Enhancing Unsupervised Generative Dependency Parser with Contextual Information

Wenjuan Han, Yong Jiang and **Kewei Tu**[*]
{hanwj,jiangyong,tukw}@shanghaitech.edu.cn
School of Information Science and Technology
ShanghaiTech University, Shanghai, China

Abstract

Most of the unsupervised dependency parsers are based on probabilistic generative models that learn the joint distribution of the given sentence and its parse. Probabilistic generative models usually explicit decompose the desired dependency tree into factorized grammar rules, which lack the global features of the entire sentence. In this paper, we propose a novel probabilistic model called discriminative neural dependency model with valence (D-NDMV) that generates a sentence and its parse from a continuous latent representation, which encodes global contextual information of the generated sentence. We propose two approaches to model the latent representation: the first deterministically summarizes the representation from the sentence and the second probabilistically models the representation conditioned on the sentence. Our approach can be regarded as a new type of autoencoder model to unsupervised dependency parsing that combines the benefits of both generative and discriminative techniques. In particular, our approach breaks the context-free independence assumption in previous generative approaches and therefore becomes more expressive. Our extensive experimental results on seventeen datasets from various sources show that our approach achieves competitive accuracy compared with both generative and discriminative state-of-the-art unsupervised dependency parsers.

1 Introduction

Dependency parsing is a very important task in natural language processing. The dependency relations identified by dependency parsing convey syntactic information useful in subsequent applications such as semantic parsing, information extraction, and question answering. In this paper, we focus on unsupervised dependency parsing, which aims to induce a dependency parser from training sentences without gold parse annotation.

Most previous approaches to unsupervised dependency parsing are based on probabilistic generative models, for example, the Dependency Model with Valence (DMV) (Klein and Manning, 2004) and its extensions (Cohen and Smith, 2009; Headden III et al., 2009; Cohen and Smith, 2010; Berg-Kirkpatrick et al., 2010; Gillenwater et al., 2010; Jiang et al., 2016). A disadvantage of such approaches comes from the context-freeness of dependency grammars, a strong independence assumption that limits the information available in determining how likely a dependency is between two words in a sentence. In DMV, the probability of a dependency is computed from only the head and child tokens, the dependency direction, and the number of dependencies already connected from the head token. Additional information used for computing dependency probabilities in later work is also limited to local morpho-syntactic features such as word forms, lemmas and categories (Berg-Kirkpatrick et al., 2010), which does not break the context-free assumption.

More recently, researchers have started to utilize discriminative methods in unsupervised dependency parsing based on the idea of discriminative clustering (Grave and Elhadad, 2015), the CRFAE framework (Cai et al., 2017) or the neural variational transition-based parser (Li et al., 2019). By conditioning dependency prediction on the whole input sentence, discriminative methods are capable of utilizing not only local information, but also global and contextual information of a dependency in determining its strength. Specifically, both Grave and Elhadad (2015) and Cai et al. (2017) include in the feature set of a dependency the information of the tokens around the head or child token of the dependency. In this way,

[*] Corresponding author

they break the context-free independence assumption because the same dependency would have different strength in different contexts. Besides, Li et al. (2019) propose a variational autoencoder approach based on Recurrent Neural Network Grammars.

In this paper, we propose a novel approach to unsupervised dependency parsing in the middle between generative and discriminative approaches. Our approach is based on neural DMV (Jiang et al., 2016), an extension of DMV that employs a neural network to predict dependency probabilities. Unlike neural DMV, however, when computing the probability of a dependency, we rely on not only local information as in DMV, but also global and contextual information from a compressed representation of the input sentence produced by neural networks. In other words, instead of modeling the joint probability of the input sentence and its dependency parse as in a generative model, we model the conditional probability of the sentence and parse given global information of the sentence. Therefore, our approach breaks the context-free assumption in a similar way to discriminative approaches, while it is still able to utilize many previous techniques (e.g., initialization and regularization techniques) of generative approaches.

Our approach can be seen as an autoencoder. The **decoder** is a conditional generative neural DMV that generates the sentence as well as its parse from a continuous representation that captures the global features of the sentence. To model such global information, we propose two types of **encoders**, one deterministically summarizes the sentence with a continuous vector while the other probabilistically models the continuous vector conditioned on the sentence. Since the neural DMV can act as a fully-fledged unsupervised dependency parser, the encoder can be seen as a supplementary module that injects contextual information into the neural DMV for context-specific prediction of dependency probabilities. This is very different from the previous unsupervised parsing approach based on the autoencoder framework (Cai et al., 2017; Li et al., 2019), in which the encoder is a discriminative parser and the decoder is a generative model, both of which are required for performing unsupervised parsing.

Our experiments verify that our approach achieves a comparable result with recent state-of-the-art approaches on extensive datasets from various sources.

2 Related Work

2.1 Dependency Model with Valence

The Dependency Model with Valence (DMV) (Klein and Manning, 2004) is an extension of an earlier dependency model (Carroll and Charniak, 1992) for grammar induction. Different from the earlier model, there are three types of probabilistic grammar rules in DMV, namely `ROOT`, `CHILD` and `CHILD` rules. To generate a token sequence and its corresponding dependency parse tree, the DMV model first generates a token c from the `ROOT` distribution $p(c|root)$. Then the generation continues in a recursive procedure. At each generation step, it makes a decision as to whether a new token needs to be generated from the current head token h in the dir direction by sampling a STOP or CONTINUE symbol dec from the `CHILD` distribution $p(dec|h, dir, val)$ where val is an indicator representing whether token h has already generated a token before. If dec is CONTINUE, a new token is generated from the `CHILD` distribution $p(c|h, dir, val)$. If dec is STOP, then the generation process switches to a new direction or a new head token. DMV can be trained from an unannotated corpus using the expectation-maximization algorithm.

2.2 Neural DMV

The DMV model is very effective in inducing syntactic dependency relations between tokens in a sentence. One limitation of DMV is that correlation between similar tokens (such as different verb POS tags) is not taken into account during learning and hence rules involving similar tokens have to be learned independently. Berg-Kirkpatrick et al. (2010) proposed a feature-based DMV model in which the grammar rule probabilities are computed by a log-linear model with manually designed features that reflect token similarity. Jiang et al. (2016) proposed the neural DMV model which learns token embeddings to better capture correlations between tokens and utilizes a neural network to calculate grammar rule probabilities from the embeddings. Both approaches significantly outperform the original DMV. However, because of the strong independence assumption in such generative models, they can only utilize local information of a grammar rule (e.g., the head and child tokens, direction, and valence) when computing its probability.

3 Discriminative Neural DMV

We extend the neural DMV such that when predicting the probability of a grammar rule in parsing a sentence, the model incorporates not only local information of the rule but also global information of the sentence. Specifically, we model each grammar rule probability conditioned on a continuous vector. We therefore call our model the discriminative neural DMV (D-NDMV). In this way, the probability of a dependency rule becomes sensitive to the input sentence, which breaks the context-free assumption in the neural DMV. Here, we provide two approaches to model this global continuous vector.

3.1 *Deterministic Variant* for D-NDMV

Model

Suppose we have a sentence (i.e., a word sequence) $\mathbf{w}$, the corresponding POS tag sequence $\mathbf{x}$, and the dependency parse $\mathbf{z}$ which is hidden in unsupervised parsing. DMV and its variants model the joint probability of the POS tag sequence and the parse $P(\mathbf{x}, \mathbf{z})$ and, because of the context-free assumption, factorize the probability based on the grammar rules used in the parse. In contrast, to the global features of the sentence, we model the conditional probability of the POS tag sequence and the parse given the sequence $\mathbf{w}$: $P(\mathbf{x}, \mathbf{z}|\mathbf{w})$. We assume conditional context-freeness and factorize the conditional probability based on the grammar rules.

$$P_{\Theta}(\mathbf{x}, \mathbf{z}|\mathbf{w}) = \prod_{r \in (\mathbf{x}, \mathbf{z})} p(r|\mathbf{w}) \tag{1}$$

where r ranges over all the grammar rules used in the parse $\mathbf{z}$ of tag sequence $\mathbf{x}$, Θ is the set of parameters to compute parameters of the distribution. Since one can reliably predict the POS tags $\mathbf{x}$ from the words $\mathbf{w}$ without considering the parse $\mathbf{z}$ (as most POS taggers do), to avoid degeneration of the model, we compute $p(r|\mathbf{w})$ based on global information of $\mathbf{w}$ produced by a long short-term memory network (LSTM).

Figure 1 shows the neural network structure for parametering $p(c|h, dir, val, \mathbf{w})$, the probabilities of CHILD rules given the input sentence $\mathbf{w}$. The structure is similar to the one used in neural DMV except for using LSTM sentence encoder to get the representation $\mathbf{s}$ from the sentence $\mathbf{w}$. The embeddings of the head POS tag and valence are represented by $\mathbf{v}_h$ and $\mathbf{v}_{val}$. The concatenation $[\mathbf{v}_{val}; \mathbf{v}_h; \mathbf{s}]$ is fed into a fully-connected layer with a direction-specific weight matrix $\mathbf{W}_{dir}$ and the ReLU activation function to produce the hidden layer g. All possible child POS tags are represented by the matrix $\mathbf{W}_{chd}$. The i-th row of $\mathbf{W}_{chd}$ represents the output embedding of the i-th POS tag. We take the product of the hidden layer g and the child matrix $\mathbf{W}_{chd}$ and apply a softmax function to obtain the CHILD rule probabilities. ROOT and CHILD rule probabilities are computed in a similar way.

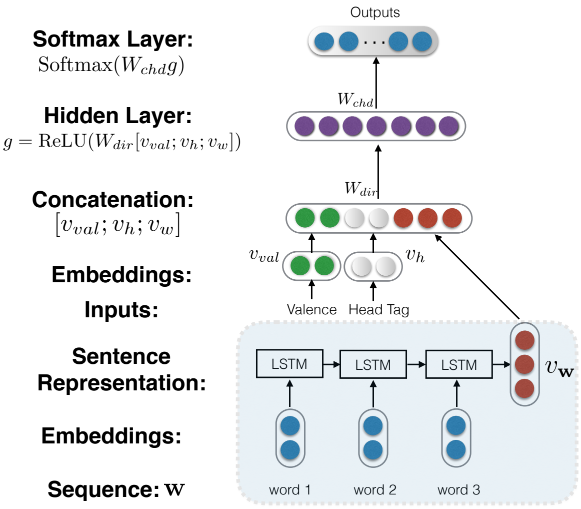

Figure 1: The neural network structure for computing the probabilities of CHILD rules.

Since the mapping from $\mathbf{w}$ to $\mathbf{s}$ is deterministic, we call it the *deterministic variant* of D-NDMV. To make the notations consistent with subsequent sections, we add an auxiliary random variable $\mathbf{s}$ to represent the global information of sentence $\mathbf{w}$. The probabilistic distribution of $\mathbf{s}$ is defined as,

$$P_{\Phi}(\mathbf{s}|\mathbf{w}) = \delta(\mathbf{s} - v_{\mathbf{w}}) \tag{2}$$

where Φ is the set of parameters of the LSTM neural network.

Figure 2 (left) shows the directed graphical representation of this model. If we diminish the capacity of $\mathbf{s}$ (e.g., by shrinking its dimension), then our model gradually reduces to neural DMV.

Parsing

Given a *deterministic variant* with fixed parameters Φ, Θ. we can parse a sentence represented by POS tag sequence $\mathbf{x}$ and word sequence $\mathbf{w}$

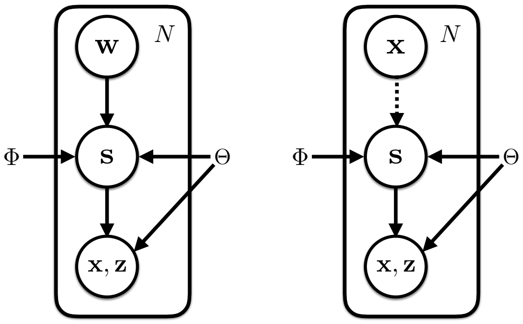

Figure 2: **Left:** the illustration of the *deterministic variant* of D-NDMV as a directed graph. The *deterministic variant* models an autoencoder with $P_\Phi(\mathbf{s}|\mathbf{w})$ as the encoder and $P_\Theta(\mathbf{x}, \mathbf{z}|\mathbf{s})$ as the decoder. **Right:** the illustration of the *variational variant* of D-NDMV as a directed graph. We use dashed lines to denote the variational approximation $q_\Phi(\mathbf{s}|\mathbf{x})$ to the intractable posterior $P_\Phi(\mathbf{s}|\mathbf{x})$, and the solid lines to denote the generative model $P(\mathbf{s})P_\Theta(\mathbf{x}, \mathbf{z}|\mathbf{s})$.

by searching for a dependency tree $\mathbf{z}^*$ which has the highest probability $p(\mathbf{x}, \mathbf{z}|\mathbf{w})$ among the set of valid parse trees $\mathcal{Z}(\mathbf{x})$.

$$\mathbf{z}^* = \arg\max_{\mathbf{z}\in\mathcal{Z}(\mathbf{x})} P_{\Theta,\Phi}(\mathbf{x}, \mathbf{z}|\mathbf{w}) \tag{3}$$

Note that once we compute all the grammar rule probabilities based on $\mathbf{w}$, our model becomes a standard DMV and therefore dynamic programming can be used to parse each sentence efficiently (Klein and Manning, 2004).

Unsupervised Learning

Objective Function: In a typical unsupervised dependency parsing setting, we are given a set of training sentences with POS tagging but without parse annotations. The objective function of learning *deterministic variant* is as follows.

$$J(\Theta, \Phi) = \frac{1}{N}\sum_{i=1}^{N} \log P_{\Theta,\Phi}(\mathbf{x}^{(i)}|\mathbf{w}^{(i)}) \tag{4}$$

The log conditional likelihood is defined as:

$$\log P_{\Theta,\Phi}(\mathbf{x}|\mathbf{w}) = \log \sum_{\mathbf{z}\in\mathcal{Z}(\mathbf{x})} P_{\Theta,\Phi}(\mathbf{x}, \mathbf{z}|\mathbf{w}) \tag{5}$$

We may replace summation with maximization so that it becomes the conditional Viterbi likelihood.

Learning Algorithm: We optimize our objective function using the expectation-maximization (EM) algorithm. Specifically, the EM algorithm alternates between E-steps and M-steps to maximize a lower-bound of the objective function. For each training sentence, the lower bound is defined as:

$$\begin{aligned} Q(q, \Theta, \Phi) = &\log P_{\Theta,\Phi}(\mathbf{x}|\mathbf{w}) \\ &- KL(q(\mathbf{z})\|P_{\Theta,\Phi}(\mathbf{z}|\mathbf{x}, \mathbf{w})) \end{aligned} \tag{6}$$

where $q(\mathbf{z})$ is an auxiliary distribution over the latent parse $\mathbf{z}$.

In the E-step, we fix Θ, Φ and maximize $Q(q, \Theta, \Phi)$ with respect to q. The maximum is reached when the Kullback-Leibler divergence is zero, i.e.,

$$q(\mathbf{z}) = P_{\Theta,\Phi}(\mathbf{z}|\mathbf{x}, \mathbf{w}) \tag{7}$$

Based on the optimal q, we compute the expected counts $E_{q(\mathbf{z})}c(r, \mathbf{x}, \mathbf{z})$ using the inside-outside algorithm, where $c(r, \mathbf{x}, \mathbf{z})$ is the number of times rule r is used in producing parse $\mathbf{z}$ of tag sequence $\mathbf{x}$.

In the M-step, we fix q and maximize $Q(q, \Theta, \Phi)$ with respect to Θ, Φ. The lower bound now takes the following form:

$$\begin{aligned} Q(\Theta, \Phi) = &\sum_r \log p(r|\mathbf{w}) E_{q(\mathbf{z})} c(r, \mathbf{x}, \mathbf{z}) \\ &- \text{Constant} \end{aligned} \tag{8}$$

where r ranges over all the grammar rules and Constant is a constant value. The probabilities $p(r|\mathbf{w}, \Theta, \Phi)$ are computed by the neural networks and we can back-propagate the objective $Q(\Theta, \Phi)$ into the parameters of the neural networks.

We initialize the model either heuristically (Klein and Manning, 2004) or using a pre-trained unsupervised parser (Jiang et al., 2016); then we alternate between E-steps and M-steps until convergence.

Note that if we require $q(\mathbf{z})$ to be a delta function, then the algorithm becomes hard-EM, which computes the best parse of each training sentence in the E-step and set the expected count to 1 if the rule is used in the parse and 0 otherwise. It has been found that hard-EM outperforms EM in unsupervised dependency parsing (Spitkovsky et al., 2010; Tu and Honavar, 2012), so we use hard-EM in our experiments.

3.2 *Variational Variant* for D-NDMV

Motivated by (Bowman et al., 2016), we propose to model the global representation s as drawing from a prior distribution, generally a standard

Gaussian distribution. We also propose a variational posterior distribution $q_\Phi(\mathbf{s}|\mathbf{x})$ to approximate this prior distribution. In this way, we formalize it into a variational inference framework. We call this model *variational variant* and illustrate its graphical model in Figure 2 (right). It can be seen from Figure 2 (right) that the *variational variant* shares the same formulation of the encoder part with the variational autoencoder (VAE). Different from the vanilla VAE model with a simple multilayered feedforward neural network as the decoder, our decoder is a generative latent variable model with the structured hidden variable $\mathbf{z}$.

For the learning of the *variational variant*, we use the log likelihood as the objective function and optimize its lower bound. We show the derivation as followings:

$$\begin{aligned}&\log P_{\Phi,\Theta}(\mathbf{x})\\ \geq& -\mathrm{KL}(q_\Phi(\mathbf{s}|\mathbf{x})||p(\mathbf{s})) + E_{q_\Phi(\mathbf{s}|\mathbf{x})}\log P_\Theta(\mathbf{x}|\mathbf{s})\end{aligned} \tag{9}$$

By performing the Monte Carlo method to estimate the expectation w.r.t. $q_\Phi(\mathbf{s}|\mathbf{x})$ and set the number of samples L to 1, we rewrite the second term as:

$$\begin{aligned}&E_{q_\Phi(\mathbf{s}|\mathbf{x})}\log p_\Theta(\mathbf{x}|\mathbf{s})\\ \simeq&\frac{1}{L}\sum_{l=1}^{L}\log\sum_{\mathbf{z}\in\mathcal{Z}(\mathbf{x})}p_\Theta(\mathbf{x},\mathbf{z}|\mathbf{s}^{(l)})\\ =&\log\sum_{\mathbf{z}\in\mathcal{Z}(\mathbf{x})}p_\Theta(\mathbf{x},\mathbf{z}|\mathbf{s}^{(1)})\end{aligned} \tag{10}$$

where $\mathbf{s}^{(l)}$ is estimated by the *reparameterization trick* (Kingma and Welling, 2014), which enables low gradient variances and stabilizes training.

Because this formula is similar to Eq. 5, we can follow the subsequent derivation of *deterministic variant* and learn the *variational variant* using EM. It is worth noting that different from *deterministic variant*, in M-step an additional KL divergence term in Eq. 9 should be optimized by back-propagation.

4 Experiments

We tested our methods on seventeen treebanks from various sources. For each dataset, we compared with current state-of-the-art approaches on the specific dataset.

4.1 Dataset and Setup

English Penn Treebank We conducted experiments on the Wall Street Journal corpus (WSJ) with section 2-21 for training, section 22 for validation and section 23 for testing. We trained our model with training sentences of length ≤ 10, tuned the hyer-parameters on validation sentences of length ≤ 10 the and evaluated on testing sentences of length ≤ 10 (WSJ10) and all sentences (WSJ). We reported the directed dependency accuracy (DDA) of the learned grammars on the test sentences.

Universal Dependency Treebank Following the setup of Jiang et al. (2017); Li et al. (2019), we conducted experiments on selected eight languages from the Universal Dependency Treebank 1.4 (Nivre et al., 2016). We trained our model on training sentences of length ≤ 15 and report the DDA on testing sentences of length ≤ 15 and ≤ 40.

Datasets from PASCAL Challenge on Grammar Induction We conducted experiments on corpora of eight languages from the PASCAL Challenge on Grammar Induction (Gelling et al., 2012). We trained our model with training sentences of length ≤ 10 and evaluated on testing sentences of length ≤ 10 and all sentences.

Note that on the UD Treebanks and PASCAL datasets, we used the same hyper-parameters as in the WSJ experiments without further tuning.

Setup Following previous work, we conducted experiments under the unlexicalized setting where a sentence is represented as a sequence of gold part-of-speech tags with punctuations removed. The embedding length was set to 10 for the head and child tokens and the valence. The sentence embedding length was also set to 10. We trained the neural networks using stochastic gradient descent with batch size 10 and learning rate 0.01. We used the change of the loss on the validation set as the stop criteria. For our methods in the WSJ experiments, we followed Han et al. (2017) and initialized our model using the pre-trained model of Naseem et al. (2010), which significantly increased the accuracy and decreased the variance. For the other experiments, we used a pre-trained NDMV model to initialize our method. We ran our model for 5 times and report the average DDA.

METHODS	WSJ10	WSJ
Systems in Basic Setup		
DMV (Klein and Manning, 2004)	58.3	39.4
LN (Cohen et al., 2008)	59.4	40.5
Convex-MST (Grave and Elhadad, 2015)	60.8	48.6
Shared LN (Cohen and Smith, 2009)	61.3	41.4
Feature DMV (Berg-Kirkpatrick et al., 2010)	63.0	-
PR-S (Gillenwater et al., 2010)	64.3	53.3
E-DMV (Headden III et al., 2009)	65.0	-
TSG-DMV (Blunsom and Cohn, 2010)	65.9	53.1
UR-A E-DMV (Tu and Honavar, 2012)	71.4	57.0
CRFAE (Cai et al., 2017)	71.7	55.7
Neural E-DMV(Jiang et al., 2016)	72.5	57.6
HDP-DEP (Naseem et al., 2010)	73.8	-
NVTP (Li et al., 2019)	54.7	37.8
L-EVG* (Headden III et al., 2009)	68.8	-
LexTSG-DMV* (Blunsom and Cohn, 2010)	67.7	55.7
L-NDMV* (Han et al., 2017)	75.1	59.5
variational variant D-NDMV	75.5	60.4
deterministic variant D-NDMV	**75.6**	**61.4**
Systems with Additional Training Data (for reference)		
CS (Spitkovsky et al., 2013)	72.0	64.4
MaxEnc* (Le and Zuidema, 2015)	73.2	65.8

Table 1: Comparison on WSJ. $*$: approaches with lexicalized information.

4.2 Results on English Penn Treebank

In Table 1, we compared our method with a large number of previous approaches to unsupervised dependency parsing. Both *variational variant* and *deterministic variant* outperform recent approaches in the basic setup, which demonstrates the benefit of utilizing contextual information in dependency strength prediction. *Deterministic variant* has a slightly better parsing accuracy than *variational variant* but *variational variant* is more stable. The standard derivations of *deterministic variant* and *variational variant* are 0.530 and 0.402 respectively for 5 runs.

4.3 Results on Universal Dependency Treebank

We compare our model with several state-of-the-art models on the UD Treebanks and report the results in Table 2.

We first compare our model with two generative models: NDMV and left corner DMV (LC-DMV) (Noji et al., 2016). The LC-DMV is the recent state-of-the-art generative approach on Universal Dependency Treebank. Our *variational variant* D-NDMV outperforms the LC-DMV and the NDMV on average.

Furthermore, we compare our model with current state-of-the-art discriminative models, the neural variational transition-based parser (NVTP) (Li et al., 2019) and Convex-MST (Grave and Elhadad, 2015). Note that current discriminative approaches usually rely on strong universal linguistic prior [1] to get better performance. So the comparisons may not be fair for our model. Despite this, we find that our model can achieve competitive accuracies compared with these approaches.

	NO UP				+ UP	
	NDMV	LD	DV	VV	NVTP	CM
Length $\leq$ 15						
Basque	**48.3**	47.9	40.6	42.7	52.9	52.5
Dutch	44.1	35.5	42.1	43.0	39.6	43.4
French	59.5	52.1	59.0	**61.7**	59.9	61.6
German	56.2	51.9	56.4	**58.5**	57.5	54.4
Italian	72.7	**73.1**	59.6	63.5	59.7	73.2
Polish	72.7	66.2	70.5	**75.8**	57.1	66.7
Portuguese	34.4	**70.5**	68.8	69.1	52.7	60.7
Spanish	38.1	65.5	63.8	**66.6**	55.6	61.6
Average	53.3	57.8	57.6	**60.1**	54.4	59.3
Length $\leq$ 40						
Basque	**47.8**	45.4	39.9	42.4	48.9	50.0
Dutch	35.6	34.1	42.4	**43.7**	42.5	45.3
French	38.1	48.6	57.2	**58.5**	55.4	62.0
German	50.4	50.5	**54.5**	52.9	54.2	51.4
Italian	63.6	**71.1**	60.2	61.3	55.7	69.1
Polish	62.8	63.7	66.7	**73.0**	51.7	63.4
Portuguese	49.0	**67.2**	64.7	65.7	45.3	57.1
Spanish	58.0	61.9	64.3	**64.4**	52.4	61.9
Average	50.7	55.3	56.2	**57.7**	50.8	57.5

Table 2: Comparison on Universal Dependency Treebank. No UP: Systems without universal linguistic prior. +UP: Systems with universal linguistic prior. LD: LC-DMV (Noji et al., 2016). DV: *deterministic variant* of D-NDMV. VV: *variational variant* of D-NDMV. NVTP: neural variational transition-based parser (Li et al., 2019). CM: Convex-MST.

4.4 Results on Datasets from PASCAL Challenge

We also perform experiments on the datasets from PASCAL Challenge (Gelling et al., 2012), which contains eight languages: Arabic, Basque, Czech, Danish, Dutch, Portuguese, Slovene and Swedish. We compare our approaches with NDMV (Jiang et al., 2016), Convex-MST (Grave and Elhadad, 2015) and CRFAE (Cai et al., 2017). NDMV and CRFAE are two state-of-the-art approaches on the PASCAL Challenge datasets. We show the directed dependency accuracy on the testing sentences no longer than 10 (Table 3) and on all the testing sentences (Table 4). It can be seen that on average our models outperform other state-of-the-

[1] Universal linguistic prior (UP) is a set of syntactic dependencies that are common in many languages (Naseem et al., 2010).

	Arabic	**Basque**	**Czech**	**Danish**	**Dutch**	**Portuguese**	**Slovene**	**Swedish**	**Average**
Approaches Without Using Universal Linguistic Prior									
E-DMV	38.4	41.5	45.5	52.4	37.0	40.9	35.2	52.6	42.9
Neural DMV	60.0	44.1	46.2	63.3	33.2	36.9	31.6	**48.3**	45.4
Convex-MST	55.2	29.4	36.5	49.3	35.5	**43.2**	27.5	30.2	38.3
CRF-AE	42.4	**45.8**	24.4	23.9	28.8	33.0	33.4	45.6	34.6
deterministic variant	54.4	44.8	55.2	58.9	**37.2**	40.1	**35.2**	50.3	47.0
variational variant	**60.0**	45.4	**59.1**	**63.6**	34.6	42.7	28.3	45.9	**47.5**
Approaches Using Universal Linguistic Prior									
Convex-MST	39.0	27.8	43.8	48.1	35.9	55.6	62.6	49.6	45.3
CRF-AE	39.2	33.9	45.1	44.5	42.2	61.9	41.9	66.0	46.8

Table 3: DDA on testing sentences no longer than 10 on eight additional languages from PASCAL Challenge.

	Arabic	**Basque**	**Czech**	**Danish**	**Dutch**	**Portuguese**	**Slovene**	**Swedish**	**Average**
Approaches Without Using Universal Linguistic Prior									
E-DMV	27.4	33.8	37.4	44.9	24.7	34.8	23.2	40.2	33.3
Neural DMV	30.9	37.7	38.1	53.3	22.9	30.7	19.9	33.9	33.4
Convex-MST	47.7	30.5	33.4	44.2	**28.3**	**35.9**	18.1	29.2	33.4
CRF-AE	29.9	39.1	20.3	18.6	17.8	32.6	**28.0**	37.0	27.9
deterministic variant	**38.2**	38.8	47.3	47.3	24.7	34.1	23.2	40.1	36.7
variational variant	33.9	**41.2**	**48.4**	**54.7**	25.3	35.8	28.1	**40.5**	**38.5**
Approaches Using Universal Linguistic Prior									
Convex-MST	34.2	24.9	39.0	36.3	35.2	46.0	51.7	39.6	38.3
CRF-AE	37.2	30.3	36.4	33.2	38.3	52.4	29.2	47.1	38.2

Table 4: DDA on all the testing sentences on eight additional languages from PASCAL Challenge.

art approaches including those utilizing the universal linguistic prior.

5 Analysis

In this section, we studies what information is captured in the sentence embeddings and the some configurations that are sensitive to our model. Here we use *deterministic variant* of D-NDMV to conduct the following analysis. *deterministic variant* of D-NDMV performs similar to *deterministic variant* of D-NDMV.

5.1 Rule Probabilities in Different Sentences

The motivation behind D-NDMV is to break the independence assumption and utilize global information in predicting grammar rule probabilities. Here we conduct a few case studies of what information is captured in the sentence embedding and how it influences grammar rule probabilities.

We train a D-NDMV on WSJ and extract all the embeddings of the training sentences. We then focus on the following two sentences: "What 's next" and "He has n't been able to replace the M'Bow cabal". We now examine the dependency rule probability of VBZ generating JJ to the right with valence 0 in these two sentences (illustrated in Figure 3). In the first sentence, this rule is used in the gold parse ("'s" is the head of "next"); but in the second sentence, this rule is not used in the gold parse (the head of "able" is "been" instead of "has"). We observe that the rule probability predicted by D-NDMV given the first sentence is indeed significantly larger than that given the second sentence, which demonstrates the positive impact of conditioning rule probability prediction on the sentence embedding.

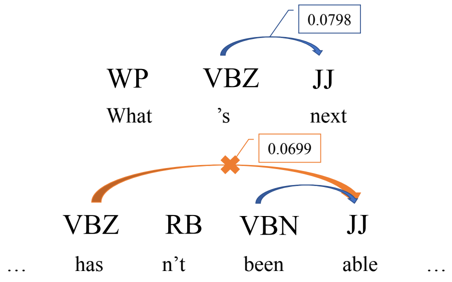

Figure 3: Rule probabilities predicted by D-NDMV given the two example sentences

To obtain a more holistic view of how rule probabilities change in different sentences, we collect the probabilities of a particular rule ("IN" generating "CD" to the right with valence 1) predicted by our model for all the sentences of WSJ. Figure 4 shows two distributions over the rule probability when the rule is used in the gold parse vs. when the rule is applicable to parsing the sentence but is not used in the gold parse. It can be seen that when

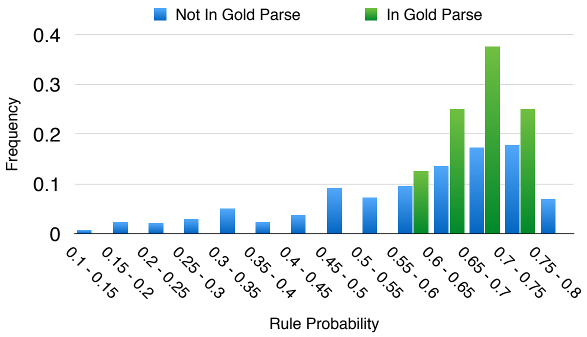

Figure 4: Comparison of the distributions over the rule probability when the rule appears vs. does not appear in the gold parse.

AVERAGE PROBABILITY	D-NDMV	E-DMV
All	0.107	0.094
In gold parse	0.253	0.219
Not in gold parse	0.097	0.085

Table 5: Comparison of the average probabilities in D-NDMV and E-DMV when the rule is used and not used in the gold parse.

the rule appears in the gold parse, its probability is clearly boosted in our model.

Finally, for every sentence of WSJ, we collect the probabilities predicted by our model for all the rules that are applicable to parsing the sentence. We then calculate the average probability 1) when the rule is used in the gold parse, 2) when the rule is not used in the gold parse, and 3) regardless of whether the rule is used in the gold parse or not. We use the E-DMV model as the baseline in which rule probabilities do not change with sentences. The results are shown in Table 5. We observe that compared with the E-DMV baseline, the rule probabilities predicted by our model are increased by 14.0% on average, probably because our model assigns higher probabilities to rules applicable to parsing the input sentence than to rules not applicable (e.g., if the head or child of the rule does not appear in the sentence). The increase of the average probability when the rule is used in the gold parse (15.7%) is higher than when the rule is not used in the gold parse (13.7%), which again demonstrates the advantage of our model.

5.2 Choice of Sentence Encoder

Besides LSTM, there are a few other methods of producing the sentence representation. Table 6 compares the experimental results of these methods. The bag-of-tags method simply computes the average of all the POS tag embeddings and has the lowest accuracy, showing that the word order is informative for sentence encoding in D-NDMV. The anchored words method replaces the POS tag embddings used in the neural network of the neural DMV with the corresponding hidden vectors produced by a LSTM on top of the input sentence, which leads to better accuracy than bag-of-tags but is still worse than LSTM. Replacing LSTM with Bi-LSTM or attention-based LSTM also does not lead to better performance, probably because these models are more powerful and hence more likely to result in degeneration and overfitting.

SENTENCE ENCODER	DDA
Bag-of-Tags Method	74.1
Anchored Words Method	75.1
LSTM	**75.9**
Attention-Based LSTM	75.5
Bi-LSTM	74.2

Table 6: Comparison of different sentence encoders in D-NDMV.

5.3 Impact of Genres

All the sentences in WSJ come from newswire, which conform to very similar syntactic styles. Here we study whether our method can capture different syntactic styles by learning our method from Chinese Treebank 9.0 (2005) which contains sentences of two different genres: the informal genre and the formal genre. The experimental setup is the same as that in section 4. We pick the rule of "CD" (number) generating "AD" (adverb) to the left with valence 0 and collect the rule probability in sentences from the two genres.In informal sentences our model assigns smaller probabilities to the rule than in formal sentences. This may reflect the fact that formal texts are more precise than informal text when presenting numbers, which is captured by our model[2].

5.4 Impact of Sentence Embedding Dimension

The dimension of sentence embeddings in our model is an important hyper-parameter. If the dimension is too large, the sentence embedding may capture too much information of the sentence and hence the model is very likely to degenerate or overfit as discussed in section 3.1. If the dimension is too small, the model loses the benefit of sentence information and becomes similar to neural DMV. As Figure 5 illustrates, dimension 10 leads to the best parsing accuracy, while dimen-

[2] More details can be found in the supplementary materials.

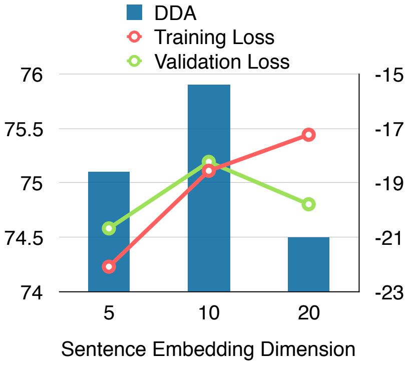

Figure 5: Impact of the sentence embedding dimension on both the testing set parsing accuracy and the average conditional log Viterbi likelihood (w.r.t. loss) of the training set and the validation set.

sion 20 produces lower parsing accuracy probably because of a combination of degeneration and overfitting. The conditional log Viterbi likelihood curves on the training set and the validation set in Figure 5 confirm that overfitting indeed occur with dimension 20.

6 Conclusion

We propose D-NDMV, a novel unsupervised parser with characteristics from both generative and discriminative approaches to unsupervised parsing. D-NDMV extends neural DMV by parsing a sentence using grammar rule probabilities that are computed based on global information of the sentence. In this way, D-NDMV breaks the context-free independence assumption in generative dependency grammars and is therefore more expressive. Our extensive experimental results show that our approach achieves competitive accuracy compared with state-of-the-art parsers.

Acknowledgments

This work was supported by the Major Program of Science and Technology Commission Shanghai Municipal (17JC1404102).

References

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590. Association for Computational Linguistics.

Phil Blunsom and Trevor Cohn. 2010. Unsupervised induction of tree substitution grammars for dependency parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1204–1213. Association for Computational Linguistics.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. *CoNLL 2016*, page 10.

Jiong Cai, Yong Jiang, and Kewei Tu. 2017. Crf autoencoder for unsupervised dependency parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1638–1643.

Glenn Carroll and Eugene Charniak. 1992. *Two experiments on learning probabilistic dependency grammars from corpora*. Department of Computer Science, Univ.

Shay B Cohen, Kevin Gimpel, and Noah A Smith. 2008. Logistic normal priors for unsupervised probabilistic grammar induction. In *Advances in Neural Information Processing Systems*, pages 321–328.

Shay B Cohen and Noah A Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 74–82. Association for Computational Linguistics.

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

Douwe Gelling, Trevor Cohn, Phil Blunsom, and Joao Graça. 2012. The pascal challenge on grammar induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 64–80. Association for Computational Linguistics.

Jennifer Gillenwater, Kuzman Ganchev, Joao Graça, Fernando Pereira, and Ben Taskar. 2010. Sparsity in dependency grammar induction. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 194–199. Association for Computational Linguistics.

Edouard Grave and Noémie Elhadad. 2015. A convex and feature-rich discriminative approach to dependency grammar induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1375–1384.

Wenjuan Han, Yong Jiang, and Kewei Tu. 2017. Dependency grammar induction with neural lexicalization and big training data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1683–1688.

William P Headden III, Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 101–109. Association for Computational Linguistics.

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

Yong Jiang, Wenjuan Han, and Kewei Tu. 2017. Combining generative and discriminative approaches to unsupervised dependency parsing via dual decomposition. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1689–1694, Copenhagen, Denmark. Association for Computational Linguistics.

Diederik P Kingma and Max Welling. 2014. Autoencoding variational bayes. In *ICLR*.

Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Phong Le and Willem Zuidema. 2015. Unsupervised dependency parsing: Let's use supervised parsers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 651–661.

Bowen Li, Jianpeng Cheng, Yang Liu, and Frank Keller. 2019. Dependency grammar induction with a neural variational transition-based parser. In *AAAI*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85.

Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1244. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Hiroshi Noji, Yusuke Miyao, and Mark Johnson. 2016. Using left-corner parsing to encode universal structural constraints in grammar induction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 33–43.

Valentin I Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2013. Breaking out of local optima with count transforms and model recombination: A study in grammar induction. In *EMNLP*, pages 1983–1995.

Valentin I Spitkovsky, Hiyan Alshawi, Daniel Jurafsky, and Christopher D Manning. 2010. Viterbi training improves unsupervised dependency parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 9–17. Association for Computational Linguistics.

Kewei Tu and Vasant Honavar. 2012. Unambiguity regularization for unsupervised learning of probabilistic grammars. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1324–1334. Association for Computational Linguistics.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207–238.

A Impact of Genres

All the sentences in WSJ come from newswire, which conform to very similar syntactic styles. Here we study whether our method can capture different syntactic styles by learning our method from Chinese Treebank 9.0 (2005) which contains sentences of two different genres: the informal genre (chat messages and transcribed conversational telephone speech) and the formal genre (newswire, broadcast and so on). The experimental setup is the same as that in section 4.

We extract the embeddings of the training sentences from the learned model and map them onto a 3D space via the t-SNE algorithm (Van der Maaten and Hinton, 2008) (Figure 6). It can be seen that although the two types of sentences are mixed together overall, many regions are clearly dominated by one type or the other. This verifies that sentence embeddings learned by our approach can capture some genre information.

We pick the rule of "CD" (number) generating "AD" (adverb) to the left with valence 0 and illustrate the distributions of the rule probability in sentences from the two genres in Figure 7. It can be seen that in informal sentences our model assigns

What 's next WP VBZ JJ	**He has n't been able to replace the M'Bow cabal** PRP VBZ RB VBN JJ TO VB DT NNP NN
The government is nervous. DT NN VBZ JJ.	I was shaking the whole time. PRP VBD VBG DT JJ NN.
Both were right. DT VBD JJ.	But says Mr. Bock It was a close call. CC VBZ NNP NNP PRP VBD DT JJ NN.
That is n't easy. DT VBZ RB JJ.	Then there 'll be another swing. RB EX MD VB DT NN.
The IRA portion of the Packwood-Roth plan is irresponsible. DT NNP NN IN DT NNP NN VBZ JJ.	He 's totally geared to a punitive position. PRP VBZ RB VBN TO DT JJ NN.
These figures are n't seasonally adjusted. DT NNS VBP RB RB JJ.	Her sister Cynthia wishes Toni had a different job. PRP$ NN NNP VBZ NNP VBD DT JJ NN.

Table 7: Sentences closest to the two example sentences in terms of the L2 distance between their learned embeddings. Both the word sequence and the POS tag sequence are shown for each sentence.

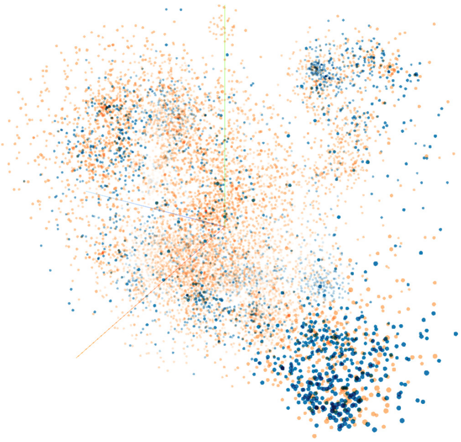

Figure 6: 3D visualization of the learned sentence embeddings from CTB. Orange dots denote informal sentences and blue dots denote formal sentences.

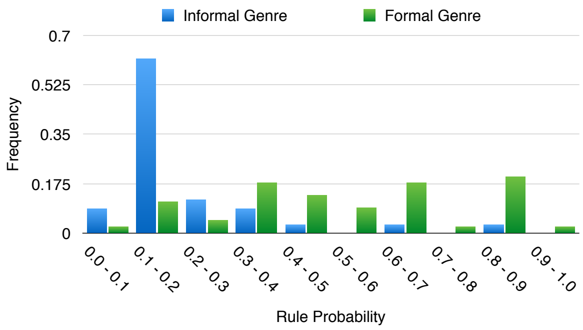

Figure 7: Comparison of the distributions over the rule probability in sentences from the two genres.

smaller probabilities to the rule than in formal sentences. This may reflect the fact that formal texts are more precise than informal text when presenting numbers, which is captured by our model.

B Nearby Sentences in Embedding Space

We train a our method on WSJ and extract all the embeddings of the training sentences. We then focus on the following two sentences: "What 's next" and "He has n't been able to replace the M'Bow cabal".

Table 7 shows the two sentences as well as a few other sentences closest to them measured by the L2 distance between their embeddings. It can be seen that most sentences close to the first sentence contain a copula followed by a predicative adjective, while most sentences close to the second sentence end with a noun phrase where the noun has a preceding modifier. These two examples show that the sentence embeddings learned by our approach encode syntactic information that can be useful in parsing.