A Supplemental Material

A.1 ConvS2S

For the WMT'14 En→De task, both the encoder and decoder have 15 layers, with 512 hidden units in the first ten layers, 768 units in the subsequent three layers and 2048 units in the final two layers. The first 13 layers use kernel width 3 and the final two layers use kernel width 1. For the WMT'14 En→Fr task, both the encoder and decoder have 14 layers, with 512 hidden units in the first five layers, 768 units in the subsequent four layers, 1024 units in the next three layers, 2048 units and 4096 units in the final two layers. The first 12 layers use kernel width 3 and the final two layers use kernel width 1. We train the ConvS2S models with synchronous training using 32 GPUs.

A.2 Transformer

Both the encoder and the decoder have 6 Transformer layers. Transformer base model has model dimension 512, hidden dimension 2048 and 8 attention heads. The Transformer Big model uses model dimension 1024, hidden dimension 8192 and 16 attention heads. We group the dropout in Transformer models into four types: *input dropout* - dropout applied to the sum of token embeddings and position encodings, *residual dropout* - dropout applied to the output of each sublayer before added to the sublayer input, *relu dropout* - dropout applied to the inner layer output after ReLU activation in each feed-forward sub-layer, *attention dropout* - dropout applied to attention weight in each attention sub-layer. All Transformer models use the following learning rate schedule:

$$lr = \frac{r_0}{\sqrt{d_{model}}} \cdot \min\left(\frac{t+1}{p\sqrt{p}}, \frac{1}{\sqrt{(t+1)}}\right) \qquad (2)$$

where t is the current step, p is the number of warmup steps, d_{model} is the model dimension and r_0 is a constant to adjust the magnitude of the learning rate.

On WMT'14 En→De, the Transformer Base model employs all four types of dropout with $dropout\_probs = 0.1$. We use $r_0 = 2.0$ and $p = 8000$ in the learning rate schedule. For the Transformer Big model, only residual dropout and input dropout are applied, both with $dropout\_probs = 0.3$. $r_0 = 3.0$ and $p = 40000$ are used in the learning rate schedule.

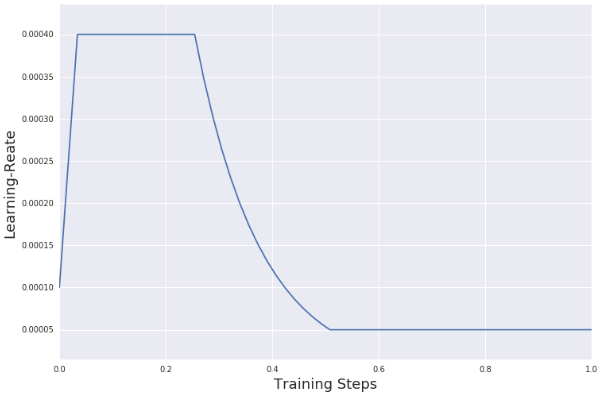

Figure 3: RNMT+ learning-rate schedule.

On WMT'14 En→Fr, the Base model applies only residual dropout and input dropout, each with $dropout\_probs = 0.1$. The learning rate schedule uses $r_0 = 1.0$ and $p = 4000$. For the big model, we apply all four types of dropout, each with $dropout\_probs = 0.1$. The learning rate schedule uses $r_0 = 3.0$ and $p = 40000$.

We train both Transformer base model and big model with synchronous training using 16 GPUs.

A.3 RNMT+

RNMT+ has 1024 LSTM nodes in all encoder and decoder layers. The input embedding dimension is 1024 as well. The encoder final projection layer projects the last bidirectional layer output from dimension 2048 to 1024. We use 4 attention heads in the multi-head additive attention. Label smoothing is applied with an $uncertainty = 0.1$. Figure 3 illustrates our learning rate schedule defined in Eq. 1.

On WMT'14 En→De, we use $p = 500$, $s = 600000$, $e = 1200000$ for the learning rate schedule and apply all dropout types with $dropout\_probs = 0.3$. We apply L2 regularization to the weights with $\lambda = 10^{-5}$. On WMT'14 En→Fr, we use $p = 500$, $s = 1200000$, $e = 3600000$, $dropout\_probs = 0.2$. No weight decay is applied.

RNMT+ models are trained with synchronous training using 32 GPUs.

A.4 Encoder-Decoder Hybrids

For both encoder-decoder hybrids, i.e., Transformer Big encoder with RNMT+ decoder and RNMT+ encoder with Transformer Big decoder, we use the exactly same model hyperparameters as in the Transformer Big and RNMT+ models described in above sections.

We use Transformer learning rate schedule (Eq.

2) for both hybrids. For the WMT'14 En→Fr task, we use $r_0 = 4.0$ and $p = 50000$ for the hybrid with Transformer encoder and RNMT+ decoder, and use $r_0 = 3.0$ and $p = 40000$ for the hybrid with RNMT+ encoder and Transformer decoder. Both hybrid models are trained with synchronous training using 32 GPUs.

A.5 Cascaded Encoder Hybrid

In this hybrid we stack a transformer encoder on top of the RNMT+ encoder. In our experiments we used a pre-trained RNMT+ encoder, including the projection layer, exactly as described in section 4. The outputs of the RNMT+ encoder are layer normalized and fed into a transformer encoder. This structure is illustrated in Figure 2a. The transformer encoder is identical to the one described in subsection 2.3 except for the different number of layers. Our best setup uses 4 Transformer layers stacked on top of a pre-trained RNMT+ encoder with 6 layers. To speed up convergence, we froze gradient updates in the pre-trained RNMT+ encoder. This enables us to increase the encoder capacity significantly, while avoiding optimization issues encountered in non-frozen variants of the hybrid. As an additional benefit, this enables us to train the model on P100s without the need for model parallelism.

Note that this specific layout allows us to drop hand-crafted sinusoidal positional embeddings (since position information is already captured by the underlying RNNs).

We use the Transformer learning rate schedule (Eq. 2) for this hybrid with $r_0 = 2.0$ and $p = 16000$ and train with synchronous training using 32 GPUs. We apply the same dropouts used for the transformer model to the transformer layers in the encoder, and apply L2 weight decay with $\lambda = 10^{-5}$ to the decoder layers.

A.6 Multi-Column Encoder Hybrid

We use a simple concatenation as the merger-operator without fine-tuning any other model hyperparameters. After concatenation, the combined representation is projected down to the decoder dimension with a layer-normalized affine transformation. Although in this paper we only use two columns, there is no practical restriction on the total number of columns that this hybrid can combine. By combining multiple encoder representations, the network may capture different factors of variations in the input sequence.

Similar to the Cascaded-RNMT+ hybrid, we use pre-trained encoders that are borrowed from an RNMT+ model (we used a pretrained RNMT+ encoder as the first column) and an Encoder-Decoder hybrid model with Transformer encoder and RNMT+ decoder (we used the pretrained Transformer encoder). Multi-column encoder with RNMT+ decoder is trained using 16 GPUs in a synchronous training setup. We stick to the simple concatenation operation as the merger-operator, and after concatenation, the combined representation is projected down the decoder dimension with a simple layer-normalized affine transformation. One additional note that we observed for the sake of stability and trainability, each column output should be first mapped to a space where the representation ranges are compatible, e.g., RNMT+ encoder output has not limitation on its range, but a Transformer Encoder output range is constrained by the final layer normalization applied to the entire Transformer encoder body. Therefore, we also applied layer normalization to the RNMT+ encoder outputs to match the ranges of individual encoders.

On WMT'14 En→De, we use $p = 50$, $s = 300000$, $e = 900000$ for the learning rate schedule and apply all dropout types with $dropout\_probs = 0.3$. We apply L2 regularization to the weights with $\lambda = 10^{-5}$. On WMT'14 En→Fr, we use Transformer learning rate schedule (Eq. 2) $r_0 = 1.0$ and $p = 10000$. No weight decay or dropout is applied.