Semi-Supervised Dependency Parsing with Variational Autoencoder

Wang Ge

2019/3/20

Why is Semi-Supervised Learning Necessary

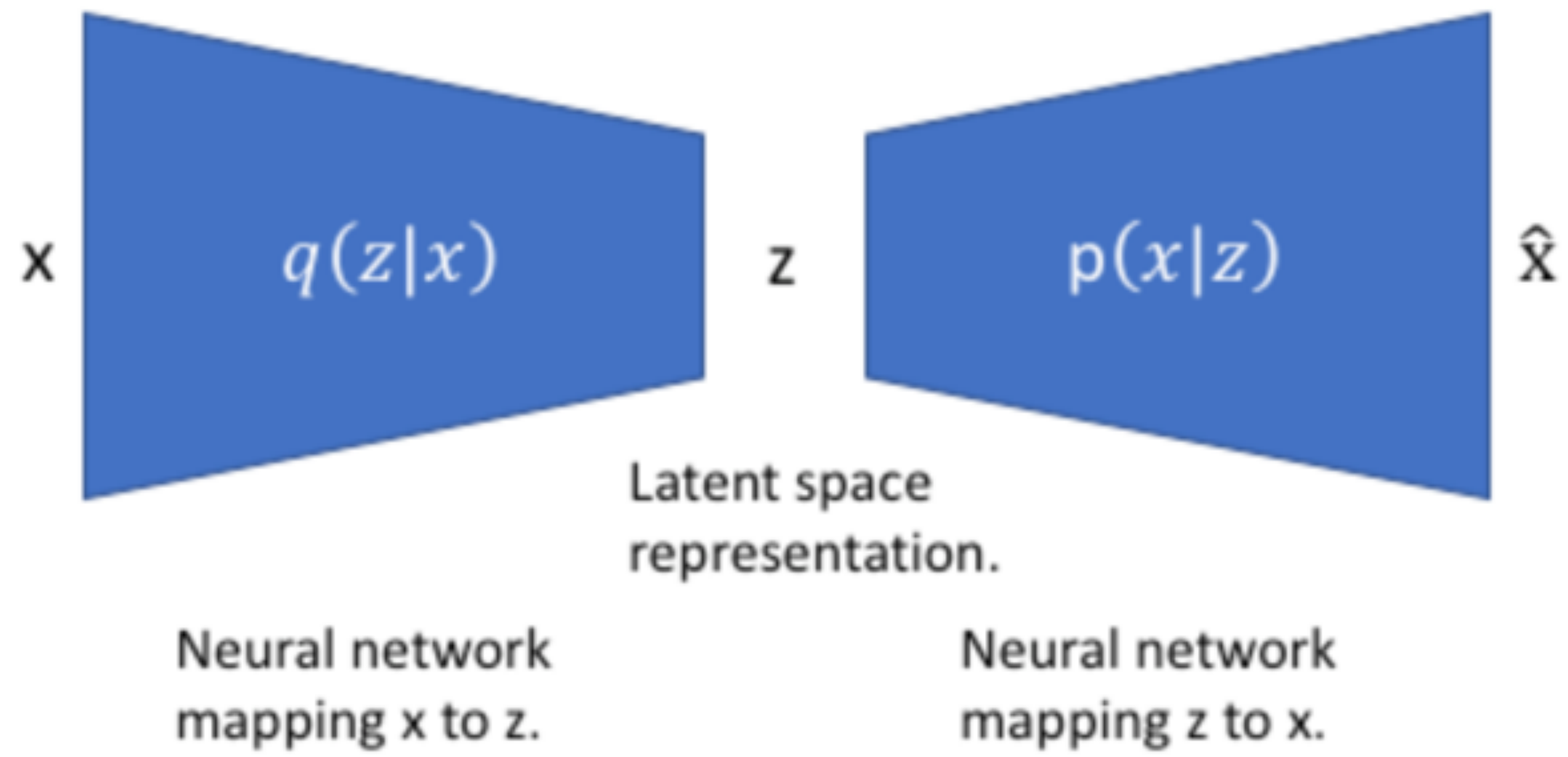
- Learning a reliable dependency parser requires a large amount of labelled data, e.g. Penn Treebank (39000+ sentence for training).

- Labelling data with dependency parse trees is a challenging and laborious work. For many languages, enough labelled data are not available.

- Unsupervised methods require POS tag information, lacking robustness over different annotation criteria such as WSJ and UD

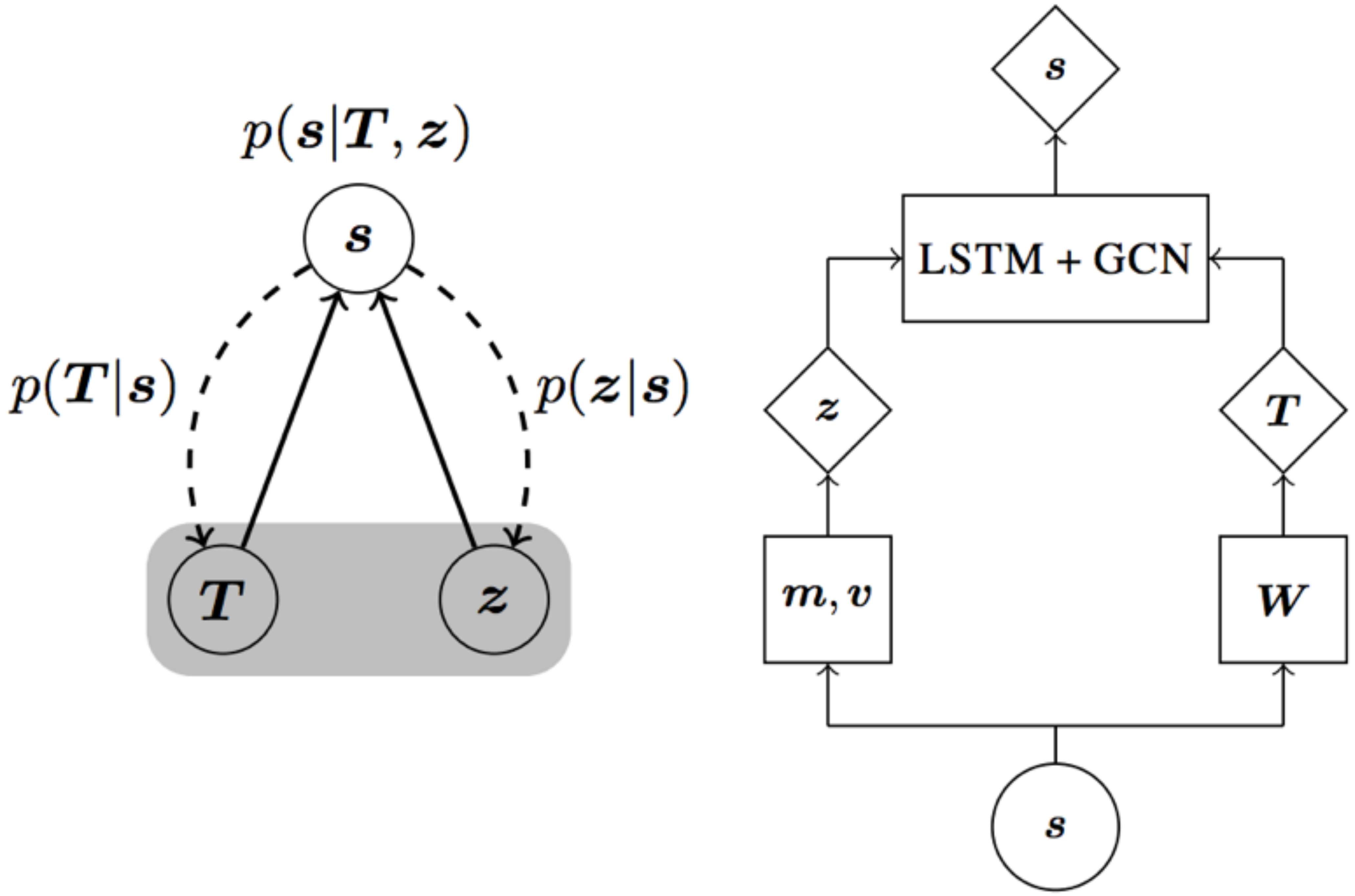
Variational Autoencoder

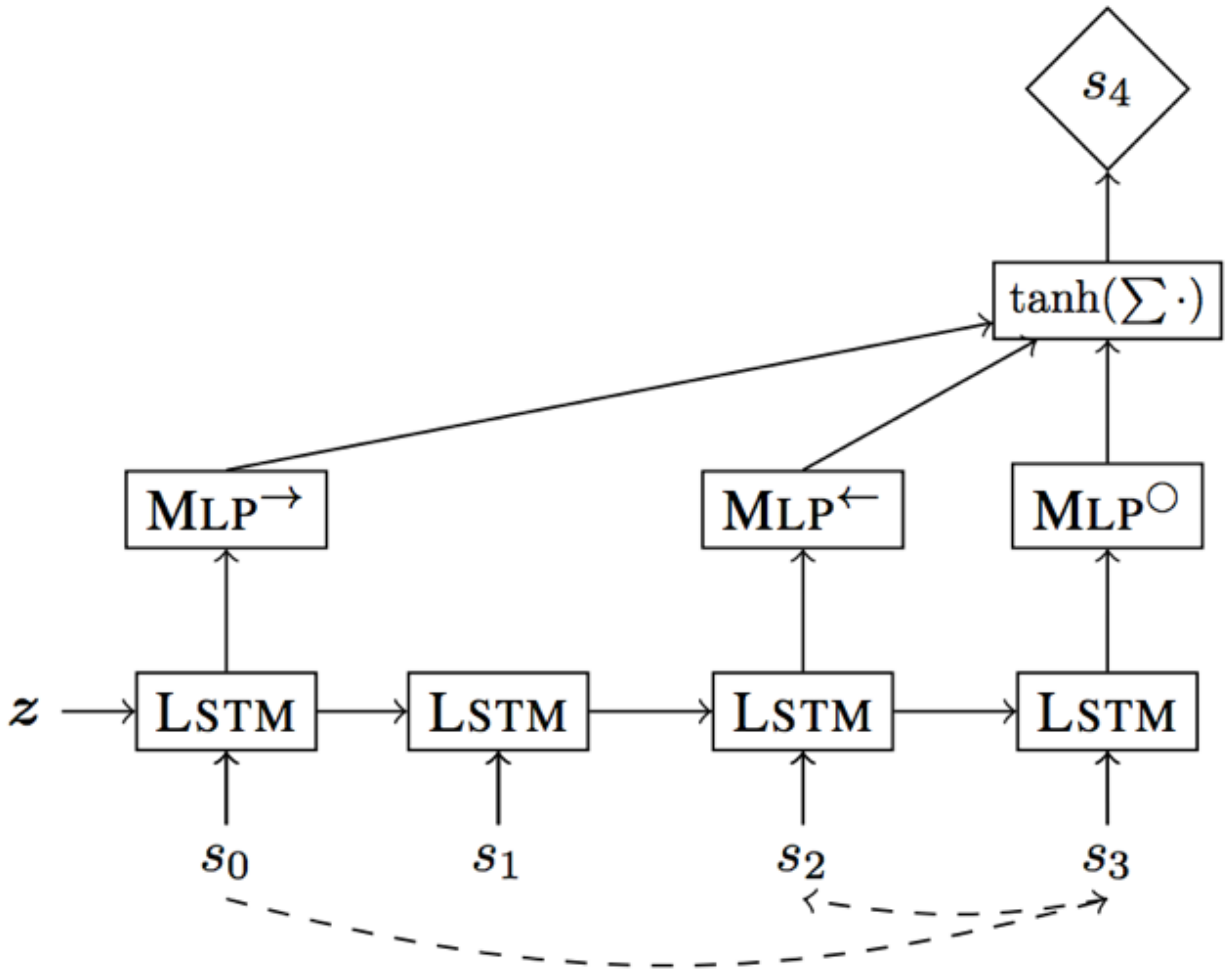
- An elegant generative model with latent variables



- Easy to be applied to semi-supervised learning, not suitable for structured latent variable like parse tree

Recent Work Combining Semi-Supervised Parsing with VAE





Graphical Model and Computational Graph

Decoder Implementation

DIFFERENTIABLE PERTURB-AND-PARSE: SEMI-SUPERVISED PARSING WITH STRUCTURED VARIATIONAL AUTOENCODER ICLR2019

Model Detail

- Generative Story

$$\boldsymbol{T} \sim p(\boldsymbol{T}|n) \qquad \boldsymbol{z} \sim p(\boldsymbol{z}|n) \qquad \boldsymbol{s} \sim p(\boldsymbol{s}|\boldsymbol{T}, \boldsymbol{z}, n)$$

- Objective Function

$$\log p_\theta(\boldsymbol{s}) = \mathbb{E}_{q_\phi(\boldsymbol{T},\boldsymbol{z}|\boldsymbol{s})}[\log p_\theta(\boldsymbol{s}|\boldsymbol{T}, \boldsymbol{z})] - \mathrm{KL}[q_\phi(\boldsymbol{T}, \boldsymbol{z}|\boldsymbol{s})|p(\boldsymbol{T}, \boldsymbol{z})] + \mathrm{KL}\left[q_\phi(\boldsymbol{T}, \boldsymbol{z}|\boldsymbol{s})\|p_\theta(\boldsymbol{T}, \boldsymbol{z}|\boldsymbol{s})\right]$$

$$\log p_\theta(\boldsymbol{s}) \geq \mathbb{E}_{q_\phi(\boldsymbol{T},\boldsymbol{z}|\boldsymbol{s})}[\log p_\theta(\boldsymbol{s}|\boldsymbol{T}, \boldsymbol{z})] - \mathrm{KL}[q_\phi(\boldsymbol{T}, \boldsymbol{z}|\boldsymbol{s})|p(\boldsymbol{T}, \boldsymbol{z})] = \tilde{\mathcal{E}}_{\theta,\phi}(\boldsymbol{s})$$

$$\bar{\mathcal{E}}_{\theta,\phi}(\boldsymbol{s}, \boldsymbol{T}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{s})}[\log p_\theta(\boldsymbol{s}|\boldsymbol{T}, \boldsymbol{z})] - \mathrm{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{s})|p(\boldsymbol{z})]$$

$$\bar{\mathcal{E}}_{\theta,\phi}(\boldsymbol{s}, \boldsymbol{T}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{s})}[\log p_\theta(\boldsymbol{s}|\boldsymbol{T}, \boldsymbol{z})] - \mathrm{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{s})|p(\boldsymbol{z})]$$

Encoder and Decoder

- For the encoder:

$$\boldsymbol{W} = \textsc{DepWeights}(\boldsymbol{s})$$

$$q_\phi(\boldsymbol{T}|\boldsymbol{s}) = \frac{\exp(\sum_{i,j} W_{i,j} T_{i,j})}{\sum_{\boldsymbol{T}'} \exp(\sum_{i,j} W_{i,j} T'_{i,j})}$$

$$\boldsymbol{m}, \log \boldsymbol{v}^2 = \textsc{EmbParams}(\boldsymbol{s})$$

$$q_\phi(\boldsymbol{z}|\boldsymbol{s}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{m}, \boldsymbol{v})$$

- For the decoder:

$$\boldsymbol{g}^i = \tanh\left(\textsc{Mlp}^{\bigcirc}(\boldsymbol{o}^i) + \sum_{h=0}^{i-1} T_{h,i} \times \textsc{Mlp}^{\curvearrowright}(\boldsymbol{o}^h) + \sum_{m=0}^{i-1} T_{i,m} \times \textsc{Mlp}^{\curvearrowleft}(\boldsymbol{o}^m)\right)$$

Differentiable Perturb-and-Parse

- Sampling dependency tree by perturbing weight matrix

$$\begin{aligned} \boldsymbol{W} &= \textsc{EmbParams}(\boldsymbol{s}) \\ \boldsymbol{P} &\sim \mathcal{G}(0,1) \\ \boldsymbol{T} &= \textsc{Eisner}(\boldsymbol{W}+\boldsymbol{P}) \end{aligned}$$

where the perturbation is sampled from Gumbel distribution

- Sampling dependency tree by perturbing weight matrix

Replace the one-hot-argmax

$$o_i = \mathbb{1}[\forall 1 \leq j \leq k, j \neq i : v_i > v_j]$$

with

peaked-softmax

$$o_i = \frac{\exp(1/\tau \; v_i)}{\sum_{1 \leq j \leq k} \exp(1/\tau \; v_j)}$$

Experiment Results

(a) Parsing results

	English	French	Swedish
Supervised	88.79 / 84.74	84.09 / 77.58	86.59 / 78.95
VAE w. z	89.39 / 85.44	84.43 / 77.89	86.92 / 80.01
VAE w/o z	89.50 / 85.48	84.69 / 78.49	86.97 / 79.80
Kipperwasser & Goldberg	89.88 / 86.49	84.30 / 77.83	86.93 / 80.12

(b) Dependency length analysis

Distance	**Supervised Re / Pr**	**Semi-sup. Re / Pr**
(to root)	93.46 / **89.30**	93.84 / **92.41**
1	95.61 / 94.07	95.33 / 94.57
2	93.01 / 90.88	92.50 / 92.09
3 . . . 6	85.95 / 88.13	87.31 / 87.93
$>$ **7**	**72.47** / 83.26	**78.72** / 83.11

(c) Dependency label analysis

Label	**Supervised Re / Pr**	**Semi-sup. Re / Pr**
mwe	75.58 / 81.25	90.70 / 84.78
advmod	87.27 / 85.95	87.32 / 87.51
appos	77.49 / 80.27	81.39 / 81.03