SUPPLEMENTARY MATERIALS
Reinforcement Learning for Bandit Neural Machine Translation with Simulated Human Feedback

A Neural Machine Translation

Our neural machine translation (NMT) model consists of an encoder and a decoder, each of which is a recurrent neural network (RNN). We closely follow (Luong et al., 2015) for the structure of our model. It directly models the posterior distribution $P_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x})$ of translating a source sentence $\boldsymbol{x} = (x_1, \cdots, x_n)$ to a target sentence $\boldsymbol{y} = (y_1, \cdots, y_m)$:

$$P_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{t=1}^{m} P_{\boldsymbol{\theta}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) \tag{1}$$

where $\boldsymbol{y}_{<t}$ are all tokens in the target sentence prior to y_t.

Each local disitribution $P_{\boldsymbol{\theta}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$ is modeled as a multinomial distribution over the target language's vocabulary. We compute this distribution by applying a linear transformation followed by a softmax function on the decoder's output vector $\boldsymbol{h}_t^{dec}$:

$$P_{\boldsymbol{\theta}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) = \text{softmax}(\boldsymbol{W}_s \, \boldsymbol{h}_t^{dec}) \tag{2}$$

$$\boldsymbol{h}_t^{dec} = \tanh(\boldsymbol{W}_o[\tilde{\boldsymbol{h}}_t^{dec}; \boldsymbol{c}_t]) \tag{3}$$

$$\boldsymbol{c}_t = \text{attend}(\tilde{\boldsymbol{h}}_{1:n}^{enc}, \tilde{\boldsymbol{h}}_t^{dec}) \tag{4}$$

where $[\cdot;\cdot]$ is the concatenation of two vectors, $\text{attend}(\cdot,\cdot)$ is an attention mechanism, $\tilde{\boldsymbol{h}}_{1:n}^{enc}$ are all encoder's hidden vectors and $\tilde{\boldsymbol{h}}_t^{dec}$ is the decoder's hidden vector at time step t. We use the "concat" global attention in (Luong et al., 2015).

During training, the encoder first encodes $\boldsymbol{x}$ to a continuous vector $\Phi(\boldsymbol{x})$, which is used as the initial hidden vector for the decoder. In our paper, $\Phi(\boldsymbol{x})$ simply returns the last hidden vector of the encoder. The decoder performs RNN updates to produce a sequence of hidden vectors:

$$\begin{aligned} \tilde{\boldsymbol{h}}_0^{dec} &= \Phi(\boldsymbol{x}) \\ \tilde{\boldsymbol{h}}_t^{dec} &= f_{\boldsymbol{\theta}}\left(\tilde{\boldsymbol{h}}_{t-1}^{dec}, \left[\boldsymbol{h}_{t-1}^{dec}; e(y_t)\right]\right) \end{aligned} \tag{5}$$

where $e(.)$ is a word embedding lookup function and y_t is the ground-truth token at time step t. Feeding the output vector $\boldsymbol{h}_{t-1}^{dec}$ to the next step is known as "input feeding".

At prediction time, the ground-truth token y_t in Eq 5 is replaced by the model's own prediction $\hat{y}_t$:

$$\hat{y}_t = \arg\max_y P_{\boldsymbol{\theta}}(y \mid \hat{\boldsymbol{y}}_{<t}, \boldsymbol{x}) \tag{6}$$

In a supervised learning framework, an NMT model is typically trained under the maximum log-likelihood objective:

$$\max_{\boldsymbol{\theta}} \mathcal{L}_{sup}(\boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D_{\text{tr}}} \left[\log P_{\boldsymbol{\theta}}\left(\boldsymbol{y} \mid \boldsymbol{x}\right)\right] \tag{7}$$

where D_{tr} is the training set. However, this learning framework is not applicable to bandit learning since ground-truth translations are not available.

References

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*.