Supplementary Materials for Segment-Level Sequence Modeling using Gated Recursive Semi-Markov Conditional Random Fields

Jingwei Zhuo[1,2] *, Yong Cao[2], Jun Zhu[1] †, Bo Zhang[1], Zaiqing Nie[2]
[1]Dept. of Comp. Sci. & Tech., State Key Lab of Intell. Tech. & Sys., TNList Lab,
Tsinghua University, Beijing, 100084, China
[2]Microsoft Research, Beijing, 100084, China
{zjw15@mails, dcszj@mail, dcszb@mail}.tsinghua.edu.cn; {yongc, znie}@microsoft.com

1 Training and Inference of Semi-CRFs

In this section, we show more details about the training and inference of Semi-CRFs following the settings we made in the main paper.

1.1 Training of Semi-CRF-based Parameters

Given training data, all the parameters of grSemi-CRFs can be learnt by maximizing log likelihood, i.e., $\mathcal{L} = \log p(\mathbf{s}|\mathbf{x})$. To simplify representations, we introduce some auxiliary notations, including $g(h_j, d_j, y_{j-1}, y_j) = F(s_j, \mathbf{x}) + A(y_{j-1}, y_j)$ and $G(\mathbf{s}, \mathbf{x}) = \sum_{j=1}^{|\mathbf{s}|} g(h_j, d_j, y_{j-1}, y_j)$. Then the likelihood can be rewritten as $p(\mathbf{s}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp(G(\mathbf{s}, \mathbf{x}))$ where the normalization factor $Z(\mathbf{x}) = \sum_{\mathbf{s}'} \exp(G(\mathbf{s}', \mathbf{x}))$.

We further define

$$\alpha_{y,t} = \log \sum_{\mathbf{s}' \in \mathbf{s}_{1:t,y}} \exp(G(\mathbf{s}', \mathbf{x})), \tag{1}$$

where $\mathbf{s}_{1:k,y}$ denotes all segmentations for $(x_1, ..., x_k)$ with y being the tag of the ending segment. And we also define

$$\beta_{y,k} = \log \sum_{\mathbf{s}' \in \mathbf{s}_{k+1:T,y}} \exp(G(\mathbf{s}', \mathbf{x})), \tag{2}$$

where $\mathbf{s}_{k:T,y}$ denotes all segmentations for $(x_{k+1}, ..., x_T)$ with y being the tag of the segment which contains x_k.

Then, by using a Semi-CRF version of forward-backward algorithms, we can compute $\alpha_{y,k}$ and $\beta_{y,k}$ iteratively, i.e.,

$$\alpha_{y,k} = \log \sum_{d=1}^{L} \sum_{y' \in \mathcal{Y}} \exp\left(\alpha_{y',k-d} + g(k-d+1, d, y', y)\right), \tag{3}$$

$$\beta_{y,k} = \log \sum_{d=1}^{L} \sum_{y' \in \mathcal{Y}} \exp\left(\beta_{y',k+d} + g(k+1, d, y, y')\right), \tag{4}$$

where the boundary conditions are setted as $\alpha_{y,k} = 0$ for $k \leq 0$ and $\beta_{y,k} = 0$ for $k \geq T$.

Then, the normalization factor $Z(\mathbf{x})$ can be denoted as

$$Z(\mathbf{x}) = \sum_{y \in \mathcal{Y}} \exp(\alpha_{y,k}), \tag{5}$$

and corresponding partial derivative is

$$\frac{\partial Z(\mathbf{x})}{\partial g(k, d, y', y)} = \frac{1}{Z(\mathbf{x})} \exp(\alpha_{y',k-d} + g(k, d, y', y) + \beta_{y,d}). \tag{6}$$

Thus, the derivative of the objective function is

$$\frac{\partial \mathcal{L}}{\partial g(k, d, y', y)} = \sum_{j=1}^{|\mathbf{s}|} \mathbb{I}(s_j = \langle k, d, y \rangle, y_{j-1} = y') - \frac{\partial Z(\mathbf{x})}{\partial g(k, d, y', y)}, \tag{7}$$

where $\mathbb{I}(\cdot)$ is the indicator function[1].

Then, we can easily compute gradients for Semi-CRF-based parameters, i.e.,

$$\frac{\partial \mathcal{L}}{\partial A(y', y)} = \sum_{d=1}^{L} \sum_{k=d}^{T} \frac{\partial \mathcal{L}}{\partial g(k, d, y', y)}, \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial [\mathrm{V}_0]_{y,j}} = \sum_{d=1}^{L} \sum_{k=d}^{T} \sum_{y' \in \mathcal{Y}} \frac{\partial \mathcal{L}}{\partial g(k, d, y', y)} z_{k,j}^{(d)}, \tag{9}$$

and

$$\left[\frac{\partial \mathcal{L}}{\partial F(\mathbf{s}_k^{(d)})}\right]_y = \sum_{y' \in \mathcal{Y}} \frac{\partial \mathcal{L}}{\partial g(k, d, y', y)}. \tag{10}$$

where $\left[\frac{\partial \mathcal{L}}{\partial F(\mathbf{s}_k^{(d)})}\right]_y$ is the yth entry of the length-$|\mathcal{Y}|$ vector $\frac{\partial \mathcal{L}}{\partial F(\mathbf{s}_k^{(d)})}$.

1.2 Training of grConv Parameters

Thanks to the recursive structure, the backpropagated gradients follow

$$\frac{\partial \mathcal{L}}{\partial \mathbf{z}_k^{(d)}} = \frac{\partial \mathbf{z}_k^{(d+1)}}{\partial \mathbf{z}_k^{(d)}} \frac{\partial \mathcal{L}}{\partial \mathbf{z}_k^{(d+1)}} + \frac{\partial \mathbf{z}_{k-1}^{(d+1)}}{\partial \mathbf{z}_k^{(d)}} \frac{\partial \mathcal{L}}{\partial \mathbf{z}_{k-1}^{(d+1)}} + \mathrm{V}_0^{(d)\mathrm{T}} \frac{\partial \mathcal{L}}{\partial F(\mathbf{s}_k^{(d)}, \mathbf{x})}, \tag{11}$$

---

* This work was done when J.W.Z was on an internship with Microsoft Research.

† J.Z is the corresponding author.

[1] $\mathbb{I}(E) = 1$ when condition $E =$ *true* and $\mathbb{I}(E) = 0$ when condition $E =$ *false*.

where

$$\frac{\partial \mathbf{z}_k^{(d+1)}}{\partial \mathbf{z}_k^{(d)}} = \mathrm{diag}(\boldsymbol{\theta}_L) + \mathrm{diag}(\boldsymbol{\theta}_M \circ g'(\boldsymbol{\alpha}_k^{(d+1)}))\mathrm{W}_L,$$

$$\frac{\partial \mathbf{z}_{k-1}^{(d+1)}}{\partial \mathbf{z}_k^{(d)}} = \mathrm{diag}(\boldsymbol{\theta}_R) + \mathrm{diag}(\boldsymbol{\theta}_M \circ g'(\boldsymbol{\alpha}_{k-1}^{(d+1)}))\mathrm{W}_R, \tag{12}$$

and $\frac{\partial \mathcal{L}}{\partial F(\mathbf{s}_k^{(d)},\mathbf{x})}$ is computed in Eq. (10).

Embeddings can be learnt using $\frac{\partial \mathcal{L}}{\partial \mathbf{z}_k^{(0)}}$ as grSemi-CRFs use embeddings as length-1 segment-level features directly.

For W_L, we can compute the local partial derivative first, i.e.,

$$\left[\frac{\partial \mathbf{z}_k^{(d)}}{\partial \mathrm{W}_L}\right]_{i,j} = \theta_{M,i} g'(\alpha_{k,i}^{(d)}) z_{k,j}^{(d-1)}. \tag{13}$$

Thus we have

$$\frac{\partial \mathcal{L}}{\partial \mathrm{W}_L} = \sum_{d=1}^{L} \sum_{k=1}^{T-d+1} \left[\boldsymbol{\theta}_M \circ g'(\boldsymbol{\alpha}_k^{(d)}) \circ \frac{\partial \mathcal{L}}{\partial \mathbf{z}_k^{(d)}}\right] \mathbf{z}_k^{(d-1)\mathrm{T}}. \tag{14}$$

The gradients for W_R and $\mathbf{b}_\mathrm{W}$ can be computed in almost the same ways.

For G_L, the local partial derivative can be denoted as

$$\left[\frac{\partial \mathbf{z}_k^{(d)}}{\partial \mathrm{G}_L}\right]_{D\times\ell+i,j} = z_{k,i}^{(d-1)} \left[\frac{\partial \boldsymbol{\theta}_L}{\partial \mathrm{G}_L}\right]_{D\times\ell+i,j} + z_{k+1,i}^{(d-1)} \left[\frac{\partial \boldsymbol{\theta}_R}{\partial \mathrm{G}_L}\right]_{D\times\ell+i,j} + \hat{z}_{k,i}^{(d)} \left[\frac{\partial \boldsymbol{\theta}_M}{\partial \mathrm{G}_L}\right]_{D\times\ell+i,j}. \tag{15}$$

Notice that $\mathrm{G}_L \in \mathbb{R}^{3D\times D}$ has $3D$ rows where $\theta_{L,i}, \theta_{R,i}$ and $\theta_{M,i}$ corresponds to the ith, $(D+i)$th, and $(2D+i)$th rows of G_L. With a little abuse of notations (i.e., we use ℓ to denote numbers $0, 1, 2$ corresponding to rows of G_L, and characters L, R, M corresponding to the gating coefficients),

$$\left[\frac{\partial \boldsymbol{\theta}_L}{\partial \mathrm{G}_L}\right]_{D\times\ell+i,j} = \theta_{L,i} z_{k,j}^{(d-1)} \left(\mathbb{I}(\ell = L) - \theta_{\ell,i}\right),$$

$$\left[\frac{\partial \boldsymbol{\theta}_R}{\partial \mathrm{G}_L}\right]_{D\times\ell+i,j} = \theta_{R,i} z_{k,j}^{(d-1)} \left(\mathbb{I}(\ell = R) - \theta_{\ell,i}\right),$$

$$\left[\frac{\partial \boldsymbol{\theta}_M}{\partial \mathrm{G}_L}\right]_{D\times\ell+i,j} = \theta_{M,i} z_{k,j}^{(d-1)} \left(\mathbb{I}(\ell = M) - \theta_{\ell,i}\right). \tag{16}$$

Finally, we have,

$$\left[\frac{\partial \mathcal{L}}{\partial \mathrm{G}_L}\right]_{D\times\ell+i,j} = \sum_{d=1}^{L} \sum_{k=1}^{T-d+1} \frac{\partial \mathcal{L}}{\partial z_{k,i}^{(d)}} \left[\frac{\partial \mathbf{z}_k^{(d)}}{\partial \mathrm{G}_L}\right]_{D\times\ell+i,j}. \tag{17}$$

The gradients for G_R and $\mathbf{b}_\mathrm{G}$ can be computed in a similar way.

1.3 Inference of grSemi-CRFs

The inference problem is, given parameters and $\mathbf{x}$, find the best tag segmentation $\mathbf{s}^* = \mathrm{argmax}_\mathbf{s} \log p(\mathbf{s}|\mathbf{x}) = \mathrm{argmax}_\mathbf{s} \sum_{j=1}^{|\mathbf{s}|} G(\mathbf{s}, \mathbf{x})$. This can be solved by using a Semi-Markov version of the Viterbi algorithm. We use $V_{y,k}$ to denote the maximum value for $\sum_{\mathbf{s}' \in \mathbf{s}_{1:k,y}} G(\mathbf{s}', \mathbf{x})$. Then the update equation is, for $i > 0$,

$$V_{y,k} = \max_{y' \in \mathcal{Y}, d=1,\ldots,L} V_{y',k-d} + g(k-d+1, d, y', y). \tag{18}$$

For the boundary case, we set $V_{y,k} = 0$ for $k \leq 0$. Finally, the best segmentation $\mathbf{s}^*$ corresponds to the path traced by $\max_{y\in\mathcal{Y}} V_{y,T}$.