Supplementary Material

Derivation of the M-step for Revised EM

Original Score Function Under the posterior regularized EM framework, the original score function without a baseline should be defined as:

$$\log \frac{p_\theta(x,a)}{q(a)} = \log \frac{p_\theta(x,a)}{q_\omega(a|x)\gamma(a,x)}$$

But in practical training, we observed that $\gamma(a,x)$ will assign large weights (larger than 1) to more likely parse trees and small weights (less than 1) to less likely parse trees. Thus $-\log\gamma(a,x)$ would effectively reverse the direction of optimization, which could dramatically mislead the learning process. To cope with this issue, we simply define the score function for our revised EM algorithm as Eq. (7). We will show how this definition will affect the loss function.

$$\begin{aligned}
&\mathbb{E}_{q(a)}[\log\gamma(x,a)] \\
&= \sum_a q(a)\log\gamma(x,a) \\
&= \sum_a q_\omega(a|x)\gamma(x,a)\log\gamma(x,a) \\
&\leqslant \sum_a q_\omega(a|x)\gamma(x,a)\gamma(x,a) \\
&= \mathbb{E}_{q_\omega(a|x)}[\gamma^2(x,a)] \\
&= \mathrm{Var}_{q_\omega(a|x)}[\gamma(x,a)] + \mathbb{E}_{q_\omega(a|x)}[\gamma(x,a)]^2
\end{aligned}$$

Since

$$\mathbb{E}_{q_\omega(a|x)}[\gamma(x,a)] = \sum_a q_\omega(a|x)\gamma(x,a) = 1$$

we have

$$\begin{aligned}
&\mathbb{E}_{q(a)}[\log\frac{p_\theta(x,a)}{q_\omega(a|x)}] \\
&= \mathbb{E}_{q(a)}[\log\frac{p_\theta(x,a)}{q_\omega(a|x)\gamma(a,x)} + \log\gamma(a,x)] \\
&= \mathbb{E}_{q(a)}[\log\frac{p_\theta(x,a)}{q(a)}] + \mathbb{E}_{q(a)}[\log\gamma(a,x)] \\
&\leqslant \mathcal{L}_x + \mathrm{Var}_{q_\omega(a|x)}[\gamma(x,a)] + 1.
\end{aligned}$$

Thus, in theory, $\mathrm{Var}_{q_\omega(a|x)}[\gamma(x,a)]$ can be viewed as a regularization for posterior regularization.

Parameter Updating In the revised EM algorithm, the parameters of both encoder and decoder should be updated under the distribution of $q(a)$ rather than $p_\omega(a|x)$. Since $q(a) = \gamma(x,a)p_\omega(a|x)$, the gradient for the parameter of the encoder via MC sampling will be:

$$\begin{aligned}
\frac{\partial\mathcal{L}_x}{\partial\omega} &= \frac{1}{M}\sum_m \gamma(x,a^{(m)})l(x,a^{(m)})\frac{\partial\log q(a^{(m)})}{\partial\omega} \\
&= \frac{1}{M}\sum_m \gamma(x,a^{(m)})l(x,a^{(m)})\frac{\partial\log p_\omega(a^{(m)}|x)}{\partial\omega}
\end{aligned}$$

Algorithm 1: Pretraining for Variational Inference Dependency Parser.

Parameters: $\omega, \theta, \boldsymbol{\lambda}, \varepsilon, ||\cdot||_\beta, M$
Constrained Feature Functions: $\boldsymbol{\phi}(x,a)$
Initialization;
while *not converged* **do**
  Sample $a^{(m)} \sim q_\omega(a|x), 1 \leqslant m \leqslant M$;
  PR Computation:
  $Z(\boldsymbol{\lambda}) \approx \frac{1}{M}\sum_m \exp(-\boldsymbol{\lambda}^T\boldsymbol{\phi}(x,a^{(m)}))$,
  $\gamma(x,a^{(m)}) = \frac{1}{Z(\boldsymbol{\lambda})}\exp(-\boldsymbol{\lambda}^T\boldsymbol{\phi}(x,a^{(m)}))$;
  Update parameters in mini-batch:
  Update θ w.r.t. its gradient
  $\frac{1}{M}\sum_m \gamma(x,a^{(m)})\frac{\partial\log p_\theta(x,a^{(m)})}{\partial\theta}$,
  Update ω w.r.t. its gradient
  $\frac{1}{M}\sum_m \gamma(x,a^{(m)})\frac{\partial\log q_\omega(a|x)}{\partial\omega}$,
  Update $\boldsymbol{\lambda}$ to optimize
  $\max_{\boldsymbol{\lambda}\geqslant 0} \quad -\boldsymbol{b}^T\boldsymbol{\lambda} - \log Z(\boldsymbol{\lambda}) - \varepsilon||\boldsymbol{\lambda}||_{\beta^*}$ (with projected gradient descent algorithm).
end

For the decoder, to boost the performance, we also use the score function for optimization:

$$\begin{aligned}
&\frac{\partial\mathcal{L}_x}{\partial\theta} \\
&= \frac{1}{M}\sum_m \gamma(x,a^{(m)})l(x,a^{(m)})\frac{\partial\log p_\theta(x,a^{(m)})}{\partial\theta}
\end{aligned}$$

Algorithm 2: Revised EM Algorithm for Variational Inference Dependency Parser.

Parameters: $\omega, \theta, \boldsymbol{\lambda}, \varepsilon, ||\cdot||_\beta, M$
Constrained Feature Functions: $\boldsymbol{\phi}(x,a)$
Critic: $l_{\text{CRITIC}}(x,a)$
Initialization;
while *not converged* **do**
  E′-step:
  Sample $a^{(m)} \sim q_\omega(a|x), 1 \leqslant m \leqslant M$;
  PR Computation:
  $Z(\boldsymbol{\lambda}) \approx \frac{1}{M}\sum_m \exp(-\boldsymbol{\lambda}^T\boldsymbol{\phi}(x,a^{(m)}))$,
  $\gamma(x,a^{(m)}) = \frac{1}{Z(\boldsymbol{\lambda})}\exp(-\boldsymbol{\lambda}^T\boldsymbol{\phi}(x,a^{(m)}))$,
  $q(a^{(m)}) = q_\omega(a^{(m)}|x)\gamma(x,a^{(m)})$;
  Compute $l_{\text{CRITIC}}(x,a)$ for specific critic;
  M-step:
  Update parameters in mini-batch:
  Update θ w.r.t. its gradient
  $\frac{1}{M}\sum_m \gamma(x,a^{(m)})\frac{\partial\log p_\theta(x,a^{(m)})}{\partial\theta}$,
  Update ω w.r.t. its gradient
  $\frac{1}{M}\sum_m \gamma(x,a^{(m)})\frac{\partial\log q_\omega(a|x)}{\partial\omega}$,
  Update $\boldsymbol{\lambda}$ to optimize
  $\max_{\boldsymbol{\lambda}\geqslant 0} \quad -\boldsymbol{b}^T\boldsymbol{\lambda} - \log Z(\boldsymbol{\lambda}) - \varepsilon||\boldsymbol{\lambda}||_{\beta^*}$ (with projected gradient descent algorithm).
end

Linguistic Rules

Table 1 and 2 present the universal linguistic rules used for WSJ and the Universal Dependency Treebank respectively.

Root → Auxiliary	Noun → Adjective
Root → Verb	Noun → Article
Verb → Noun	Noun → Noun
Verb → Pronoun	Noun → Numeral
Verb → Adverb	Preposition → Noun
Verb → Verb	Adjective → Adverb
Auxiliary → Verb	

Table 1: Universal dependency rules for WSJ (Naseem et al., 2010).

ROOT → VERB	NOUN → ADJ
ROOT → NOUN	NOUN → DET
VERB → NOUN	NOUN → NOUN
VERB → ADV	NOUN → NUM
VERB → VERB	NOUN → CONJ
VERB → AUX	NOUN → ADP
ADJ → ADV	

Table 2: Universal dependency rules for the Universal Dependency Treebank Noji, Miyao, and Johnson (2016).

Experimental Details

Algorithm 1 and 2 provide the outlines of our pretraining and revised EM algorithms respectively. To avoid starting training from arbitrarily initialized encoder and decoder, we pretrain our encoder and decoder separately via posterior regularization, where $\gamma(x, a)$ serves the reward for the REINFORCE.

Model Configuration

Encoder and Decoder We follow the model configuration in Cheng, Lopez, and Lapata (2017) to build the encoder and decoder by using Stack-LSTMs (Dyer et al., 2015). The differences are (1) we use neither the parent non-terminal embedding nor the action history embedding for both the decoder and encoder; (2) we do not use the adaptive buffer embedding for the encoder.

RL-BL For the baseline ($b(x) + b$) in RL-BL, we first pretrain a LSTM language model. We use word embeddings of size 100, 2 layer LSTM with 100 hidden size, and tie weights of output classifiers and word embeddings. During training the RL-BL, we fix the LSTM language model and rescale and shift the output $\log p(x)$ to fit the ELBO of the given sentence as

$$b(x) + b = \alpha \log p(x) + \tau$$

Hyper-Parameters and Optimization In all experiments, both PoS tag and word embeddings are used in our lexicalized models while only PoS tag embeddings are used in our unlexicalized models. Table 3 presents the hyperparameter settings. For pretrained word embeddings, we project them into lower dimension (word embedding dimensions). We select ε in Eq. (8) via a grid search on WSJ-10 development set. And we use Glorot for parameter initialization, and Adagrad for optimization except for posterior regularization.

word embeddings dimensions	80
PoS embeddings dimensions	80
Encoder LSTM dimensions	64
Decoder LSTM dimensions	64
LSTM layer	1
Encoder dropout	0.5
Decoder dropout	0.5
Learning rate	0.01
gradient clip	0.25
gradient clip (for pretraining)	0.5
ℓ_2 regularization	1e-4
ε in Eq. (8)	0.1
number of MC samples	20

Table 3: Hyperparameters.

References

Cheng, J.; Lopez, A.; and Lapata, M. 2017. A generative parser with a discriminative recognition algorithm. In *Proceedings of the ACL Conference*.

Dyer, C.; Ballesteros, M.; Ling, W.; Matthews, A.; and Smith, N. A. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the ACL Conference*.

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

Noji, H.; Miyao, Y.; and Johnson, M. 2016. Using left-corner parsing to encode universal structural constraints in grammar induction. In *Proceedings of the EMNLP Conference*.