A Supplemental Material

A.1 Machine Translation

We conducted experiments on the widely-used WMT14 English⇒German dataset[1] consisting of about 4.56M sentence pairs. We used newstest2013 and newstest2014 as development set and test set respectively. We applied byte pair encoding (BPE) toolkit[2] with 32K merge operations. The case-sensitive NIST BLEU score (Papineni et al., 2002) is used as the evaluation metric. All models were trained on eight NVIDIA Tesla P40 GPUs where each was allocated with a batch size of 4096 tokens.

For *Base* model, it has embedding size and hidden size of 512, filter size of 2048 and attention heads of 8. Compared with *Base* model, *Big* model has embedding size and hidden size of 1024, filter size of 4096 and attention heads of 16. For both *Base* and *Big* models, the number of encoder and decoder layer is 6, all types of dropout rate is 0.1. Adam (Kingma and Ba, 2015) is used with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. The learning rate is 1.0 and linearly warms up over the first 4,000 steps, then decreases proportionally to the inverse square root of the step number. Label smoothing is 0.1 during training (Szegedy et al., 2016). All the results we reported are based on the individual models without using the averaging model or ensemble.

A.2 Targeted Linguistic Evaluation

We conducted 10 probing tasks[3] to study what linguistic properties are captured by the encoders (Conneau et al., 2018). A probing task is a classification problem that focuses on simple linguistic properties of sentences. 'SeLen' predicts the length of sentences in terms of number of words. 'WC' tests whether it is possible to recover information about the original words given its sentence embedding. 'TrDep' checks whether an encoder infers the hierarchical structure of sentences. In 'ToCo' task, sentences should be classified in terms of the sequence of top constituents immediately below the sentence node. 'BShif' tests whether two consecutive tokens within the sentence have been inverted. 'Tense' asks for the tense of the main-clause verb. 'SubN' focuses on the number of the main clause's subject. 'ObjN' tests for the number of the direct object of the main clause. In 'SoMo', some sentences are modified by replacing a random noun or verb with another one and the classifier should tell whether a sentence has been modified. 'CoIn' contains sentences made of two coordinate clauses. Half of sentences are inverted the order of the clauses and the task is to tell whether a sentence is intact or modified.

Each of our probing model consists a pretrained encoder of model variations from machine translation followed by a MLP classifier (Conneau et al., 2018). The mean of the encoding layer is served as the sentence representation passed to the classifier. The MLP classifier has a dropout rate of 0.3, a learning rate of 0.0005 with Adam optimizer and were trained for 250 epochs.

A.3 Logical Inference

We used the artificial data[4] described in Bowman et al. (2015). The train/dev/test dataset ratios are set to 0.8/0.1/0.1 with the number of logical operations range from 1 to 12. We followed Tran et al. (2018) to implement the architectures: premise and hypothesis sentences are encoded in fixed-size vectors, which are concatenated and fed to a three layer feed-forward network for classification of the logical relation. For LSTM based models, we took the last hidden state of the top layer as a fixed-size vector representation of the sentence. For the hybrid and SANs models, we used two trainable queries to obtain the fixed-size representation.

In our experiments, both word embedding size and hidden size are set to 256. All models have two layers, a dropout rate of 0.2, a learning rate of 0.0001 with Adam optimizer, and were trained for 100 epochs. Especially, for hybrid model, we stacked one ON-LSTM layer and one SANs layer subsequently. Short-Cut connection between layers is added into all models for fair comparison.

References

Samuel R. Bowman, Christopher D. Manning, and Christopher Potts. 2015. Tree-structured composition in neural networks without tree-structured architectures. *NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*.

[1] http://www.statmt.org/wmt14/translation-task.html

[2] https://github.com/rsennrich/subword-nmt

[3] https://github.com/facebookresearch/SentEval/tree/master/data/probing

[4] https://github.com/sleepinyourhat/vector-entailment

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *ACL*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *ICLR*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *CVPR*.

Ke Tran, Arianna Bisazza, and Christof Monz. 2018. The importance of being recurrent for modeling hierarchical structure. In *EMNLP*.