A Supplemental Materials

A.1 Implementation of Length Matching Penalty

Let the first state of the backward encoder in a standard NMT model be $\bar{h}_0$. We predict the mean and variance of the Gaussian of expected length of translation with a parameterized function $f^{l_x}_x$:

$$v_x = f^{l_x}_x(\bar{h}_0; \theta_x),$$  \hspace{1cm} (1)

$$\mu_x = v_x[0]; \sigma_x = \text{softplus}(v_x[1]),$$  \hspace{1cm} (2)

where, $f^{l_x}_x$ is a simple two-layer neural network with a ReLU non-linearity applied after the hidden layer, which has 256 units. The final layer $v_x$ is a two-dimensional vector, which contains the predicted mean and variance of the Gaussian.

To predict the distribution of the final length for a hypothesis $y$, we use a tiny LSTM followed by a transformation $f^{d_y}_y$:

$$h_y = \text{LSTM}(e(y); \theta_y)$$  \hspace{1cm} (3)

$$v_y = f^{d_y}(h_y + \bar{h}_0; \theta_y),$$  \hspace{1cm} (4)

$$\mu_y = v_y[0]; \sigma_y = \text{softplus}(v_y[1]),$$  \hspace{1cm} (5)

where, $e(\cdot)$ represents the embeddings of tokens. We train the parameters $\theta_x$ and $\theta_y$ with fixed NMT parameters. Let $L^*$ be the length of the gold output, $L$ be the length of a sampled output obtained by greedy decoding, the loss function is based on the negative log-likelihood of the Gaussians:

$$J = -\log P(L^*; \mu_x, \sigma_x) - \frac{1}{L} \sum_{l=1}^{L} \log P(L; \mu_{y_{1:l}}, \sigma_{y_{1:l}})$$  \hspace{1cm} (6)

where $P(\cdot)$ is a Gaussian distribution with the specified mean and variance. The model is trained with Adam optimizer with a learning rate of 0.0001 for six epochs.