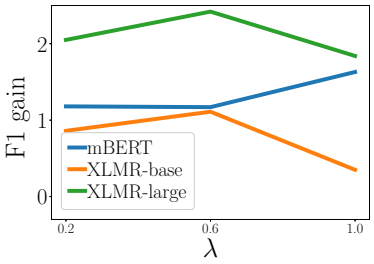

Figure 9: Average F1 gains over the baseline on NER task.

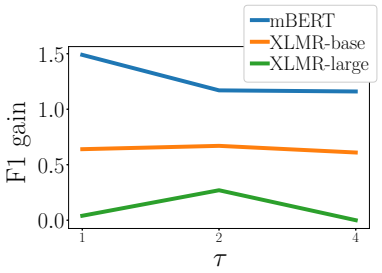

Figure 10: Average F1 gains over the baseline on QA tasks.

A Appendix

A.1 Effect of hyperparameters

In this section, we study the effect of two hyperparameters in MVR: the weight of consistency loss λ and the temperature τ for flattening the prediction distribution. First, we analyze the effect of λ on the NER tasks in Fig. 9. We notice that mBERT performs better using a larger λ, while in general a value between 0.2 to 0.6 works reasonably well for all models. Note that we still use $\lambda = 0.2$ for mBERT because we selected this value based on the performance on the English dev set.

We only tune the temperature τ for question answering tasks because it has a larger output space than other tasks. We plot the average F1 improvement on the QA tasks in Fig. 10.

Simplying using $\tau = 1$ works well for the smaller mBERT and XLM-R base models. The best-performing XLM-R large model benefits from a larger τ, or a flattened distribution, probably because its prediction distribution is relatively sharper or more confident than others. Generally the model is not particularly sensitive to the value of τ.

Sequence tagging task requires truncating the inputs and targets to a predefined length. This could be a problem for calculating KL divergence when the deterministic and probabilistic segmented inputs have different number of tags been discarded. In our implementation of MVR, we simply calculate the KL divergence on the tags shared by the two inputs. Details of this implementation can be found in the code base.

A.2 Training details

We select hyperparameters based on the validation performance of English on the NER task. We finetune all models on the NVIDIA V100 GPU. Both SR and MVR have the same number of model parameters as the baseline. Baseline experiments on all tasks except for the XNLI classification task generally finish within 5 hours on a single GPU. The baseline experiment on XNLI takes about 24 hours on 2 GPUs. SR takes about the same training time as the baseline, and MVR takes about twice the amount of time.

A.3 Other analysis

MVR improves more for languages with non-Latin script We further compare the gains of MVR over SR on languages with Latin and non-Latin script. The plots can be found in Fig. 11. Overall MVR leads to larger improvements over subword regularization for languages with non-Latin script for both mBERT and XLM-R large. By enforcing prediction consistency between different segmentation, MVR is better than SR at making the model robust to languages with very different segmentation than the language used for finetuning.

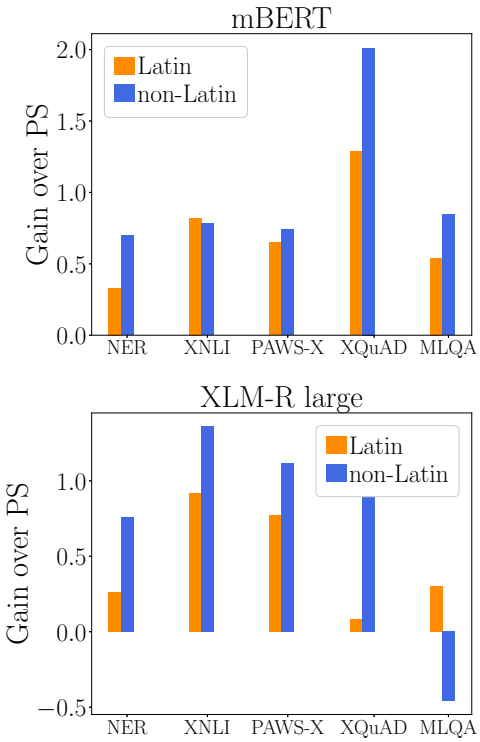

Figure 11: Gains of MVR over SR for languages with Latin vs. non-Latin script.