Unsupervised Learning of PCFGs with Normalizing Flow

Anonymous ACL submission

Abstract

Unsupervised PCFG inducers hypothesize sets of compact context-free rules as explanations for sentences. These models not only provide tools for low-resource languages, but also play an important role in modeling language acquisition (Bannard et al., 2009; Abend et al., 2017). However, current PCFG induction models, using word tokens as input, are unable to incorporate semantics and morphology into induction, and may encounter issues of sparse vocabulary when facing morphologically rich languages. This paper describes a neural PCFG inducer which employs context embeddings (Peters et al., 2018) in a normalizing flow model (Dinh et al., 2015) to extend PCFG induction to use semantic and morphological information. Linguistically motivated sparsity and categorical distance constraints are imposed on the inducer as regularization. Experiments show that the PCFG induction model with normalizing flow produces grammars with state-of-the-art accuracy on a variety of different languages. Ablation further shows a positive effect of normalizing flow, context embeddings and proposed regularizers.

1 Introduction

Unsupervised PCFG inducers (Jin et al., 2018b) automatically bracket sentences into nested spans, and label these spans with consistent, linguistically relevant syntactic categories, which may be useful in downstream applications or linguistic research on under-resourced languages. Their success also provides evidence for learnability of grammar in absence of strong linguistic universals (MacWhinney and Bates, 1993; Plunkett and Wood, 2004; Bannard et al., 2009). However, current PCFG induction models, using word tokens as input, are unable to incorporate semantics and morphology into induction, and may encounter issues of sparse vocabulary when facing morphologically rich languages.

This paper describes a PCFG induction model which exploits recent advances in deep generative models and context embeddings to generalize over rare, morphologically rich forms. We contextualize a PCFG's terminal emission rules with context embeddings (Peters et al., 2018) as observations, in order to bring context and subword information into the model. Probabilities for these contextualized terminal emission rules are modeled by transforming distributions with normalizing flow (Rezende and Mohamed, 2015; Dinh et al., 2015; He et al., 2018). Through invertible transformations, flow models transform simple distributions (e.g. Gaussian) into complex and potentially multi-modal distributions over observation vectors. These improvements help increase the expressivity of the induction model and give the model the ability to generalize over rare words, but still preserve the tractability of marginal likelihood computation so that inference is possible with marginal likelihood maximization.

Experiments described in this paper show that the model is able to achieve state-of-the-art or competitive results on multiple languages compared with existing PCFG induction and unlabeled tree induction models, especially on languages where complex morphology may cause induction models with discrete observations to succumb to data sparsity. Further analyses show (1) that the flow-based inducer is able to use morphological and semantic information in embeddings for grammar induction, (2) that the model produces consistent and meaningful labels at phrasal and lexical levels, and (3) that both the normalizing flow and the linguistically-motivated regularization terms make substantial improvements to parsing accuracy.

2 PCFGs with vector terminals

We first consider factoring the Chomsky normal form PCFG with C non-terminal categories into two separate parts: binary-branching non-terminal expansion rule[1] probabilities, and unary-branching terminal emission rule probabilities. We first define a set of Bernoulli distributions that distribute probability mass between these two sets of rules:

$$\mathsf{P}(\text{Term} = 1 \mid c) = \frac{1}{1 + \exp(-\delta_c^\top \mathbf{d})}, \quad (1)$$

where c is a non-terminal category, δ_c is a Kronecker delta function – a vector with value one at index c and zeros everywhere else – and $\delta_c^\top \mathbf{d}$ is a parameter for the Bernoulli distribution of c. $\mathbf{d}$ has C dimensions.

Binary-branching non-terminal expansion rule probabilities for a non-terminal category c are defined as:

$$\mathsf{P}(c \to a\ b) = \mathsf{P}(\text{Term} = 0 \mid c) \cdot \frac{\exp(\delta_c^\top \mathbf{N})(\delta_a \otimes \delta_b)}{\exp(\delta_c^\top \mathbf{N})\mathbf{1}} \quad (2)$$

where $\otimes$ is a Kronecker product, a is the category of the left child, b is the category of the right child, and $\delta_c^\top \mathbf{N}$ is a parameter vector for the multinomial distribution of the category c with $\mathbf{N} \in \mathbb{R}^{C \times C^2}$.

The contextualized unary-branching terminal emission rule probabilities for a preterminal category c are defined as:

$$\mathsf{P}(c \to \mathbf{x}_w) = \mathsf{P}(\text{Term} = 1 \mid c) \cdot f_c(\mathbf{x}_w; \delta_c^\top \mathbf{L}) \quad (3)$$

where w is an observed word token, $\mathbf{x}_w$ is the vectorial representation of that word with D dimensions, f_c is a probability density or mass function, and $\delta_c^\top \mathbf{L}$ is a parameter vector for the probability function of the category c. We can recover the multinomial PCFG formulation by setting $\mathbf{x}_w$ to be a one-hot word representation and the probability function f_c to be a multinomial distribution parameterized by $\delta_c^\top \mathbf{L}$. We can also set $\mathbf{x}_w$ to be a word embedding and f_c to be Gaussian distributions parameterized by $\delta_c^\top \mathbf{L}$, giving us a PCFG with Gaussian emission.

In order to incorporate more information into the induction model, context embeddings (Peters et al., 2018) can be used here for $\mathbf{x}_w$. The ELMo model combines learned word embeddings with character embeddings through CNN encoders, and composes contextualized embeddings with bidirectional LSTMs over the combined representations. The output from the BiLSTM contains both subword information, word information and context information and is used as contextualized embeddings for words. While simple D-dimensional multivariate Gaussians can be used as the emission density f, it is unrealistic to assume that such embeddings follow simple Gaussian distributions. This work explores more complex transformed distributions using normalizing flows.

[1]They include the expansion rules generating the top node in the tree.

3 Normalizing flows

Flow models (Dinh et al., 2015, 2017; Kingma and Dhariwal, 2018) are a class of deep generative models that model unknown yet complex distributions by transforming the observation through a series of invertible transformations to create latent representations to be used with known distributions like Gaussians. For PCFG induction with embeddings, we first consider the generative story for the observed embeddings. Let c_η be a category label where $\eta \in \{1, 2\}^*$ is a Gorn address specifying a path of left or right branches from the root. $\mathbf{M} \in \mathbb{R}^{C \times D}$ is the matrix of the means of the Gaussian distributions for the latent representations, and $\mathbf{S} \in \mathbb{R}^{C \times D}$ the diagonal covariances with $\mathbf{L} = [\mathbf{M}\ \mathbf{S}]$. A probability model over trees may be defined as follows:

1. Sample an expansion decision Term $\sim$ Bernoulli$\left(\frac{1}{1+\exp(-\delta_{c_\eta}{}^\top \mathbf{d})}\right)$ to expand node η with category c_η to a lexical item, or to a binary branch.

2. If expanded as a binary branch (Term=0), given the category of the node c_η, sample a non-terminal expansion, $c_{\eta 1}\ c_{\eta 2} \sim \text{Mult}\left(\frac{\exp(\delta_{c_\eta}{}^\top \mathbf{N})}{\exp(\delta_{c_\eta}{}^\top \mathbf{N})\mathbf{1}}\right)$.

3. If lexically expanded (Term = 1), sample from Gaussian with diagonal covariance over latent representations: $\mathbf{h}_w \sim \mathcal{N}(\delta_{c_\eta}{}^\top \mathbf{M}, \text{diag}(\delta_{c_\eta}{}^\top \mathbf{S}))$.

4. Again, if Term=1, transform the latent representation deterministically to generate the observed embedding $\mathbf{x}_w$ for word w: $\mathbf{x}_w = g(\mathbf{h}_w)$.

In order to compute the likelihood given the observation, we need to invert this process. If we integrate over $\mathbf{x}'_w = g(\mathbf{h}_w)$, with the change-of-variable formula, we have:

$$
\begin{aligned}
f_c(\mathbf{x}_w; \delta_c^\top \mathbf{L}) &= \int \mathsf{P}(c \rightarrow \mathbf{h}_w)\, \delta(\mathbf{x}_w - g(\mathbf{h}_w))\, \mathrm{d}\mathbf{h}_w \\
&= \int \mathsf{P}(c \rightarrow g^{-1}(\mathbf{x}'_w))\, \delta(\mathbf{x}_w - \mathbf{x}'_w) \left|\det \frac{\partial g^{-1}}{\partial \mathbf{x}'_w}\right| \mathrm{d}\mathbf{x}'_w \\
&= \mathsf{P}(c \rightarrow g^{-1}(\mathbf{x}_w)) \cdot \left|\det \frac{\partial g^{-1}}{\partial \mathbf{x}_w}\right|, \qquad (4)
\end{aligned}
$$

where δ is the Dirac delta function. This can be used to directly compute the likelihood of the observed embedding exactly given a category. In order to make this calculation tractable, the requirements on g^{-1} are usually (1) that it is invertible, and (2) that computing the log Jacobian determinant is possible without calculating the full Jacobian matrix or its full determinant. Note that g need not be explicitly constructed as it is only used in sampling, not in inference.

Given a tree as a set τ of nodes η undergoing non-terminal expansions $c_\eta \rightarrow c_{\eta 1}\ c_{\eta 2}$ (where $\eta \in \{1, 2\}^*$), and a set τ' of nodes η undergoing terminal emissions $c_\eta \rightarrow \mathbf{x}_\eta$ (where $\mathbf{x}_\eta$ is an embedding for the word at node η), the marginal probability of a sentence σ_i can be computed as:

$$
\mathsf{P}(\sigma_i) = \sum_{\tau, \tau'} \prod_{\eta \in \tau} \mathsf{P}(c_\eta \rightarrow c_{\eta 1}\ c_{\eta 2}) \cdot \prod_{\eta \in \tau'} \mathsf{P}(c_\eta \rightarrow \mathbf{x}_\eta) \qquad (5)
$$

The first term in the right hand side can be calculated with Equation 2 and the second term can be calculated with Equation 4. The inside algorithm can be used to calculate the exact marginal probability of the sentence σ, and the best parse can be found using the Viterbi algorithm on the parse chart.

There have been many proposed invertible functions that can be used as g^{-1}. The volume preserving invertible transformation is first proposed by Dinh et al. (2015) in the NICE model and later used in unsupervised learning (He et al., 2018). Because of the volume preserving property, the log Jacobian determinant is always 0. This property may allow the structural features of the original embedding space to be better preserved than other, less restrictive, invertible functions.

The invertible transformation g^{-1} consists of I stacked-up coupling layers. The input $\mathbf{x}$ to it is divided into two equal parts $\mathbf{h}_1^{(0)}, \mathbf{h}_2^{(0)}$:

$$
g^{-1}\left(\begin{bmatrix} \mathbf{h}_1^{(0)} \\ \mathbf{h}_2^{(0)} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{h}_1^{(I)} \\ \mathbf{h}_2^{(I)} \end{bmatrix}, \qquad (6)
$$

and the coupling layers in g^{-1} transform the two parts at alternating layers:

$$
\begin{aligned}
\begin{bmatrix} \mathbf{h}_1^{(i-1)} \\ \mathbf{h}_2^{(i-1)} \end{bmatrix} &= \begin{bmatrix} \mathbf{h}_1^{(i-2)} \\ \mathbf{h}_2^{(i-2)} + q^{(i-1)}(\mathbf{h}_1^{(i-2)}) \end{bmatrix}; \\
\begin{bmatrix} \mathbf{h}_1^{(i)} \\ \mathbf{h}_2^{(i)} \end{bmatrix} &= \begin{bmatrix} \mathbf{h}_1^{(i-1)} + q^{(i)}(\mathbf{h}_2^{(i-1)}) \\ \mathbf{h}_2^{(i-1)} \end{bmatrix}. \qquad (7)
\end{aligned}
$$

The volume-preserving restriction is removed in the coupling layer in the Real NVP model (Dinh et al., 2017), in which the coupling layers transform the inputs as follows:

$$
\begin{aligned}
\begin{bmatrix} \mathbf{h}_1^{(i-1)} \\ \mathbf{h}_2^{(i-1)} \end{bmatrix} &= \begin{bmatrix} \mathbf{h}_1^{(i-2)} \\ \mathbf{h}_2^{(i-2)} \odot \exp(q_1^{(i-1)}(\mathbf{h}_1^{(i-2)})) + q_2^{(i-1)}(\mathbf{h}_1^{(i-2)}) \end{bmatrix}; \\
\begin{bmatrix} \mathbf{h}_1^{(i)} \\ \mathbf{h}_2^{(i)} \end{bmatrix} &= \begin{bmatrix} \mathbf{h}_1^{(i-1)} \odot \exp(q_1^{(i)}(\mathbf{h}_2^{(i-1)})) + q_2^{(i)}(\mathbf{h}_2^{(i-1)}) \\ \mathbf{h}_2^{(i-1)} \end{bmatrix}. \qquad (8)
\end{aligned}
$$

All $q : \mathbb{R}^{D/2} \rightarrow \mathbb{R}^{D/2}$ in both models can be arbitrary nonlinear transformations. For Real NVP, the log Jacobian determinant is:

$$
\sum_{i=1}^{I/2} \sum_{j=1}^{D} (q_1^{(2i-1)}(\mathbf{h}_1^{(2i-2)}) + q_1^{(2i)}(\mathbf{h}_2^{(2i-1)}))_{[j]} \qquad (9)
$$

, where the sum over j is the sum over the result vector.

4 Regularization

In order to avoid undesirable yet possible grammars, we impose two linguistically-motivated regularization terms onto the model. It has been observed (Johnson et al., 2007; Jin et al., 2018a) that natural language grammars are sparse. In Bayesian induction models, the prior over PCFG rule probabilities usually encourages sparsity. In experiments described in this paper, an L1 regularization term is also imposed on the expansion parameters to encourage sparsity. Secondly, for the emission parameters, we want to discourage the model from finding a solution in which all words are equally likely to be generated by any category, so we impose a second regularization term on the model to encourage the rows of $\mathbf{M}$ to be far apart. The flow models can learn arbitrary transformations over the pretrained context embeddings. Because each token in the corpus has an embedding, the flow models may learn transformations that cue off arbitrary information in those embeddings, effectively making changes to observations. An Euclidean distance penalty is put between the output of the flow transformation $g^{-1}(\mathbf{x}_\eta)$ and the input embedding $\mathbf{x}_\eta$ to penalize the output drifting too far from the input embedding. The final objective for optimization is:

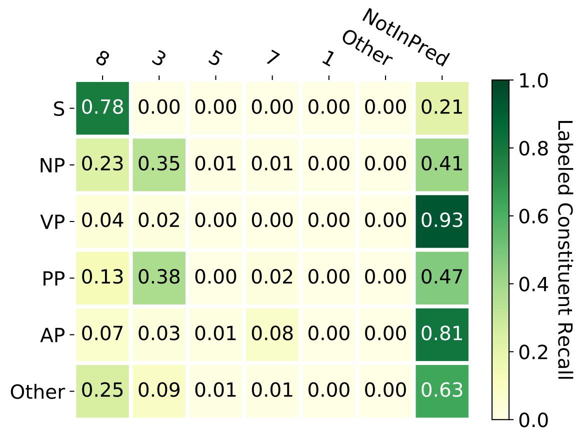
$$\begin{aligned} \mathrm{L}(\sigma) = & \frac{1}{|\sigma|}\sum_{i=0}^{|\sigma|}\log \mathsf{P}(\sigma_i) - \lambda_0 \sum_{a,b,c} \|\mathsf{P}(c \rightarrow a\ b)\|_1 \\ & + \lambda_1 \sum_{d,e} \|\delta_d^{\top}\mathbf{M} - \delta_e^{\top}\mathbf{M}\|_2 \\ & + \lambda_2 \sum_{\eta\in\sigma_i} \|g^{-1}(\mathbf{x}_\eta) - \mathbf{x}_\eta\|_2, \qquad (10) \end{aligned}$$

where σ is a minibatch of sentences, a, b, c, d, e are all category labels, λ_0 and λ_1 and λ_2 are the weights for the three regularization terms and $\|\ldots\|_n$ is the n-norm.

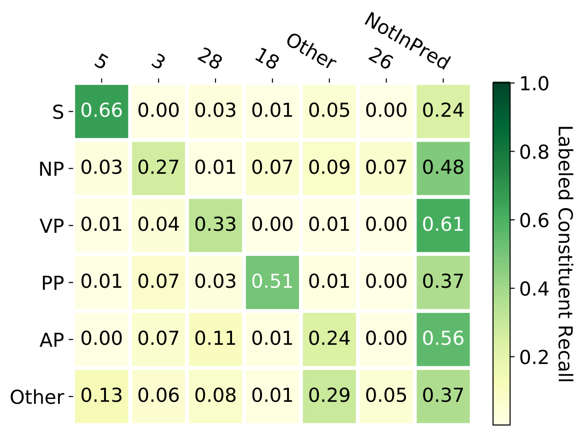
5 Experiments

We report results of labeled parsing evaluation and unlabeled parsing evaluation against existing grammar induction and unsupervised parsing models. We evaluate our models on full English (The Penn Treebank; Marcus et al., 1993), Chinese (The Chinese Treebank 5.0; Xia et al., 2000) and German (NEGRA 2.0; Skut et al., 1998) constituency treebanks and the 20-or-fewer-word subsets for labeled parsing performance.[2] For unlabeled parsing evaluation, we first report results on a set of languages with complex morphology chosen prior to evaluation. This set includes Czech and Russian, which are fusional languages, Korean and Uyghur, which are agglutinative languages, and Finnish, which has elements of both types. Dependency trees from the Universal Dependency Treebank (Nivre et al., 2016) of these languages are converted into constituency trees (Collins et al., 1999) by keeping largest constituents where there is a single incoming and no outgoing dependency arc. For example, constituents like noun phrases that are kept in the conversion may only have one incoming arc from the main verb, and no outgoing arc to any modifier. Each dataset has 15000 sentences randomly sampled from the dependency treebank if the treebank has more than enough sentences, or is augmented with sentences randomly sampled from Wikipedia if it has fewer. Finally unlabeled parsing experiments on the three constituency treebanks are reported, one following Jin et al. (2018a) and one following Htut et al. (2018).

[2] WSJ20test is second half of WSJ20.



(a) the DIMI system.



(b) the flow-based system.

Figure 1: The confusion matrices for DIMI and the flow-based system on the constituents in NEGRA20. The runs with best RVM scores are chosen for plotting. NotInPred means the proportion of gold constituents not in predicted trees.

The hyperparameters of the model for all experiments are tuned on the Brown Corpus portion of the Penn Treebank. We set the number of categories C to 30, the sparsity constraint strength λ_0 to be 100, and the categorical distance constraint strength λ_1 to be 0.0001. g^{-1} is set to have 8 coupling layers with $q^(i)$ being a feed-forward network with one hidden layer for both NICE and Real NVP, following He et al. (2018). We train the system until the marginal likelihood over the whole training set starts to oscillate, around 10,000 batches for smaller corpora and around 20,000 for larger corpora. Because the inside algorithm is quadratic to the length of the sentences, the batch size for training gets quadratically smaller from 400 to 1 as sentences get longer. We use the Adam optimizer (Kingma and Ba, 2015), initialized with learning rates 0.1 for **d** and **N**, and 0.001 for **L** and parameters in g^{-1}. Means and standard deviations of evaluation metrics are reported in tables with 10 runs of the proposed system.

We use ELMo embeddings with 1024 dimensions (Peters et al., 2018) from averaging representations from two BiLSTM layers and the word encoder in ELMo for all languages (Che et al., 2018).[3] These embeddings are trained with sentences from Wikipedia and Common Crawl. We initialize $\mathbf{d}$ and $\mathbf{N}$ with multinomials drawn from a Dirichlet distribution with 0.2 as the concentration parameter, following PCFG induction work with Bayesian models (Jin et al., 2018b). We assign the same diagonal variance matrix to all latent Gaussian distributions, calculated empirically from embeddings from 5000 randomly sampled sentences. $\mathbf{M}$ is initialized with the empirical mean of the same sampled embeddings, but with random Gaussian noise added to each row. The parameters of the normalizing flow g^{-1} are initialized from a uniform distribution with 0 mean and a standard deviation of $\sqrt{1/D}$.

For labeled constituency evaluation, we compare against the state-of-the-art PCFG induction system DIMI (D2K15, depth bounded at 2 and 15 categories) (Jin et al., 2018a) which takes word tokens as input and produces labeled trees.[4] For unlabeled constituency evaluation, results from other unsupervised systems are used for comparison, including CCL (Seginer, 2007), UPPARSE (Ponvert et al., 2011), PRPN (Shen et al., 2018), as well as systems which use gold POS tags: DMV+CCM (Klein and Manning, 2002) and UML-DOP (Bod, 2006).

5.1 Labeled parsing evaluation

Metric: Labeled trees induced by DIMI (Jin et al., 2018a) and the flow-based system are evaluated on six different datasets. In this evaluation, predicted labels of the induced constituents that are in the gold trees are compared against gold labels of these constituents[5] using V-Measure (Rosenberg and Hirschberg, 2007). Recall of the induced trees is used to weight these V-Measure scores. The final Recall-V-Measure (RVM) score is computed as the product of these two measures. RVM can be maximized when the gold constituents are included in the induced trees and their clustering is consistent with gold annotation. RVM is equal to unlabeled recall when the matching constituents have the same clustering of labels as the gold annotation.

Results: Left- and right-branching baselines are constructed by assigning 21 random labels[6] to constituents in purely left- and right-branching trees. However, both branching baselines perform poorly in this evaluation, due to the fact that there is no straightforward way to assign labels to constituent spans that may correspond to how gold labels are organized. VM scores for both baselines are close to 0, leading to RVM scores close to 0. Table 1 shows RVM scores for both the DIMI system and the flow-based system. For the labeled grammar induction systems, results show that the flow-based model outperforms DIMI on two of the three test datasets. Table 3 shows only the performance of the systems on bracketing. Although DIMI performs much better than the flow-based system in terms of bracketing F1 on WSJ20test, the flow-based system's performance on average RVM is much closer to DIMI, which indicates that the flow-based system assigns more consistent labels to constituents than DIMI. On CTB20 and NEGRA20, where the bracketing performance of the flow-based system is better, this system outperforms DIMI by a large margin on RVM. Also, runs with the highest performance on bracketing are not the highest on RVM in general, showing that for labeled induction models, bracketing accuracy may be traded for labeling accuracy.

Confusion matrix: Figure 1 shows the gold constituent recall on NEGRA20 for the two labeled grammar induction systems. We show 5 main phrasal categories in gold annotation and in predicted trees. Grammars from DIMI are prone to category collapse where only a few categories are active as non-terminal. Figure 1a shows that categories 8 and 3 are the main active categories containing the majority of all constituents, with category 8 covering 78% of all S categories, 23% of NPs, and many others. In Figure 1b, the clear diagonal pattern shows that the gold categories do have separate corresponding predicted categories. For example, VP is almost exclusively in category 3 and PP is predominately in category 18. Similar

[3] https://github.com/HIT-SCIR/ELMoForManyLangs.

[4] The DB-PCFG system (Jin et al., 2018b) is formally equivalent to the DIMI system.

[5] The maximal projection category is used when a span is labeled with several categories in the gold annotation. All functional tags are removed.

[6] There are 21 phrase level tags in the Penn Treebank II tag set.

Model	WSJ20test		WSJ		CTB20		CTB		NEGRA20		NEGRA	
	$\mu(\sigma)$	max	$\mu(\sigma)$	max	$\mu(\sigma)$	max	$\mu(\sigma)$	max	$\mu(\sigma)$	max	$\mu(\sigma)$	max
DIMI	**23.0**(6.5)	**34.1**	-	-	15.4(4.4)	20.7	-	-	13.6(1.6)	17.5	-	-
this work	22.8(6.0)	24.0	**22.2**(3.8)	**27.0**	**19.7**(1.9)	**24.0**	**13.8**(3.4)	**20.2**	**26.2**(2.8)	**30.4**	**24.5**(2.7)	**29.1**

Table 1: Recall-V-Measure scores for labeled grammar induction models trained on the listed treebanks with punctuation. For all tables, μ (σ) means the mean (standard deviation) of the reported scores.

Lang.	LB	RB	DIMI μ (σ)	this work μ (σ)
Czech	24.8	50.3	49.3 (8.5)	**52.9** (4.7)
Finnish	30.5	52.1	49.0 (5.0)	**52.5** (5.2)
Korean	40.4	20.2	22.6 (2.1)	**51.1** (2.6)
Russian	45.5	28.7	50.2 (8.1)	**58.0** (4.7)
Uyghur	45.8	24.6	33.0 (3.2)	**54.1** (1.4)

Table 2: Recall scores on a set of morphologically rich languages for the proposed system, DIMI and the left- and right-branching baselines.

System	WSJ20test	CTB20	NEGRA20
CCL	60.9	37.1	33.7
UPPARSE	43.9	38.2	47.7
DB-PCFG	60.5	-	-
DIMI	**63.1**	38.9	40.8
this work	51.7	**43.5**	**48.2**

Table 3: Unlabeled parsing F1 scores for different grammar induction systems trained on only the 20 words or less subsets of the three constituency treebanks as in Jin et al. (2018a).

patterns can also be observed in figures of labeled constituent precision and recall on other datasets in the appendix.

5.2 Unlabeled parsing evaluation

We additionally perform three unlabeled parsing evaluations against baseline systems. The first experiment uses a set of dependency-derived treebanks in morphologically rich languages to examine how morphology is used by the proposed system. The second experiment induces on datasets used in Jin et al. (2018a) and the final experiment uses the WSJ, CTB and NEGRA datasets without any punctuation for evaluation against published results by Htut et al. (2018).

Morphologically rich languages: Table 2 shows unlabeled parsing performance on the morphologically rich languages described at the beginning of this section, compared against branching baselines and DIMI. There is a substantial performance improvement observed across all languages when context embeddings are used as observations. Korean and Uyghur both have very sparse vocabulary, leading to poor performance of the DIMI system.

Constituency treebanks: We also compare the flow-based system to published unlabeled parsing results from previous work. Table 3 shows the unlabeled parsing F1 scores for several grammar induction systems on the WSJ20test, CTB20 and NEGRA20 datasets reported in Jin et al. (2018a). Posterior inference on constituents (PIoC) proposed in Jin et al. (2018a) is also used with parse trees from 10 runs of the flow-based system. The flow-based system is able to produce more accurate trees on the CTB20 and NEGRA20 datasets despite not being depth-bounded. However, its performance is subpar on the WSJ20test dataset.

Finally, the flow-based model is compared against other unsupervised parsing models on the three full constituency treebanks and their 10-or-fewer-word subsets, trained with sentences without punctuation, following Htut et al. (2018). The results are shown in Table 4. First, the flow-based system performs better than reported results from all systems using raw text only on both NEGRA and CTB, showing that the system is able to accurately generate structure. Second, there is a smaller performance gap between the flow-based system and the best-performing one on WSJ than on WSJ10.

The fact that the flow-based model underperforms on English may be due to the fact that the English vocabulary contains a relatively large number of high frequency words, which makes contexts for words similar, showing up as the context embeddings for different words are similar to each other. This confuses the model because it relies on the observed embeddings being distinct and representative for induction. 50,000 pairs of ELMo embeddings of different words are ran-

Model	WSJ10		WSJ		CTB10		CTB		NEGRA10		NEGRA	
	$\mu(\sigma)$	max	$\mu(\sigma)$	max	$\mu(\sigma)$	max	$\mu(\sigma)$	max	$\mu(\sigma)$	max	$\mu(\sigma)$	max
CCL	67.3(0.0)	67.3	**44.9**(0.0)	**44.9**	47.8(0.0)	47.8	21.1(0.0)	21.1	48.0(0.0)	48.0	27.6(0.0)	27.6
UPPARSE	44.8(0.0)	44.8	23.6(0.0)	23.6	44.7(0.0)	44.7	24.2(0.0)	24.2	53.4(0.0)	53.4	33.4(0.0)	33.4
PRPN-UP	62.2(3.9)	70.3	26.0(2.3)	32.8	-	-	-	-	-	-	-	-
PRPN-LM	**70.5**(0.4)	**71.3**	37.4(0.3)	38.1	-	-	-	-	-	-	-	-
DIMI	49.0(4.8)	55.8	-	-	41.1(2.9)	45.9	-	-	47.5 (2.7)	54.1	-	-
this work	56.0(6.1)	63.6	38.5(3.9)	42.7	49.4(1.3)	**50.7**	**29.2**(2.1)	**31.9**	51.8 (3.1)	**58.5**	**37.1**(2.5)	**41.2**
RB	61.7(0.0)	61.7	39.5(0.0)	39.5	**50.4**(0.0)	50.4	21.8(0.0)	21.8	43.3(0.0)	43.3	22.8(0.0)	22.8
LB	28.7(0.0)	28.7	11.6(0.0)	11.6	35.8(0.0)	35.8	11.7(0.0)	11.7	35.1(0.0)	35.1	16.9(0.0)	16.9
DMV+CCM	*77.6(0.0)*	*77.6*	-	-	-	-	-	-	*63.9(0.0)*	*63.9*	-	-
UML-DOP	*82.9(0.0)*	*82.9*	-	-	-	-	-	-	*67.0(0.0)*	*67.0*	-	-

Table 4: Parsing accuracy scores for different constituency grammar induction systems trained on the full set of the treebanks where punctuation is removed from all data in training and evaluation with results reported in Htut et al. (2018). PRPN models train and test on different subsets of the corpora, whereas other models use the full corpora to train and evaluate. All models except DIMI and this work produce unlabeled trees. DMV+CCM and UML-DOP use gold POS tags as observations for induction, listing here for reference.

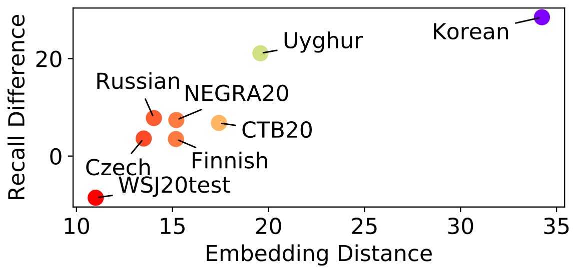


Figure 2: Correlation between recall difference of the flow-based system and DIMI and the average distance between ELMo embeddings.

domly sampled from each dataset, and the average Euclidean distance between these embeddings is calculated. Figure 2 shows that the averaged distance between the embeddings is positively correlated with the gain of the flow-based system to DIMI, indicating the importance of varied contexts for grammar induction.

5.3 Induced interpretable categories

PCFG induction systems usually create syntactic categories that correspond to coarse-grained linguistic classes like nouns and verbs using co-occurrence statistics. However the flow-based system also creates classes that are morphological or semantic in nature. The ability of the system to use morphological and semantic information to help grammar induction is shown in Table 5.

Grammars induced on Korean from the flow-based system are greatly improved over baselines which use words only as input. Korean is an agglutinative language with many morphemes per token, so approaches that treat tokens as words must address severe sparsity issues. As ELMo embeddings include subword information from Korean characters, they may contain information useful for understanding morphology – the nominative clitics 이/가 and the accusative clitics 을/를, for example, may encode strong biases towards a word token being a noun along with its case.

Categories like 11 and 12 in Table 5 reliably capture nouns in the nominative and accusative cases, respectively, even though in both cases the

Cat.	Interp.	Most common words
Korean		
3	ADJ	큰 (big), 많은 (many) 새로운 (new), 중요한 (important)
11	N-NOM	사람이 (person), 일이 (work) 사람들이 (people), 문제가 (problem)
12	N-ACC	사실을 (fact), 영향을 (influence) 일을 (work), 의미를 (meaning)
German		
7	DAT	den, dem, einem, diesem, ihren, seinen
8	GEN	der, des, einer, dieser, seiner, eines
20	NOM/ACC	die, das, der, ein, eine, ihre, keine
Chinese		
1	V-TRANS	提供(provide), 进行(carry out) 举行(hold), 利用(utilize)
14	V-MODAL	要(would like), 会(will) 能(can), 可以(be able)
28	V-SCOMP	说(say), 希望(hope) 认为(think), 指出(point out)

Table 5: Analysis of predicted syntactic categories (Cat.) and their interpreted syntactic categories (Interp.) in runs with highest RVM scores for Korean, German and Chinese. The most common words in each predicted category are listed.

marking clitic differs depending on whether the noun preceding it ends in a vowel or consonant. Similarly, category 3 shows noun-preceding adjectives, which in Korean are formed by verb stems plus ㄴ/은, and the inducer is again able to cluster words with both endings together.

For German, the cased articles also have similar endings. The dative articles usually end with *-en* or *-em*, and the genitive articles usually end with *-er* or *-es*. Having access to the subword information, the flow-based system is able to come up with these distinctions with no supervision, because the cases may provide important clues to relative positions of the following nouns to verbs or prepositions. Contextual information also helps greatly, seen here as the system distinguishes the genitive *der* in category 8 and the nominative/accusative *der* in category 20 in the phrases like *der*(20) *Pächter der*(8) *Junkerstube* (the lessee of the junkerstube).

Finally, for languages like Chinese where there are few morphological markings, semantic information may help the system induce syntactic categories. Category 28 is a category of verbs related to cognition and expression, which also characteristically accepts sentential complements (Vendler, 1972; Fisher et al., 1991). Syntactic categories like these are not seen in systems inducing with word only. This indicates that semantics of these verbs may play a role here, especially since Chinese has no complementizer to signal an upcoming sentential complement.

5.4 Ablation experiments

Model setup		RVM μ (σ)	RVM max
Multi		18.9 (1.6)	21.0
Gauss	+Fasttext	17.5 (1.5)	19.4
	+ELMo	23.4 (2.0)	26.7
NICE	+ELMo	13.9 (4.6)	22.3
	+ELMo+sim	25.7 (2.2)	28.7
	+ELMo+sim+$l1$	25.8 (3.2)	31.2
	+ELMo+sim+$l1$+μDist	26.2 (2.8)	30.3
RNVP	+ELMo+sim+$l1$+μDist	24.1 (3.2)	27.9

Table 6: Parsing performance on the NEGRA20 dataset with different configurations of the model. NICE and RNVP are the NICE and RealNVP models used for modeling emission. Sim, $l1$ and μDist are the similarity penalty, $l1$ and category distance regularizers respectively.

Table 6 shows the ablation and comparison experiments on NEGRA20. ELMo embeddings provide a large performance boost over both the multinomial emission model, which has no access to contextual and subword information, and the Gaussian emission model with character-n-gram-based Fasttext embeddings (Joulin et al., 2016), showing that both context and subword information helps grammar induction. The three linguistically-motivated regularization terms help the model perform even better. Most notably, the similarity performance helps the flow models greatly by restricting the freedom that the flow models have for changing the context embeddings, indicating that the information in context embeddings is valuable for induction. The Real NVP model produces higher data likelihood but its performance is lower than other NICE-based models, indicating that the volume-preserving property of NICE is important for preventing overfitting.

6 Related work

Earlier work on PCFG induction (Carroll and Charniak, 1992; Johnson et al., 2007; Liang et al., 2009; Tu, 2012) shows that directly inducing PCFGs from raw text is difficult. Recent work (Shain et al., 2016; Jin et al., 2018b,a) shows that inducing PCFGs from raw text is possible, and cognitive constraints are useful for helping the induction model find good grammars. Closely related to PCFG induction is the task of unsupervised constituency parsing from raw text where trees are unlabeled. Earlier work (Seginer, 2007; Ponvert et al., 2011) induce unlabeled trees and achieve good results. More recent work (Shen et al., 2018) utilize complex neural architecture for unsupervised parsing and language modeling and also shows good results on English. Although unlabeled parsing evaluation is common, other work (Bisk and Hockenmaier, 2015) has argued for labeled parsing evaluation for grammar induction.

Early unsupervised dependency grammars and part-of-speech induction models (Klein and Manning, 2004; Christodoulopoulos and Steedman, 2010) have been similarly augmented with neural networks and word embeddings (Tran et al., 2016; Jiang et al., 2016). Neural networks provide flexible ways to parameterize distributions, and word embeddings (Mikolov et al., 2013; Pennington et al., 2014) allow these models to use semantic information in these distributed representations. Results show that these improvements produce more accurate dependencies and POS assignments, but these improvements have not been applied to PCFG induction.

Normalizing flows have been shown to be powerful models for complex densities (Dinh et al., 2015, 2017; Rezende and Mohamed, 2015; Papamakarios et al., 2017). He et al. (2018) showed improved performance on POS induction and dependency induction by incorporating normalizing flows into baseline models (Klein and Manning, 2004; Lin et al., 2015).

7 Conclusion

In this work, we proposed a neural PCFG inducer which employs context embeddings (Peters et al., 2018) in a normalizing flow model (Dinh et al., 2015) to extend PCFG induction to using semantic and morphological information. Linguistically motivated sparsity and categorical distance constraints are also imposed on the inducer as regularization. Labeled and unlabeled evaluation shows that the PCFG induction model with normalizing flow and context embeddings produces grammars with state-of-the-art accuracy on a variety of different languages. We observed consistent and meaningful use of labels at phrasal and lexical levels by the flow-based model. Ablation further shows a positive effect of normalizing flow, context embeddings and proposed regularizers.

References

Omri Abend, Tom Kwiatkowski, Nathaniel J. Smith, Sharon Goldwater, and Mark Steedman. 2017. Bootstrapping language acquisition. In *Cognition*, volume 164, pages 116–143. Elsevier B.V.

Colin Bannard, Elena Lieven, and Michael Tomasello. 2009. Modeling children's early grammatical knowledge. *Proceedings of the National Academy of Sciences of the United States of America*, 106(41):17284–9.

Yonatan Bisk and Julia Hockenmaier. 2015. Probing the linguistic strengths and limitations of unsupervised grammar induction. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, 1:1395–1404.

Rens Bod. 2006. Unsupervised parsing with U-DOP. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 85–92.

Glenn Carroll and Eugene Charniak. 1992. Two experiments on learning probabilistic dependency grammars from corpora. *Working Notes of the Workshop on Statistically-Based NLP Techniques*, (March):1–13.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards Better UD Parsing: Deep Contextualized Word Embeddings, Ensemble, and Treebank Concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Christos Christodoulopoulos and Mark Steedman. 2010. Two Decades of Unsupervised POS induction: How far have we come? *2010 Conference on Empirical Methods in Natural Language Processing*, (October):575–584.

Michael Collins, Lance Ramshaw, Jan Hajič, and Christoph Tillmann. 1999. A Statistical Parser for Czech. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 505–512.

Laurent Dinh, David Krueger, and Yoshua Bengio. 2015. NICE: Non-Linear Independent Components Estimation. In *ICLR Workshop*.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2017. Density estimation using Real NVP. *ICLR*.

Cynthia Fisher, Henry Gleitman, and Lila R Gleitman. 1991. On the semantic content of subcategorization frames. *Cognitive Psychology*, 23(3):331–392.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Unsupervised Learning of Syntactic Structure with Invertible Neural Projections. In *EMNLP*, pages 1292–1302. Association for Computational Linguistics.

Phu Mon Htut, Kyunghyun Cho, and Samuel R Bowman. 2018. Grammar Induction with Neural Language Models: An Unusual Replication. In *EMNLP*, pages 4998–5003.

Yong Jiang, Wenjuan Han, and Kewei Tu. 2016. Unsupervised neural dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 61503248, pages 763–771.

Lifeng Jin, Finale Doshi-Velez, Timothy A Miller, William Schuler, and Lane Schwartz. 2018a. Depth-bounding is effective: Improvements and evaluation of unsupervised PCFG induction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Lifeng Jin, Finale Doshi-Velez, Timothy A Miller, William Schuler, and Lane Schwartz. 2018b. Unsupervised Grammar Induction with Depth-bounded PCFG. *Transactions of the Association for Computational Linguistics*.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Bayesian Inference for PCFGs via Markov chain Monte Carlo. *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 139–146.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Diederik P Kingma and Prafulla Dhariwal. 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. *NIPS*.

Dan Klein and Christopher D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 128–135.

Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, volume 1, pages 478–485.

Percy Liang, Michael I Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - ACL-IJCNLP '09*, volume 1, page 91.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised POS Induction with Word Embeddings. In *Proceedings of Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pages 1311–1316.

Brian MacWhinney and Elizabeth Bates. 1993. *The Crosslinguistic Study of Sentence Processing*. Cambridge University Press, New York.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3:1–12.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D Manning, Ryan Mcdonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of Language Resources and Evaluation Conference*.

George Papamakarios, Theo Pavlakou, and Iain Murray. 2017. Masked Autoregressive Flow for Density Estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Kim Plunkett and Clair Wood. 2004. The development of children's understanding of grammar. *Cognitive and language development in children. Oxford: Blackwell*, pages 163–204.

Elias Ponvert, Jason Baldridge, and Katrin Erk. 2011. Simple unsupervised grammar induction from raw text with cascaded finite state models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1077–1086.

Danilo Jimenez Rezende and Shakir Mohamed. 2015. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on Machine Learning*.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.

Yoav Seginer. 2007. Fast Unsupervised Incremental Parsing. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 384–391.

Cory Shain, William Bryce, Lifeng Jin, Victoria Krakovna, Finale Doshi-Velez, Timothy Miller, William Schuler, and Lane Schwartz. 2016. Memory-bounded left-corner unsupervised grammar induction on child-directed input. In *Proceedings of the International Conference on Computational Linguistics*, pages 964–975.

Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron Courville. 2018. Neural Language Modeling by Jointly Learning Syntax and Lexicon. In *ICLR*.

Wojciech Skut, Thorsten Brants, Brigitte Krenn, and Hans Uszkoreit. 1998. A Linguistically Interpreted Corpus of German Newspaper Text. In *Proceedings of the ESSLLI Workshop on Recent Advances in Corpus Annotation.*, page 7.

Ke Tran, Yonatan Bisk, Ashish Vaswani, Daniel Marcu, and Kevin Knight. 2016. Unsupervised Neural Hidden Markov Models. In *Proceedings of the Workshop on Structured Prediction for NLP*.

Kewei Tu. 2012. *Unsupervised learning of probabilistic grammars*. Ph.D. thesis.

Zeno Vendler. 1972. Res cogitans: An essay in rational psychology.

Fei Xia, Martha Palmer, Nianwen Xue, Mary Ellen Ocurowski, John Kovarik, Fu-Dong Chiou, Shizhe Huang, Tony Kroch, and Mitch Marcus. 2000. Developing Guidelines and Ensuring Consistency for Chinese Text Annotation. In *Proceedings of the Second Language Resources and Evaluation Conference*.