(2017) use canonical correlation analysis (CCA) and a new method, singular vector canonical correlation analysis (SVCCA), to show that early layers converge faster than upper layers in convolutional neural networks. Kudugunta et al. (2019) use SVCCA to investigate the multilingual representations obtained by the encoder of a massively multilingual neural machine translation system (Aharoni et al., 2019). Kornblith et al. (2019) argues that CCA fails to measure meaningful similarities between representations that have a higher dimension than the number of data points and introduce the centered kernel alignment (CKA) to solve this problem. They successfully use CKA to identify correspondences between activations in networks trained from different initializations.

3 Cross-lingual Pretraining

We study a standard multilingual masked language modeling formulation and evaluate performance on several different cross-lingual transfer tasks, as described in this section.

3.1 Multilingual Masked Language Modeling

Our multilingual masked language models follow the setup used by both mBERT and XLM. We use the implementation of Lample and Conneau (2019). Specifically, we consider continuous streams of 256 tokens and mask 15% of the input tokens which we replace 80% of the time by a mask token, 10% of the time with the original word, and 10% of the time with a random word. Note the random words could be foreign words. The model is trained to recover the masked tokens from its context (Taylor, 1953). The subword vocabulary and model parameters are shared across languages. Note the model has a softmax prediction layer shared across languages. We use Wikipedia for training data, preprocessed by Moses (Koehn et al., 2007) and Stanford word segmenter (for Chinese only) and BPE (Sennrich et al., 2016) to learn subword vocabulary. During training, we sample a batch of continuous streams of text from one language proportionally to the fraction of sentences in each training corpus, exponentiated to the power 0.7.

Pretraining details Each model is a Transformer (Vaswani et al., 2017) with 8 layers, 12 heads and GELU activiation functions (Hendrycks and Gimpel, 2016). The output softmax layer is tied with input embeddings (Press and Wolf, 2017). The embeddings dimension is 768, the hidden dimension of the feed-forward layer is 3072, and dropout is 0.1. We train our models with the Adam optimizer (Kingma and Ba, 2014) and the inverse square root learning rate scheduler of Vaswani et al. (2017) with 10^{-4} learning rate and 30k linear warmup steps. For each model, we train it with 8 NVIDIA V100 GPUs with 32GB of memory and mixed precision. It takes around 3 days to train one model. We use batch size 96 for each GPU and each epoch contains 200k batches. We stop training at epoch 200 and select the best model based on English dev perplexity for evaluation.

3.2 Cross-lingual Evaluation

We consider three NLP tasks to evaluate performance: natural language inference (NLI), named entity recognition (NER) and dependency parsing (Parsing). We adopt the **zero-shot cross-lingual transfer** setting, where we (1) fine-tune the pretrained model on English and (2) directly transfer the model to target languages. We select the model and tune hyperparameters with the English dev set. We report the result on average of best two set of hyperparameters.

Fine-tuning details We fine-tune the model for 10 epochs for NER and Parsing and 200 epochs for NLI. We search the following hyperparameter for NER and Parsing: batch size $\{16, 32\}$; learning rate $\{$2e-5, 3e-5, 5e-5$\}$. For XNLI, we search: batch size $\{4, 8\}$; encoder learning rate $\{$1.25e-6, 2.5e-6, 5e-6$\}$; classifier learning rate $\{$5e-6, 2.5e-5, 1.25e-4$\}$. We use Adam with fixed learning rate for XNLI and warmup the learning rate for the first 10% batch then decrease linearly to 0 for NER and Parsing. We save checkpoint after each epoch.

NLI We use the cross-lingual natural language inference (XNLI) dataset (Conneau et al., 2018). The task-specific layer is a linear mapping to a softmax classifier, which takes the representation of the first token as input.

NER We use WikiAnn (Pan et al., 2017), a silver NER dataset built automatically from Wikipedia, for English-Russian and English-French. For English-Chinese, we use CoNLL 2003 English (Tjong Kim Sang and De Meulder, 2003) and a Chinese NER dataset (Levow, 2006), with realigned Chinese NER labels based on the Stanford word segmenter. We model NER as BIO tagging. The task-specific layer is a linear mapping to a softmax