) or models of parameter

[1] Although the stated objective is often to build systems for resource-poor languages and domains, for evaluation purposes, annotated treebank test data figure prominently in this research (including in this paper).

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 50–61,
Edinburgh, Scotland, UK, July 27–31, 2011.

drift down phylogenetic trees (Berg-Kirkpatrick and Klein, 2010) is comparable, but the practical assumption of supervised helper languages is new to this work. Naseem et al. (2010) used universal syntactic categories and rules to improve grammar induction, but their model required expert handwritten rules as constraints.

Herein, we specifically focus on two problems in linguistic structure prediction: unsupervised POS tagging and unsupervised dependency grammar induction. Our experiments demonstrate that the presented method outperforms strong state-of-the-art unsupervised baselines for both tasks. Our approach can be applied to other problems in which a subset of the model parameters can be linked across languages. We also experiment with unsupervised learning of dependency structures from words, by combining our tagger and parser. Our results show that combining our tagger and parser with joint inference outperforms pipeline inference, and, in several cases, even outperforms models built using gold-standard part-of-speech tags.

2 Overview

For each language ℓ, we assume the presence of a set of fine-grained POS tags $\mathcal{F}_\ell$, used to annotate the language's treebank. Furthermore, we assume that there is a set of universal, coarse-grained POS tags $\mathcal{C}$ such that, for every language ℓ, there is a deterministic mapping from fine-grained to coarse-grained tags, $\lambda_\ell : \mathcal{F}_\ell \rightarrow \mathcal{C}$. Our approach can be summarized using the following steps for a given task:

1. Select a set of L helper languages for which there exists annotated data $\langle \mathcal{D}_1, \ldots, \mathcal{D}_L \rangle$. Here, we use treebanks in these languages.
2. For all $\ell \in \{1, \ldots, L\}$, convert the examples in $\mathcal{D}_\ell$ by applying λ_ℓ to every POS tag in the data, resulting in $\tilde{\mathcal{D}}_\ell$. Estimate the parameters of a probabilistic model using $\tilde{\mathcal{D}}_\ell$. In this work, such models are generative probabilistic models based on multinomial distributions,[2] including an HMM and the dependency model with valence (DMV) of Klein and Manning (2004). Denote the subset of parameters that are unlexicalized by $\boldsymbol{\theta}^{(\ell)}$. (Lexicalized parameters will be denoted $\boldsymbol{\eta}^{(\ell)}$.)
3. For the target language, define the set of valid unlexicalized parameters

$$\Theta = \left\{ \boldsymbol{\theta} \;\middle|\; \boldsymbol{\theta}_k = \sum_{\ell=1}^{L} \beta_{\ell,k} \boldsymbol{\theta}_k^{(\ell)}, \sum_{\ell=1}^{L} \beta_{\ell,k} = 1, \boldsymbol{\beta} \geq \mathbf{0} \right\}, \tag{1}$$

   for each group of parameters k, and maximize likelihood over that set, using the target-language unannotated data $\mathcal{U}$. Because the syntactic categories referenced by each $\boldsymbol{\theta}^{(\ell)}$ and all models in Θ are in $\mathcal{C}$, the models will be in the same parametric family. (Figure 1 gives a graphical interpretation of Θ.) Let the resulting model be $\boldsymbol{\theta}$.
4. Transform $\boldsymbol{\theta}$ by expanding the coarse-grained syntactic categories into the target language's fine-grained categories. Use the resulting model to initialize parameter estimation, this time over fine-grained tags, again using the unannotated target-language data $\mathcal{U}$. Initialize lexicalized parameters $\boldsymbol{\eta}$ for the target language using standard methods (e.g., uniform initialization with random symmetry breaking).

The main idea in the approach is to estimate a certain model family for one language, while using supervised models from other languages. The link between the languages is achieved through coarse-grained categories, which are now now commonplace (and arguably central to any theory of natural language syntax). A key novel contribution is the use of helper languages for initialization, and of unsupervised learning to learn the contribution of each helper language to that initialization (step 3). Additional treatment is required in expanding the coarse-grained model to the fine-grained one (step 4).

3 Interpolated Multilingual Probabilistic Context-Free Grammars

Our focus in this paper is on models that consist of multinomial distributions that have relationships between them through a generative process such as a probabilistic context-free grammar (PCFG). More specifically, we assume that we have a model defining a probability distribution over observed surface forms $\boldsymbol{x}$ and derivations $\boldsymbol{y}$ parametrized by $\boldsymbol{\theta}$:

[2] In §4 we also consider a feature-based parametrization.

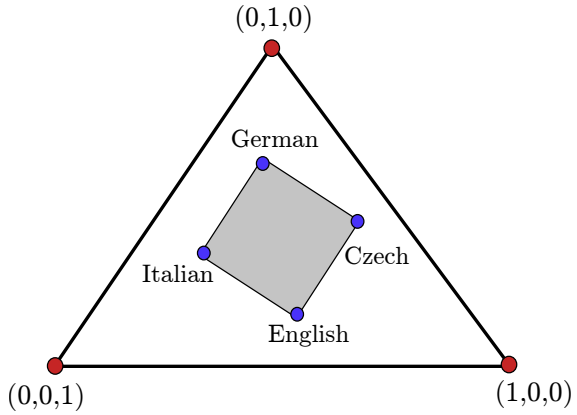

Figure 1: A simple case of interpolation within the 3-event probability simplex. The shaded area corresponds to a convex hull inside the probability simplex, indicating a mixture of the parameters of the four languages shown in the figure.

$$p(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{\theta}) = \prod_{k=1}^{K} \prod_{i=1}^{N_k} \theta_{k,i}^{f_{k,i}(\boldsymbol{x},\boldsymbol{y})} \quad (2)$$

$$= \exp \sum_{k=1}^{K} \sum_{i=1}^{N_k} f_{k,i}(\boldsymbol{x}, \boldsymbol{y}) \log \theta_{k,i} \quad (3)$$

where $f_{k,i}$ is a function that "counts" the number of times the kth distribution's ith event occurs in the derivation. The parameters $\boldsymbol{\theta}$ are a collection of K multinomials $\langle \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K \rangle$, the kth of which includes N_k events. Letting $\boldsymbol{\theta}_k = \langle \theta_{k,1}, \ldots, \theta_{k,N_k} \rangle$, each $\theta_{k,i}$ is a probability, such that $\forall k, \forall i, \theta_{k,i} \geq 0$ and $\forall k, \sum_{i=1}^{N_k} \theta_{k,i} = 1$.

3.1 Multilingual Interpolation

Our framework places additional, temporary constraints on the parameters $\boldsymbol{\theta}$. More specifically, we assume that we have L existing, parameter estimates for the multinomial families from Eq. 3. Each such estimate $\boldsymbol{\theta}^{(\ell)}$, for $1 \leq \ell \leq L$, corresponds to a the maximum likelihood estimate based on annotated data for the ℓth helper language. Then, to create a model for new language, we define a new set of parameters $\boldsymbol{\theta}$ as:

$$\theta_{k,i} = \sum_{\ell=1}^{L} \beta_{\ell,k} \theta_{k,i}^{(\ell)}, \quad (4)$$

where $\boldsymbol{\beta}$ is the set of coefficients that we will now be interested in estimating (instead of directly estimating $\boldsymbol{\theta}$). Note that for each k, $\sum_{\ell=1}^{L} \beta_{\ell,k} = 1$ and $\beta_{\ell,k} \geq 0$.

3.2 Grammatical Interpretation

We now give an interpretation of our approach relating it to PCFGs. We assume familiarity with PCFGs. For a PCFG $\langle \boldsymbol{G}, \boldsymbol{\theta} \rangle$ we denote the set of nonterminal symbols by $\mathcal{N}$, the set of terminal symbols by Σ, and the set of rewrite rules for each nonterminal $A \in \mathcal{N}$ by $R(A)$. Each $r \in R(A)$ has the form $A \rightarrow \alpha$ where $\alpha \in (\mathcal{N} \cup \Sigma)^*$. In addition, there is a probability attached to each rule $\theta_{A \rightarrow \alpha}$ such that $\forall A \in \mathcal{N}, \sum_{\alpha:(A \rightarrow \alpha) \in R(A)} \theta_{A \rightarrow \alpha} = 1$. A PCFG can be framed as a model using Eq. 3, where $\boldsymbol{\theta}$ correspond to $K = |\mathcal{N}|$ multinomial distributions, where each distribution attaches probabilities to rules with a specific left hand symbol.

We assume that the model we are trying to estimate (over coarse part-of-speech tags) can be framed as a PCFG $\langle \boldsymbol{G}, \boldsymbol{\theta} \rangle$. This is indeed the case for part-of-speech tagging and dependency grammar induction we experiment with in §6. In that case, our approach can be framed for PCFGs as following. We assume that there exists L set of parameters for this PCFG $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(L)}$, each corresponding to a helper language. We then create a new PCFG $\boldsymbol{G}'$ with parameters $\boldsymbol{\theta}'$ and $\boldsymbol{\beta}$ as follows:

1. $\boldsymbol{G}'$ contains all nonterminal and terminal symbols in $\boldsymbol{G}$, and none of the rules in $\boldsymbol{G}$.
2. For each nonterminal A in $\boldsymbol{G}$, we create a new nonterminal $a_{A,\ell}$ for $\ell \in \{1, \ldots, L\}$.
3. For each nonterminal A in $\boldsymbol{G}$, we create rules $A \rightarrow a_{A,\ell}$ for $\ell \in \{1, \ldots, L\}$ which have probabilities $\beta_{A \rightarrow a_{A,\ell}}$.
4. For each rule $A \rightarrow \alpha$ in $\boldsymbol{G}$, we add to $\boldsymbol{G}'$ the rule $a_{A,\ell} \rightarrow \alpha$ with

$$\theta'_{a_{A,\ell} \rightarrow \alpha} = \theta^{(\ell)}_{A \rightarrow \alpha}. \quad (5)$$

where $\theta^{(\ell)}_{A \rightarrow \alpha}$ is the probability associated with rule $A \rightarrow \alpha$ in the ℓth helper language.

At each point, the derivational process of this PCFG uses the nonterminal's specific $\boldsymbol{\beta}$ coefficients

to choose one of the helper languages. It then selects a rule according to the multinomial from that language. This step is repeated until a whole derivation is generated.

This PCFG representation of the approach in §3 points to a possible generalization. Instead of using an identical CFG backbone for each language, we can use a set of PCFGs, $\langle \boldsymbol{G}^{(\ell)}, \boldsymbol{\theta}^{(\ell)} \rangle$ with an identical nonterminal set and alphabet, and repeat the same construction as above, replacing step 4 with the addition of rules of the form $a_{A,\ell} \rightarrow \alpha$ for each rule $A \rightarrow \alpha$ in $\boldsymbol{G}^{(\ell)}$. Such a construction allows more syntactic variability in the language we are trying to estimate, originating in the syntax of the various helper languages. In this paper, we do not use this generalization, and always use the same PCFG backbone for all languages.

Note that the interpolated model can still be understood in terms of the exponential model of Eq. 3. For a given collection of multinomials and base models of the form of Eq. 3, we can analogously define a new log-linear model over a set of extended derivations. These derivations will now include $L \times K$ features of the form $g_{\ell,k}(\boldsymbol{x}, \boldsymbol{y})$, corresponding to a count of the event of choosing the ℓth mixture component for multinomial k. In addition, the feature set $f_{k,i}(\boldsymbol{x}, \boldsymbol{y})$ will be extended to a feature set of the form $f_{\ell,k,i}(\boldsymbol{x}, \boldsymbol{y})$, analogous to step 4 in constructed PCFG above. The model parameterized according to Eq. 4 can be recovered by marginalizing out the "g" features. We will refer to the model with these new set of features as "the extended model."

4 Inference and Parameter Estimation

The main building block commonly required for unsupervised learning in NLP is that of computing feature expectations for a given model. These feature expectations can be used with an algorithm such as expectation-maximization (where the expectations are normalized to obtain a new set of multinomial weights) or with other gradient based log-likelihood optimization algorithms such as L-BFGS (Liu and Nocedal, 1989) for feature-rich models.

Estimating Multinomial Distributions Given a surface form $\boldsymbol{x}$, a multinomial k and an event i in the multinomial, "feature expectation" refers to the calculation of the following quantities (in the extended model):

$$\mathbb{E}[f_{\ell,k,i}(\boldsymbol{x}, \boldsymbol{y})] = \sum_{\boldsymbol{y}} p(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{\theta}) f_{\ell,k,i}(\boldsymbol{x}, \boldsymbol{y}) \quad (6)$$

$$\mathbb{E}[g_{\ell,k}(\boldsymbol{x}, \boldsymbol{y})] = \sum_{\boldsymbol{y}} p(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{\theta}) g_{\ell,k}(\boldsymbol{x}, \boldsymbol{y}) \quad (7)$$

These feature expectations can usually be computed using algorithms such as the forward-backward algorithm for hidden Markov models, or more generally, the inside-outside algorithm for PCFGs. In this paper, however, the task of estimation is different than the traditional task. As mentioned in §2, we are interested in estimating $\boldsymbol{\beta}$ from Eq. 4, while fixing $\boldsymbol{\theta}^{(\ell)}$. Therefore, we are only interested in computing expectations of the form of Eq. 7.

As explained in §3.2, any model interpolating with the $\boldsymbol{\beta}$ parameters can be reduced to a new log-linear model with additional features representing the mixture coefficients of $\boldsymbol{\beta}$. We can then use the inside-outside algorithm to obtain the necessary feature expectations for features of the form $g_{\ell,k}(\boldsymbol{x}, \boldsymbol{y})$, expectations which assist in the estimation of the $\boldsymbol{\beta}$ parameters.

These feature expectations can readily be used in estimation algorithms such as expectation-maximization (EM). With EM, the update at iteration t would be:

$$\beta_{\ell,k}^{(t)} = \frac{\mathbb{E}[g_{\ell,k}(\boldsymbol{x}, \boldsymbol{y})]}{\sum_{\ell} \mathbb{E}[g_{\ell,k}(\boldsymbol{x}, \boldsymbol{y})]}, \quad (8)$$

where the expectations are taken with respect to $\boldsymbol{\beta}^{(t-1)}$ and the fixed $\boldsymbol{\theta}^{(l)}$ for $\ell = 1, \ldots, L$.

Estimating Feature-Rich Directed Models Recently Berg-Kirkpatrick et al. (2010) found that replacing traditional multinomial parameterizations with locally normalized, feature-based log-linear models was advantageous. This can be understood as parameterizing $\boldsymbol{\theta}$:

$$\theta_{k,i} = \frac{\exp \boldsymbol{\psi}^{\top} \mathbf{h}(k, i)}{\sum_{i'} \exp \boldsymbol{\psi}^{\top} \mathbf{h}(k, i')} \quad (9)$$

where $\mathbf{h}(k, i)$ are a set of features looking at event i in context k. For such a feature-rich model, our multilingual modeling framework still substitutes $\boldsymbol{\theta}$ with a mixture of supervised multinomials for L helper languages as in Eq. 4. However, for computational

convenience, we also reparametrize the mixture coefficients $\boldsymbol{\beta}$:

$$\beta_{\ell,k} = \frac{\exp \gamma_{\ell,k}}{\sum_{\ell'=1}^{L} \exp \gamma_{\ell',k}} \quad (10)$$

Here, each $\gamma_{\ell,k}$ is an unconstrained parameter, and the above "softmax" transformation ensures that $\boldsymbol{\beta}$ lies within the probability simplex for context k. This is done so that a gradient-based optimization method like L-BFGS (Liu and Nocedal, 1989) can be used to estimate $\boldsymbol{\gamma}$ without having to worry about additional simplex constraints. For optimization, derivatives of the data log-likelihood with respect to $\boldsymbol{\gamma}$ need to be computed. We calculate the derivatives following Berg-Kirkpatrick et al. (2010, §3.1), making use of feature expectations, calculated exactly as before.

In addition to these estimation techniques, which are based on the optimization of the log-likelihood, we also consider a trivially simple technique for estimating $\boldsymbol{\beta}$: setting $\beta_{l,k}$ to the uniform weight L^{-1}, where L is the number of helper languages.

5 Coarse-to-Fine Multinomial Expansion

To expand these multinomials involving coarse-grained categories into multinomials over fine-grained categories specific to the target language t, we do the following:

- Whenever a multinomial *conditions* on a coarse category $c \in \mathcal{C}$, we make copies of it for each fine-grained category in $\lambda_t^{-1}(c) \subset \mathcal{F}_t$.[3] If the multinomial does not condition on coarse categories, it is simply copied.
- Whenever a probability θ_i within a multinomial distribution involves a coarse-grained category c as an event (i.e., it is on the left side of the conditional bar), we expand the event into $|\lambda_t^{-1}(c)|$ new events, one per corresponding fine-grained category, each assigned the value $\frac{\theta_i}{|\lambda_t^{-1}(c)|}$.[4]

[3] We note that in the models we experiment with, we always condition on at most one fine-grained category.

[4] During this expansion process for a coarse event, we tried adding random noise to $\frac{\theta_i}{|\lambda_t^{-1}(c)|}$ and renormalizing, to break symmetry between the fine events, but that was found to be harmful in preliminary experiments.

The result of this expansion is a model in the desired family; we use it to initialize conventional unsupervised parameter estimation. Lexical parameters, if any, do not undergo this expansion process, and they are estimated anew in the fine grained model during unsupervised learning, and are initialized using standard methods.

6 Experiments and Results

In this section, we describe the experiments undertaken and the results achieved. We first note the characteristics of the datasets and the universal POS tags used in multilingual modeling.

6.1 Data

For our experiments, we fixed a set of four helper languages with relatively large amounts of data, displaying nontrivial linguistic diversity: Czech (Slavic), English (West-Germanic), German (West-Germanic), and Italian (Romance). The datasets are the CoNLL-X shared task data for Czech and German (Buchholz and Marsi, 2006),[5] the Penn Treebank for English (Marcus et al., 1993), and the CoNLL 2007 shared task data for Italian (Montemagni et al., 2003). This was the only set of helper languages we tested; improvements are likely possible. We leave an exploration of helper language choice (a subset selection problem) to future research, instead demonstrating that the concept has merit.

We considered ten target languages: Bulgarian (Bg), Danish (Da), Dutch (Nl), Greek (El), Japanese (Jp), Portuguese (Pt), Slovene (Sl), Spanish (Es), Swedish (Sv), and Turkish (Tr). The data come from the CoNLL-X and CoNLL 2007 shared tasks (Buchholz and Marsi, 2006; Nivre et al., 2007). For all the experiments conducted, we trained models on the training section of a language's treebank and tested on the test set. Table 1 shows the number of sentences in the treebanks and the size of fine POS tagsets for each language.

Following standard practice, in unsupervised grammar induction experiments we remove punctuation and then eliminate sentences from the data of length greater than 10.

[5] These are based on the Prague Dependency Treebank (Hajič, 1998) and the Tiger treebank (Brants et al., 2002) respectively.

	Pt	Tr	Bg	Jp	El	Sv	Es	Sl	Nl	Da
Training sentences	9,071	4,997	12,823	17,044	2,705	11,042	3,306	1,534	13,349	5,190
Test sentences	288	623	398	709	197	389	206	402	386	322
Size of POS tagset	22	31	54	80	38	41	47	29	12	25

Table 1: The first two rows show the sizes of the training and test datasets for each language. The third row shows the number of fine POS tags in each language including punctuations.

6.2 Universal POS Tags

Our coarse-grained, universal POS tag set consists of the following 12 tags: NOUN, VERB, ADJ (adjective), ADV (adverb), PRON (pronoun), DET (determiner), ADP (preposition or postposition), NUM (numeral), CONJ (conjunction), PRT (particle), PUNC (punctuation mark) and X (a catch-all for other categories such as abbreviations or foreign words). These follow recent work by Das and Petrov (2011) on unsupervised POS tagging in a multilingual setting with parallel data, and have been described in detail by Petrov et al. (2011).

While there might be some controversy about what an appropriate universal tag set should include, these 12 categories (or a subset) cover the most frequent parts of speech and exist in one form or another in all of the languages that we studied. For each language in our data, a mapping from the fine-grained treebank POS tags to these universal POS tags was constructed manually by Petrov et al. (2011).

6.3 Part-of-Speech Tagging

Our first experimental task is POS tagging, and here we describe the specific details of the model, training and inference and the results attained.

6.3.1 Model

The model is a hidden Markov model (HMM), which has been popular for unsupervised tagging tasks (Merialdo, 1994; Elworthy, 1994; Smith and Eisner, 2005; Berg-Kirkpatrick et al., 2010).[6] We use a bigram model and a locally normalized log-linear parameterization, like Berg-Kirkpatrick et al. (2010). These locally normalized log-linear models can look at various aspects of the observation x given a tag y, or the pair of tags in a transition, incorporating overlapping features. In basic monolingual experiments, we used the same set of features as Berg-Kirkpatrick et al. (2010). For the transition log-linear model, Berg-Kirkpatrick et al. (2010) used only a single indicator feature of a tag pair, essentially equating to a traditional multinomial distribution. For the emission log-linear model, several features were used: an indicator feature conjoining the state y and the word x, a feature checking whether x contains a digit conjoined with the state y, another feature indicating whether x contains a hyphen conjoined with y, whether the first letter of x is upper case along with the state y, and finally indicator features corresponding to suffixes up to length 3 present in x conjoined with the state y.

Since only the unlexicalized transition distributions are common across multiple languages, assuming that they all use a set of universal POS tags, akin to Eq. 4, we can have a multilingual version of the transition distributions, by incorporating supervised helper transition probabilities. Thus, we can write:

$$\theta_{y \to y'} = \sum_{\ell=1}^{L} \beta_{\ell,y} \theta^{(\ell)}_{y \to y'} \tag{11}$$

We use the above expression to replace the transition distributions, obtaining a multilingual mixture version of the model. Here, the transition probabilities $\theta^{(\ell)}_{y \to y'}$ for the ℓth helper language are fixed after being estimated using maximum likelihood estimation on the helper language's treebank.

6.3.2 Training and Inference

We trained both the basic feature-based HMM model as well as the multilingual mixture model by optimizing the following objective function:[7]

$$\mathcal{L}(\boldsymbol{\psi}) \quad = \sum_{i=1}^{N} \log \sum_{\boldsymbol{y}} p(\boldsymbol{x}^{(i)}, \boldsymbol{y} \mid \boldsymbol{\psi}) - C\|\boldsymbol{\psi}\|_2^2$$

[6] HMMs can be understood as a special case of PCFGs.

[7] Note that in the objective function, for brevity, we abuse notation by using $\boldsymbol{\psi}$ for both models – monolingual and multilingual; the latter model is also parameterized by $\boldsymbol{\gamma}$.

Method	Pt	Tr	Bg	Jp	El	Sv	Es	Sl	Nl	Da	Avg
Uniform+DG	45.7	**43.6**	**38.0**	60.4	36.7	37.7	31.8	35.9	43.7	36.2	41.0
Mixture+DG	51.5	38.6	35.8	**61.7**	**38.9**	**39.9**	**40.5**	**36.0**	**50.2**	39.9	**43.3**
DG (B-K et al., 2010)	**53.5**	27.9	34.7	52.3	35.3	34.4	40.0	33.4	45.4	**48.8**	40.6

(a)

Method	Pt	Tr	Bg	Jp	El	Sv	Es	Sl	Nl	Da	Avg
Uniform+DG	83.8	**50.4**	81.3	77.9	80.3	**69.0**	82.3	**82.8**	79.3	82.0	76.9
Mixture+DG	**84.7**	50.0	**82.6**	79.9	80.3	67.0	**83.3**	**82.8**	**80.0**	82.0	**77.3**
DG (B-K et al., 2010)	75.4	**50.4**	80.7	**83.4**	**88.0**	61.5	82.3	75.6	79.2	**82.3**	75.9

(b)

Table 2: Results for unsupervised POS induction (a) without a tagging dictionary and (b) with a tag dictionary constructed from the training section of the corresponding treebank. DG (at the bottom) stands for the direct gradient method of Berg-Kirkpatrick et al. (2010) using a monolingual feature-based HMM. "Mixture+DG" is the model where multilingual mixture coefficients $\boldsymbol{\beta}$ of helper languages are estimated using coarse tags (§4), followed by expansion (§5), and then initializing DG with the expanded transition parameters. "Uniform+DG" is the case where $\boldsymbol{\beta}$ are set to 1/4, transitions of helper languages are mixed, expanded, and then DG is initialized with the result. For (a), evaluation is performed using one-to-one mapping accuracy. In case of (b), the tag dictionary solves the problem of tag identification and performance is measured using per word POS accuracy. "Avg" denotes macro-average across the ten languages.

Note that this involves marginalizing out all possible state configurations $\boldsymbol{y}$ for a sentence $\boldsymbol{x}$, resulting in a non-convex objective. As described in §4, we optimized this function using L-BFGS. For the monolingual model, derivatives of the feature weights took the exact same form as Berg-Kirkpatrick et al. (2010), while for the mixture case, we computed gradients with respect to $\boldsymbol{\gamma}$, the unconstrained parameters used to express the mixture coefficients $\boldsymbol{\beta}$ (see Eq. 10). The regularization constant C was set to 1.0 for all experiments, and L-BFGS was run till convergence.

During training, for the basic monolingual feature-based HMM model, we initialized all parameters using small random real values, sampled from $\mathcal{N}(0, 0.01)$. For estimation of the mixture parameters $\boldsymbol{\gamma}$ for our multilingual model (step 3 in §2), we similarly sampled real values from $\mathcal{N}(0, 0.01)$ as an initialization point. Moreover, during this stage, the emission parameters also go through parameter estimation, but they are *monolingual*, and are initialized with real values sampled from $\mathcal{N}(0, 0.01)$; as explained in §2, coarse universal tags are used both in the transitions and emissions during multilingual estimation.

After the mixture parameters $\boldsymbol{\gamma}$ are estimated, we compute the mixture probabilities $\boldsymbol{\beta}$ using Eq. 10. Next, for each tag pair y, y', we compute $\theta_{y \to y'}$, which are the coarse transition probabilities interpolated using $\boldsymbol{\beta}$, given the helper languages. We then expand these transition probabilities (see §5) to result in transition probabilities based on fine tags. Finally, we train a feature-HMM by initializing its transition parameters with natural logarithms of the expanded $\boldsymbol{\theta}$ parameters, and the emission parameters using small random real values sampled from $\mathcal{N}(0, 0.01)$. This implies that the lexicalized emission parameters $\boldsymbol{\eta}$ that were previously estimated in the coarse multilingual model are thrown away and not used for initialization; instead standard initialization is used.

For inference at the testing stage, we use minimum Bayes-risk decoding (or "posterior decoding"), by choosing the most probable tag for each word position, given the entire observation $\boldsymbol{x}$. We chose this strategy because it usually performs slightly better than Viterbi decoding (Cohen and Smith, 2009; Ganchev et al., 2010).

6.3.3 Experimental Setup

For experiments, we considered three configurations, and for each, we implemented two variants of POS induction, one without any kind of supervision, and the other with a tag dictionary. Our baseline is

the direct gradient approach of Berg-Kirkpatrick et al. (2010), which is the current state of the art for this task, outperforming classical HMMs. Because this model achieves strong performance using straightforward MLE, it also serves as the core model within our approach. This model has also been applied in a multilingual setting with parallel data (Das and Petrov, 2011). In this baseline, we set the number of HMM states to the number of fine-grained treebank tags for the given language.

We test two versions of our model. The first initializes training of the target language's POS model using a uniform mixture of the helper language models (i.e., each $\beta_{\ell,y} = \frac{1}{L} = \frac{1}{4}$), and expansion from coarse-grained to fine-grained POS tags as described in §5. We call this model "Uniform+DG."

The second version estimates the mixture coefficients to maximize likelihood, then expands the POS tags (§5), using the result to initialize training of the final model. We call this model "Mixture+DG."

No Tag Dictionary For each of the above configurations, we ran purely unsupervised training without a tag dictionary, and evaluated using *one-to-one mapping* accuracy constraining at most one HMM state to map to a unique treebank tag in the test data, using maximum bipartite matching. This is a variant of the greedy one-to-one mapping scheme of Haghighi and Klein (2006).[8]

With a Tag Dictionary We also ran a second version of each experimental configuration, where we used a tag dictionary to restrict the possible path sequences of the HMM during both learning and inference. This tag dictionary was constructed only from the training section of a given language's treebank. It is widely known that such knowledge improves the quality of the model, though it is an open debate whether such knowledge is realistic to assume. For this experiment we removed punctuation from the training and test data, enabling direct use within the dependency grammar induction experiments.

[8]We also evaluated our approach using the greedy version of this evaluation metric, and results followed the same trends with only minor differences. We did not choose the other variant, *many-to-one mapping* accuracy, because quite often the metric mapped several HMM states to one treebank tag, leaving many treebank tags unaccounted for.

6.3.4 Results

All results for POS induction are shown in Table 2. Without a tag dictionary, in eight out of ten cases, either Uniform+DG or Mixture+DG outperforms the monolingual baseline (Table 2a). For six of these eight languages, the latter model where the mixture coefficients are learned automatically fares better than uniform weighting. *With* a tag dictionary, the multilingual variants outperform the baseline in seven out of ten cases, and the learned mixture outperforms or matches the uniform mixture in five of those seven (Table 2b).

6.4 Dependency Grammar Induction

We next describe experiments for dependency grammar induction. As the basic grammatical model, we adopt the dependency model with valence (Klein and Manning, 2004), which forms the basis for state-of-the-art results for dependency grammar induction in various settings (Cohen and Smith, 2009; Spitkovsky et al., 2010; Gillenwater et al., 2010; Berg-Kirkpatrick and Klein, 2010). As shown in Table 3, DMV obtains much higher accuracy in the supervised setting than the unsupervised setting, suggesting that more can be achieved with this model family.[9] For this reason, and because DMV is easily interpreted as a PCFG, it is our starting point and baseline.

We consider four conditions. The independent variables are (1) whether we use uniform $\boldsymbol{\beta}$ (all set to $\frac{1}{4}$) or estimate them using EM (as described in §4), and (2) whether we simply use the mixture model to decode the test data, or to initialize EM for the DMV. The four settings are denoted "Uniform," "Mixture," "Uniform+EM," and "Mixture+EM."

The results are given in Table 3. In general, the use of data from other languages improves performance considerably; all of our methods outperform the Klein and Manning (2004) initializer, and we achieve state-of-the-art performance for eight out of ten languages. Uniform and Mixture behave similarly, with a slight advantage to the trained mixture setting. Using EM to train the mixture coefficients more often hurts than helps (six languages out of ten). It is well known that likelihood does not cor-

[9]Its supervised performance is still far from the supervised state of the art in dependency parsing.

Method	Pt	Tr	Bg	Jp	El	Sv	Es	Sl	Nl	Da	Avg
Uniform	78.6	45.0	**75.6**	56.3	57.0	**74.0**	73.2	46.1	**50.7**	59.2	61.6
Mixture	76.8	45.3	75.5	58.3	59.5	73.2	75.9	46.0	51.1	**59.9**	**62.2**
Uniform+EM	78.7	43.9	74.7	59.8	**73.0**	70.5	75.5	41.3	45.9	51.3	61.5
Mixture+EM	**79.8**	44.1	72.8	**63.9**	72.3	68.7	**76.7**	41.0	46.0	55.2	62.1
EM (K & M, 2004)	42.5	36.3	54.3	43.0	41.0	42.3	38.1	37.0	38.6	41.4	41.4
PR (G et al., '10)	47.8	**53.4**	54.0	60.2	-	42.2	62.4	**50.3**	37.9	44.0	-
Phylo. (B-K & K, '10)	63.1	-	-	-	-	58.3	63.8	49.6	45.1	41.6	-
Supervised (MLE)	*81.7*	*75.7*	*83.0*	*89.2*	*81.8*	*83.2*	*79.0*	*74.5*	*64.8*	*80.8*	*79.3*

Table 3: Results for dependency grammar induction given gold-standard POS tags, reported as attachment accuracy (fraction of parents which are correct). The three existing methods are: our replication of EM with the initializer from Klein and Manning (2004), denoted "EM"; reported results from Gillenwater et al. (2010) for posterior regularization ("PR"); and reported results from Berg-Kirkpatrick and Klein (2010), denoted "Phylo." "Supervised (MLE)" are oracle results of estimating parameters from gold-standard annotated data using maximum likelihood estimation. "Avg" denotes macro-average across the ten languages.

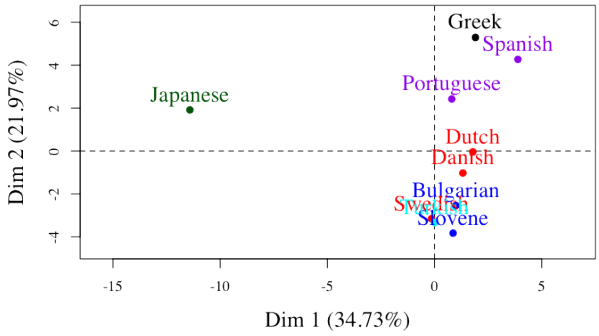

Figure 2: Projection of the learned mixture coefficients through PCA. In green, Japanese. In red, Dutch, Danish and Swedish. In blue, Bulgarian and Slovene. In magenta, Portuguese and Spanish. In black, Greek. In cyan, Turkish.

relate with the true accuracy measurement, and so it is unsurprising that this holds in the constrained mixture family as well. In future work, a different parametrization of the mixture coefficients, through features, or perhaps a Bayesian prior on the weights, might lead to an objective that better simulates accuracy.

Table 3 shows that even uniform mixture coefficients are sufficient to obtain accuracy which supercedes most unsupervised baselines. We were interested in testing whether the coefficients which are learned actually reflect similarities between the languages. To do that, we projected the learned vectors β for each tested language using principal component analysis and plotted the result in Figure 2. It is interesting to note that languages which are closer phylogenetically tend to appear closer to each other in the plot.

Our experiments also show that multilingual learning performs better for dependency grammar induction than part-of-speech tagging. We believe that this happens because of the nature of the models and data we use. The transition matrix in part-of-speech tagging largely depends on word order in the various helper languages, which differs greatly. This means that a mixture of transition matrices will not necessarily yield a meaningful transition matrix. However, for dependency grammar, there are certain universal dependencies which appear in all helper languages, and therefore, a mixture between multinomials for these dependencies still yields a useful multinomial.

6.5 Inducing Dependencies from Words

Finally, we combine the models for POS tagging and grammar induction to perform grammar induction directly from words, instead of gold-standard POS tags. Our approach is as follows:

1. *With* a tag dictionary, learn a fine-grained POS tagging model unsupervised, using either DG or Mixture+DG as described in §6.3 and shown in Table 2b.

Method	Tags	Pt	Tr	Bg	Jp	El	Sv	Es	Sl	Nl	Da	Avg
Joint	DG	**68.4**	**52.4**	62.4	61.4	63.5	**58.2**	67.7	**47.2**	48.3	**50.4**	**57.9**
Joint	Mixture+DG	62.2	47.4	**67.0**	**69.5**	52.2	49.1	**69.3**	36.8	**52.2**	50.1	55.6
Pipeline	DG	60.0	50.8	57.7	64.2	**68.2**	57.9	65.8	45.8	49.9	48.9	56.9
Pipeline	Mixture+DG	59.8	47.1	62.9	68.6	50.0	47.6	68.1	36.4	51.2	48.3	54.0
Gold-standard tags		*79.8*	*45.3*	*75.6*	*63.9*	*73.0*	*74.0*	*76.7*	*46.1*	*50.7*	*59.9*	*64.5*

Table 4: Results for dependency grammar induction over words. "Joint"/"Pipeline" refers to joint/pipeline decoding of tags and dependencies as described in the text. See §6.3 for a description of DG and Mixture+DG. For the induction of dependencies we use the Mixture+EM setting as described in §6.4. All tag induction uses a dictionary as specified in §6.3. The last row in this table indicates the best results using multilingual guidance taken from our methods in Table 3. "Avg" denotes macro-average across the ten languages.

2. Apply the fine-grained tagger to the words in the training data for the dependency parser. We consider two variants: the most probable assignment of tags to words (denoted "Pipeline"), and the posterior distribution over tags for each word, represented as a weighted "sausage" lattice (denoted "Joint"). This idea was explored for joint inference by Cohen and Smith (2007).
3. We apply the Mixture+EM unsupervised parser learning method from §6.4 to the automatically tagged sentences, or the lattices.
4. Given the two models, we infer POS tags on the test data using DG or Mixture+DG to get a lattice (Joint) or a sequence (Pipeline) and then parse using the model from the previous step.[10] The resulting dependency trees are evaluated against the gold standard.

Results are reported in Table 4. In almost all cases, joint decoding of tags and trees performs better than the pipeline. Even though our part-of-speech tagger with multilingual guidance outperforms the completely unsupervised baseline, there is not always an advantage of using this multilingually guided part-of-speech tagger for dependency grammar induction. For Turkish, Japanese, Slovene and Dutch, our unsupervised learner from words outperforms unsupervised parsing using gold-standard part-of-speech tags.

We note that some recent work gives a treatment to unsupervised parsing (but not of dependencies) directly from words (Seginer, 2007). Earlier work that induced part-of-speech tags and then performed unsupervised parsing in a pipeline includes Klein and Manning (2004) and Smith (2006). Headden et al. (2009) described the use of a lexicalized variant of the DMV model, with the use of gold part-of-speech tags.

[10]The decoding method on test data (Joint or Pipeline) was matched to the training method, though they are orthogonal in principle.

7 Conclusion

We presented an approach to exploiting annotated data in helper languages to infer part-of-speech tagging and dependency parsing models in a different, target language, without parallel data. Our approach performs well in many cases. We also described a way to do joint decoding of part-of-speech tags and dependencies which performs better than a pipeline. Future work might consider exploiting a larger number of treebanks, and more powerful techniques for combining models than simple local mixtures.

Acknowledgments

We thank Ryan McDonald and Slav Petrov for helpful comments on an early draft of the paper. This research has been funded by NSF grants IIS-0844507 and IIS-0915187 and by U.S. Army Research Office grant W911NF-10-1-0533.

References

T. Berg-Kirkpatrick and D. Klein. 2010. Phylogenetic grammar induction. In *Proceedings of ACL*.

T. Berg-Kirkpatrick, A. B. Côté, J. DeNero, and D. Klein. 2010. Painless unsupervised learning with features. In *Proceedings of NAACL-HLT*.

S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*.

S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL*.

D. Burkett and D. Klein. 2008. Two languages are better than one (for syntactic parsing). In *Proceedings of EMNLP*.

S. B. Cohen and N. A. Smith. 2007. Joint morphological and syntactic disambiguation. In *Proceedings of EMNLP-CoNLL*.

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

S. B. Cohen and N. A. Smith. 2010. Covariance in unsupervised learning of probabilistic grammars. *Journal of Machine Learning Research*, 11:3017–3051.

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

D. Elworthy. 1994. Does Baum-Welch re-estimation help taggers? In *Proceedings of ACL*.

K. Ganchev, J. Graça, J. Gillenwater, and B. Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11:2001–2049.

J. Gillenwater, K. Ganchev, J. Graça, F. Pereira, and B. Taskar. 2010. Sparsity in dependency grammar induction. In *Proceedings of ACL*.

A. Haghighi and D. Klein. 2006. Prototype driven learning for sequence models. In *Proceedings of HLT-NAACL*.

J. Hajič. 1998. Building a syntactically annotated corpus: The Prague Dependency Treebank. In *Issues of Valency and Meaning. Studies in Honor of Jarmila Panevová*. Prague Karolinum, Charles University Press.

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

M. Johnson, T. L. Griffiths, and S. Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proceedings of NAACL*.

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

D. C. Liu and J. Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Math. Programming*, 45:503–528.

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

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

B. Merialdo. 1994. Tagging English text with a probabilistic model. *Compulational Lingustics*, 20(2):155–72.

S. Montemagni, F. Barsotti, M. Battista, N. Calzolari, O. Corazzari, A. Zampolli, F. Fanciulli, M. Massetani, R. Raffaelli, R. Basili, M. T. Pazienza, D. Saracino, F. Zanzotto, N. Mana, F. Pianesi, and R. Delmonte. 2003. Building the Italian Syntactic-Semantic Treebank. In *Building and using Parsed Corpora*, Language and Speech Series. Kluwer, Dordrecht.

T. Naseem, B. Snyder, J. Eisenstein, and R. Barzilay. 2009. Multilingual part-of-speech tagging: Two unsupervised approaches. *JAIR*, 36.

T. Naseem, H. Chen, R. Barzilay, and M. Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *Proceedings of EMNLP*.

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

S. Petrov, D. Das, and R. McDonald. 2011. A universal part-of-speech tagset. *ArXiv:1104.2086*.

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

N. A. Smith and J. Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of ACL*.

D. A. Smith and J. Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of EMNLP*.

D. A. Smith and N. A. Smith. 2004. Bilingual parsing with factored estimation: Using English to parse Korean. In *Proceedings of EMNLP*.

N. A. Smith. 2006. *Novel Estimation Methods for Unsupervised Discovery of Latent Structure in Natural Language Text*. Ph.D. thesis, Johns Hopkins University.

B. Snyder and R. Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL*.

B. Snyder, T. Naseem, and R. Barzilay. 2009. Unsupervised multilingual grammar induction. In *Proceedings of ACL-IJCNLP*.

V. Spitkovsky, H. Alshawi, and D. Jurafsky. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Proceedings of NAACL*.

C. Xi and R. Hwa. 2005. A backoff model for bootstrapping resources for non-English languages. In *Proceedings of HLT-EMNLP*.

D. Yarowsky and G. Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of NAACL*.