Multi-view Subword Regularization

Anonymous

Abstract

Multilingual pretrained representations generally rely on subword segmentation algorithms to create a shared multilingual vocabulary. However, standard heuristic algorithms often lead to sub-optimal segmentation, especially for languages with limited amounts of data. In this paper, we take two major steps towards alleviating this problem. First, we demonstrate empirically that applying existing subword regularization methods (Kudo, 2018; Provilkov et al., 2020) during fine-tuning of pre-trained multilingual representations improves the effectiveness of cross-lingual transfer. Second, to take full advantage of different possible input segmentations, we propose Multi-view Subword Regularization (MVR), a method that enforces the consistency between predictions of using inputs tokenized by the standard and probabilistic segmentations. Results on the XTREME multilingual benchmark (Hu et al., 2020) show that MVR brings consistent improvements of up to 2.5 points over using standard segmentation algorithms.[1]

1 Introduction

Multilingual pre-trained representations (Devlin et al., 2019; Huang et al., 2019; Conneau and Lample, 2019; Conneau et al., 2020) are now an essential component of state-of-the-art methods for cross-lingual transfer (Wu and Dredze, 2019; Pires et al., 2019). These methods pretrain an encoder by learning in an unsupervised way from raw textual data in up to hundreds of languages which can then be fine-tuned on annotated data of a downstream task in a high-resource language, often English, and transferred to another language. In order to encode hundreds of languages with diverse vocabulary, it is standard for such multilingual models to employ a shared subword vocabulary jointly learned on the multilingual data using heuristic word segmentation methods based on byte-pair-encoding (BPE; Sennrich et al., 2016) or unigram language models (Kudo and Richardson, 2018) (details in §2). However, subword-based preprocessing can lead to sub-optimal segmentation that is inconsistent across languages, harming cross-lingual transfer performance, particularly on under-represented languages. As one example, consider the segmentation of the word "excitement" in different languages in Tab. 1. The English word is not segmented, but its translations in the other languages, including the relatively high-resourced French and German, are segmented into multiple subwords. Since each subword is mapped to a unique embedding vector, the segmentation discrepancy—which generally does not agree with a language's morphology—could map words from different languages to very distant representations, hurting cross-lingual transfer. In fact, previous work (Conneau et al., 2020; Artetxe et al., 2020) has shown that heuristic fixes such as increasing the subword vocabulary capacity and upsampling low-resource languages during learning of the subword segmentation can lead to significant performance improvements.

en	excitement	**fr**	excita/tion
de	Auf/re/gung	**pt**	excita/ção
el	εν/θ/ουσι/ασμός	**ru**	волн/ение

Table 1: XLM-R segmentation of "excitement" in different languages. The English word is not segmented while the same word in other languages is over-segmented. A better segmentation would allow the model to match the verb stem and derivational affix across languages.

Despite this, there is not much work studying or improving subword segmentation methods for cross-lingual transfer. Bostrom and Durrett (2020) empirically compare several popular word segmentation algorithms for pretrained language models of a single language. Several works propose to use different representation granularities, such as phrase-level segmentation (Zhang and Li, 2020) or character-aware representations (Ma et al., 2020)

[1] We will release the code upon acceptance of the paper.

for pretrained language models of a single high-resource language, such as English or Chinese only. However, it is not a foregone conclusion that methods designed and tested on monolingual models will be immediately applicable to multilingual representations. Furthermore, they add significant computation cost to the pretraining stage, which is especially problematic for multilingual pretraining on hundreds of languages. The problem of sub-optimal subword segmentation has drawn more attention in the context of neural machine translation (NMT). Specifically, *subword regularization* methods have been proposed to improve the NMT model of a *single* language pair by randomly sampling different segmentations of the sentences during training (Kudo, 2018; Provilkov et al., 2020). However, these methods have not been applied to multilingual NMT or pretrained language models and it is similarly not clear if they are useful for cross-lingual transfer.

In this paper, we make two contributions to close this gap. First, we perform the first (to our knowledge) empirical examination of subword regularization methods on a variety of cross-lingual transfer tasks from the XTREME benchmark (Hu et al., 2020). We demonstrate that despite its simplicity, this method is highly effective, providing consistent improvements across a wide variety of languages and tasks for both multilingual BERT (mBERT; Devlin et al., 2019) and XLM-R (Conneau et al., 2020) models. Analysis of the results shows that this method is particularly effective for languages with non-Latin scripts despite only being applied during English fine-tuning.

Further, we posit that naively applying probabilistic segmentation only during fine-tuning may be sub-optimal as it creates a discrepancy between the segmentations during the pretraining and fine-tuning stages. To address this problem, we propose Multi-view Subword Regularization (MVR; Fig. 1), a novel method—inspired by the usage of consistency regularization in semi-supervised learning methods (Clark et al., 2018; Xie et al., 2018)—which utilizes *both* the standard and probabilistically segmented inputs, enforcing the model's predictions to be consistent across the two views. Such consistency regularization further improves accuracy, with MVR finally demonstrating consistent gains of up to 2.5 points over the standard practice across all models and tasks. We analyze the sources of the improvement from consistency

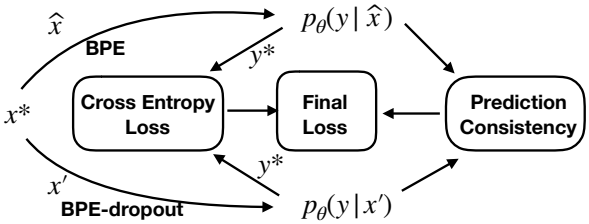

Figure 1: Fine-tuning models using MVR on data (x^*, y^*)

regularization and find that it can be attributed to both label smoothing and self-ensembling.

2 Background: Subword Segmentation

Here, we first discuss two common deterministic segmentation methods based on byte pair encoding (BPE) and unigram language models (ULM), discuss their probabilistic variants, and explain how to incorporate them in training.

2.1 Deterministic Segmentation

The most widely used subword segmentation methods first estimate a segmentation model from the training corpus in an unsupervised fashion. They then produce a segmentation $\widehat{x}$ of the input x^* under the estimated segmentation model $P(x)$:

$$\widehat{x} = \underset{x \in S(x^*)}{\operatorname{argmax}} P(x)$$

Here $S(x^*)$ is the set of all possible segmentations, and $P(x)$ is the likelihood of a given segmentation. Note that $\widehat{x}$ is deterministically selected for each input x^*.

Byte-pair encoding (BPE) The popular BPE algorithm (Sennrich et al., 2016) initializes the vocabulary with individual characters and initially represents each word as a sequence of characters. It then counts the most frequent character token bigrams in the data, merges them into a new token, and adds the new token to the vocabulary. This process is done iteratively until a predefined vocabulary size is reached.

To segment a word, BPE simply splits the word into character tokens, and iteratively merges adjacent tokens with the highest priority until no merge operation is possible. That is, for an input x^*, it assigns segmentation probability $P(\widehat{x}) = 1$ for the sequence $\widehat{x}$ obtained from the greedy merge operations, and assigns other possible segmentations a probability of 0.

Notably, a variant of this method (Schuster and Nakajima, 2012) is used for the mBERT embedding model (Devlin et al., 2019).

Unigram language model (ULM) The ULM method (Kudo and Richardson, 2018) starts from a reasonably large seed vocabulary, which is iteratively pruned to maximize the training corpus likelihood under a unigram language model of the subwords until the desired vocabulary size is reached.

During segmentation, ULM decodes the most likely segmentation of a sentence under the estimated language model using the Viterbi algorithm. This method is used in the XLM-R cross-lingual embeddings (Conneau et al., 2020).

2.2 Probabilistic Segmentation

As explained in §1, one drawback of both word segmentation algorithms is that they produce a deterministic segmentation for each sentence, even though multiple segmentations are possible given the same vocabulary. In contrast, Kudo (2018) and Provilkov et al. (2020) have proposed methods that enable the model to generate segmentations probabilistically. Instead of selecting the best subword sequence for input x^*, these method stochastically sample a segmentation x' as follows:

$$x' \sim P'(x) \text{ where } P'(x) \propto \begin{cases} P(x) \text{ if } x \in S(x^*) \\ 0 \text{ otherwise} \end{cases}$$

Here we briefly introduce these two methods.

BPE-dropout This method is used together with the BPE algorithm, randomly dropping merge operations with a given probability p while segmenting the input data (Provilkov et al., 2020).

ULM-sample As the ULM algorithm relies on a language model to score segmentation candidates for picking the most likely segmentation, Kudo (2018) propose to sample from these segmentation candidates based on their language model scores.

2.3 Subword Regularization (SR)

Subword regularization (Kudo, 2018) is a method that incorporates probabilistic segmentation at training time to improve the robustness of models to different segmentations. The idea is conceptually simple: at training time sample different segmentations x' for each input sentence x^*. Previous works (Kudo, 2018; Provilkov et al., 2020) have demonstrated that subword regularization using both BPE-dropout and ULM-sampling are effective at improving machine translation accuracy, particularly in cross-domain transfer settings where the model is tested on a different domain than the one on which it is trained.

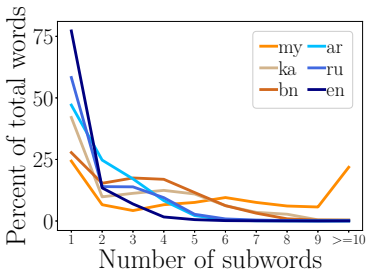

Figure 2: Percentage of words with different number of segments from different languages.

3 Subword Regularization for Cross-lingual Transfer

While sub-optimal word segmentation is a challenge in monolingual models, it is an even bigger challenge for multilingual pretrained models. These models train a shared subword segmentation model jointly on data from many languages, but the segmentation can nonetheless be different across languages, stemming from two main issues. First, the granularity of segmentation differs among languages, where the segmentation model tends to *over-segment* low-resource languages that do not have enough representation in the joint training data (Ács, 2019). Fig. 2 shows the distribution of words from languages from different language families based on the number of subwords they are split into.[2] We can see that the majority of English words are not segmented at all, while many languages only have less than half of the words unsegmented. Notably, even though Burmese (my) is a language with little inflectional morphology, almost a quarter of the words are segmented into more than nine subwords. Second, the segmentation might still be *inconsistent* between different languages even if the granularity is similar, as explained in Tab. 1. For example, neither the English word "excitement" nor the same word in French "excita/tion" are overly segmented, but segmenting the English word into "excite/ment" would allow the model to learn a better cross-lingual alignment.

Despite these issues, few methods have tried to address this subword segmentation problem for multilingual pretrained models. Chau et al. (2020) propose to adapt a pretrained multilingual model to a new language by augmenting the vocabulary with a new subword vocabulary learned on the target language, but this method might not help for lan-

[2]We use Pan et al. (2017)'s named entity recognition test data with mBERT's tokenizer.

guages other than the target language it adapts to. Chung et al. (2020) propose to separately construct a subword segmentation model for each cluster of related languages for *pretraining* the multilingual representations. However, directly modifying the word segmentation requires retraining large pretrained models, which is computationally prohibitive in most cases.

In this paper, we instead propose a more efficient approach of using probabilistic segmentation during *fine-tuning* on labeled data of a downstream task. As mismatch in segmentation is one of the factors harming cross-lingual transfer, we expect a model that becomes more robust to different varieties of segmentation in one language will be more accommodating to differing segmentations in other languages during inference. Despite the simplicity of this method it is, as far as we are aware, unattested in the literature, and we verify in § 5.3 that it significantly improves the cross-lingual transfer performance of multilingual pretrained models.

4 Multi-view Subword Regularization

Previous attempts at SR have mainly applied it to models trained *from scratch* for tasks such as MT. However, the situation is somewhat different when fine-tuning pre-trained representations, in which case the original pre-trained models are generally not trained on sampled segmentations. This discrepancy between the segmentation of the English labeled data and the segmentation of English monolingual data during pretraining might hurt the ability of the model to take full advantage of the parameters learned during the pretraining stage. To reduce this pretraining–fine-tuning discrepancy, we propose Multi-view Subword Regularization (MVR), a method for learning from multiple segmented versions of the same data and enforcing the consistency of predictions over different segmentations.

Given the input $\widehat{x}_i$ tokenized with the deterministic segmentation such as BPE, and x'_i, the same input tokenized with the corresponding probabilistic segmentation algorithm such as BPE-dropout, the objective for MVR has three components

$$J(\theta) = \sum_{i=1}^{n} \Big[-\frac{1}{2} \underbrace{\log p_\theta(y_i|\widehat{x}_i)}_{\text{Det. Seg CrossEnt}} - \frac{1}{2} \underbrace{\log p_\theta(y_i|x'_i)}_{\text{Prob. Seg CrossEnt}} + \lambda \underbrace{D(p_\theta(y_i|\widehat{x}_i) \,||\, p_\theta(y_i|x'_i))}_{\text{Consistency loss}} \Big] \quad (1)$$

1. A cross-entropy loss using the standard deterministic segmentation. This loss acts on data whose segmentation is consistent with the segmentation seen during pretraining. It thus maximizes the benefit of pretrained representations.
2. A cross entropy loss using probabilistic segmentation. It allows the model to learn from different possible segmentations of the same input.
3. A distance term $D(\cdot \,||\, \cdot)$ between the model prediction distributions over the two different versions of the input. We use KL divergence as the distance metric and a hyperparameter λ to balance the supervised cross-entropy losses and the consistency loss. Minimizing the distance between the two distributions enforces the model to make consistent predictions under different input segmentations, making it robust to sub-optimal segmentation of multilingual data.[3]

Flattening the prediction The benefit of consistency regularization might be limited if the model prediction becomes overly confident on certain classes, especially when the number of output classes is large. Inspired by a similar technique in knowledge distillation (Hinton et al., 2014), we can use a softmax temperature τ to flatten the prediction distribution when calculating the consistency loss. Specifically, the distance loss between two prediction distributions in Eq. 1 can be written as $D(p^{\text{flat}}_\theta(y_i|\widehat{x}_i) \,||\, p_\theta(y_i|x'_i))$, where

$$p^{\text{flat}}_\theta(y_i|\widehat{x}_i) = \frac{\exp(z_y)/\tau}{\sum_{y'} \exp(z_{y'})/\tau} \quad (2)$$

and z_y is the logit for output label y_i. Normally τ is set to 1, and a higher τ makes the probability distribution more evenly distributed over all classes. In our experiments, we find that $\tau = 1$ works well for most of the tasks and $\tau = 2$ works slightly better for tasks that have larger output label spaces.

Efficiency At inference time, we simply use the model prediction based on the input tokenized by deterministic segmentation only. Therefore, our method does not add additional decoding latency. MVR needs about twice the fine-tuning cost compared to the baseline. However, compared to pretraining and inference usage of a model, fine-tuning is generally the least expensive component.

[3] As in semi-supervised learning (Clark et al., 2018), we expect our method to also be effective when applied to unlabeled data, e.g. using target language adaptation (Pfeiffer et al., 2020), which we leave for future work.

5 Experiments

5.1 Training and evaluation

We evaluate the multilingual representations using tasks from the XTREME benchmark (Hu et al., 2020), focusing on the zero-shot cross-lingual transfer with English as the source language. We consider sentence classification tasks including XNLI (Conneau et al., 2018) and PAWS-X (Yang et al., 2019), a structured prediction task of multilingual NER (Pan et al., 2017), and question-answering tasks including XQuAD (Artetxe et al., 2020) and MLQA (Lewis et al., 2020).

5.2 Experiment setup

We evaluate on both the mBERT model which utilizes BPE to tokenize the inputs, and the XLM-R models which uses ULM segmentation. To replicate the baseline, we follow the hyperparameters provided in the XTREME codebase[4]. Models are fine-tuned on English training data and zero-shot transferred to other languages. We run each experiment with 5 random seeds and record the average results and the standard deviation.

SR We use BPE-dropout (Provilkov et al., 2020) for mBERT and ULM-sample (Kudo, 2018) for XLM-R models to do probabilistic segmentation of the English labeled data. BPE-dropout sets a dropout probability of $p \in [0, 1]$ for the merge operations, where a higher p corresponds to stronger regularization. ULM-sample utilizes a sampling temperature $\alpha \in [0, 1]$ to scale the scores for segmentation candidates, and a lower α leads to stronger regularization. We select the p and α values based on the model performance on the English dev set of the NER task and simply use the same values across all other tasks. We set $p = 0.1$ for BPE-dropout and $\alpha = 0.6$ for ULM-sample.

MVR We select the hyperparameters for MVR using the English dev set performance on the NER task. MVR works slightly better by using stronger regularization than SR, likely because using inputs deterministically segmented by the standard algorithm can balance the negative impact of bad tokenization by sampling from a more diverse set of segmentation candidates. We use $\lambda = 0.2, p = 0.2$ for mBERT and $\lambda = 0.6, \alpha = 0.2$ for XLM-R. We use prediction temperature $\tau = 2$ for the question-answering tasks XQuAD and MLQA for the XLMR mdoels, and simply use $\tau = 1$ for all other tasks. Further analysis of hyperparameters on the performance of MVR can be found in § A.1.

[4] https://github.com/google-research/xtreme

5.3 Main results

We compare performance of SR, MVR and the baseline for all models in Tab. 2, focusing on the average performance on all languages for each task. Our baseline numbers match or exceed the benchmark results in Hu et al. (2020) for both mBERT and XLM-R large (Hu et al. (2020) do not include results for XLM-R base) on almost all tasks.

Applying SR on English significantly improves other languages SR is surprisingly effective for mBERT—it is comparable to the baseline on XNLI and significantly improves over the baseline for the rest of the four tasks. However, the gains are less consistent for XLM-R models. For both XLM-R base and large, SR leads to improvements on the NER task and the PAWS-X classification task, but is mostly comparable to the baseline for the rest of the three tasks. SR performs better for mBERT likely because the vocabulary of mBERT is more imbalanced than that of XLM-R; it thus benefits more from the regularization methods. mBERT relies on BPE, which could be worse than ULM at tokenizing subwords into morphologically meaningful units (Bostrom and Durrett, 2020). Furthermore, mBERT has only 100K words in the vocabulary while XLM-R has a much larger vocabulary of 250K.

MVR consistently improves over SR For mBERT, it leads to improvements of over 1 to 2 points over the baseline for all tasks. It is also very effective for the XLM-R models. For both the XLM-R base and the stronger XLM-R large models, MVR improves over 1 point over the baseline on the NER task and the two classification tasks. On the question-answering tasks, MVR delivers strong improvements for the XLM-R base model while the improvements on the XLM-R large model is slightly smaller. It has around 0.5 point improvement on XQuAD and has the same performance on MLQA. MVR leads to more improvements on XQuAD, probably because it has a more diverse set of languages that potentially have more sub-optimal subword segmentation. The consistent gains on both mBERT and XLM-R show that MVR is a general and flexible method for a

Model	Method	Avg.	XNLI	PAWS-X	NER	XQuAD	MLQA
Metrics			Acc.	Acc.	F1	F1/EM	F1/EM
mBERT	Hu et al. (2020)	67.1	65.4	81.9	62.2	64.5 / 49.4	61.4 / 44.2
	Baseline (ours)	67.3	66.5±0.4	83.1±0.4	61.5±0.7	64.7±0.2 / 49.8±0.4	60.9±0.4 / 43.8±0.5
	SR	68.0	66.4±0.2	85.0±0.3	62.2±0.6	64.7±0.3 / 50.0±0.3	61.5±0.3 / 44.4±0.3
	MVR	**68.8**	**67.2**±0.3	**85.6**±0.3	**62.7**±0.4	**66.3**±0.2 / **51.7**±0.2	**62.2**±0.2 / **45.3**±0.1
XLM-R base	Baseline (ours)	71.1	74.4±0.2	84.3±0.7	60.6±0.6	70.9±0.3 / 54.9±0.5	65.5±0.3 / 47.7±0.2
	SR	71.4	74.4±0.7	85.5±0.5	61.0±0.6	70.9±0.3 / 55.7±0.2	65.4±0.1 / 47.5±0.1
	MVR	**72.3**	**75.3**±0.3	**86.3**±0.6	**61.8**±0.3	**71.6**±0.5 / **56.5**±0.4	**66.4**±0.5 / **48.5**±0.4
XLM-R large	Hu et al. (2020)	75.8	79.2	86.4	65.4	76.6 / 60.8	71.6 / 53.2
	Baseline (ours)	76.1	80.3±0.4	86.9±0.5	63.6±0.3	77.0±0.2 / 61.7±0.3	**72.8**±0.2 / **54.5**±0.1
	SR	76.5	80.1±0.5	87.3±0.4	65.5±0.6	77.2±0.2 / 62.0±0.2	72.5±0.1 / 54.0±0.2
	MVR	**77.2**	**81.3**±0.1	**88.2**±0.2	**66.0**±0.7	**77.6**±0.2 / **62.5**±0.4	**72.8**±0.2 / **54.5**±0.1

Table 2: Average performance and standard deviation of different methods for mBERT, XLM-R base and XLM-R large models. SR is especially effective for mBERT. MVR leads to significant further improvements across all models and tasks.

variety of pretrained multilingual models based on different segmentation methods.

5.4 Effect of each loss component

In this section, we verify the effectiveness of the three loss components in MVR by removing each of them from the objective. The ablation results on mBERT for all tasks are listed in Tab. 3. Removing any of the three loss components hurts the model performance by about the same amount for most of the tasks. For the question answering tasks, however, removing the cross-entropy loss on the deterministically segmented inputs reduces the model performance by almost half. This is likely because under this setting, the model only learns to locate exact spans for inputs tokenized by BPE-dropout, while we use the standard BPE to segment the inputs at test time.

6 Analysis

In this section, we perform several analyses to better understand the behavior and root causes of the accuracy gains realized by our method.

6.1 Effect on over-segmentation

In this section, we analyze the effect of our methods on languages and words with different subword segmentation granularity. We focus on the NER task because it contains a diverse set of over 40 languages. We calculate the average number of subword pieces in a language, and plot the gains over the baseline for these languages with respect to their average subwords in Fig. 3. To visualize the relationship between the two values, we also fit a trend line and record its coefficient for each method in the legend. We consider three methods for mBERT: SR, MVR without consistency loss, and the full MVR. The trend line for MVR has a positive coefficient, indicating that it improves more on languages that are more overly segmented. Removing the consistency loss tends to hurt more for these languages. SR, on the other hand, does not tend to favor languages with more subword segmentation.

Next, we bucket all the words together based on how many subwords they are segmented into, and compare the performance of our methods for each word bucket. We use the XLM-R model and plot the results in Fig. 4. SR brings slightly more improvements on average for words that are split into 4 or more pieces for the large model. MVR outperforms SR for all categories, especially for difficult words that are segmented into 5 or more subwords.

Gains on Latin vs. non-Latin script In addition, it is notable that we fine-tune the model using labeled data from English, a Latin script language, while the non-Latin scripted languages might have larger segmentation and vocabulary discrepancies from English. We thus also plot the score improvements of both SR and MVR over the baseline for languages with and without Latin script in Fig. 6. We use a lighter shade to represent improvements for Latin-script languages and a darker shade for languages with non-Latin scripts. Across all the tasks, both SR and MVR generally have larger improvements on languages with non-Latin script. MVR, which is represented by blue shades, generally outperforms SR for both the Latin and non-Latin scripted languages across all models. While SR sometimes underperforms the baseline on Latin scripted languages, especially for XLM-R models, MVR delivers consistent improvements over the

Method Metrics	Avg.	XNLI Acc.	PAWS-X Acc.	NER F1	XQuAD F1/EM	MLQA F1/EM
MVR	68.8	67.2±0.3	85.6±0.3	62.7±0.4	66.3±0.2 / 51.7±0.2	62.2±0.2 / 45.3±0.1
– Det. Seg CrossEnt	52.8	66.7±0.5	85.5±0.2	62.4±0.7	25.0±8.2 / 15.6±6.7	24.3±8.6 / 13.1±6.4
– Prob. Seg CrossEnt	67.7	66.7±0.6	85.0±0.3	62.3±0.7	64.0±0.3 / 48.7±0.4	60.3±0.4 / 43.1±0.3
– consistency loss	68.2	66.5±0.7	85.3±0.3	62.3±0.6	65.2±0.2 / 50.2±0.4	61.7±0.1 / 44.5±0.2

Table 3: Effect of removing each loss component on mBERT.

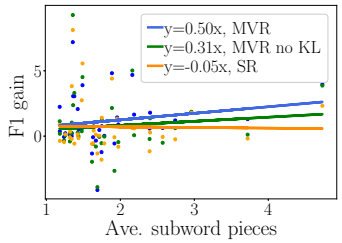

Figure 3: mBERT gains over the NER baseline by average word pieces of a language. MVR tends to benefit over-segmented languages more.

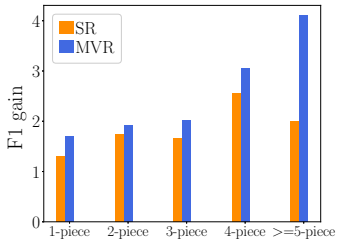

Figure 4: XLM-R large gains over the NER baseline for words with increasing number of subword pieces.

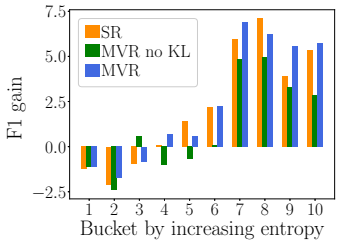

Figure 5: mBERT improvements over the NER baseline for examples with different entropy. Consistency loss helps examples with higher entropy more.

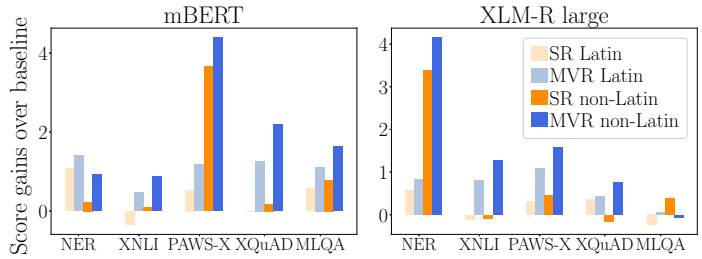

Figure 6: Gains over the baseline for languages with Latin vs. non-Latin script. Both SR and MVR improve more for non-Latin languages.

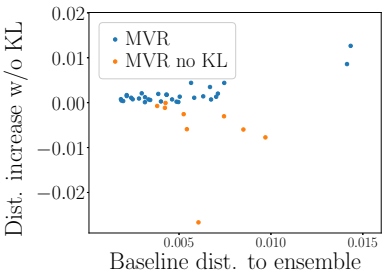

Figure 7: Increase in distance to the ensemble distribution by removing the consistency loss on the NER task. The languages are labeled based on the method with closer distance to the ensemble distribution. The consistency loss shifts model prediction closer to the ensemble distribution.

baseline across both types of languages.

6.2 Effect of consistency loss

One of the novel components of MVR is the consistency loss between two different segmentations of the input. In this section we analyze two hypotheses about the source of benefit provided thereby.

Label smoothing The first hypothesis is that the consistency loss may be able to mitigate over-confident predictions by calibrating the two output distributions against each other. This effect is similar to label smoothing (Szegedy et al., 2015; Yuan et al., 2020), which softens the one-hot target label by adding a loss of uniform distribution over all class labels and has proven helpful across a wide variety of models. To measure this, we plot the F1 improvement on the NER task for examples categorized by increasing predictive entropy in Fig. 5. MVR leads to more improvements on examples with higher entropy, or those that the model is more uncertain about, indicating that MVR is indeed helping the model improve on examples where it is not confident.

Ensemble effect The second hypothesis is that the consistency loss could regularize the model to be closer to the ensemble of models trained on standard deterministically segmented inputs and probabilistically segmented inputs. To verify this hypothesis, we first calculate the ensembled prediction probability of the baseline and the SR models for each language. Then we compare the KL divergence between this ensemble distribution and

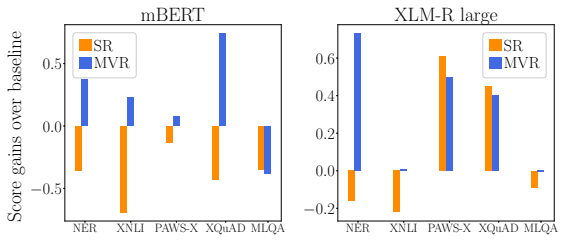

Figure 8: Gains of MVR and SR for English. While SR harms the performance on English, MVR generally improves it.

MVR with or without the consistency loss. In Fig. 7, we plot this KL divergence difference between the MVR without consistency loss and the full MVR for each language in NER. For most of the languages, the full MVR has lower KL divergence with the ensemble distribution, which indicates that the consistency loss trains the model to be closer to the ensemble of two inputs.

6.3 MVR also improves English

Although SR improves the model performance averaged over all languages, surprisingly it can hurt the performance on English, the language we use for fine-tuning. Fig. 8 shows the improvement on English over the baseline for both SR and MVR, and notably English performance decreases for all tasks on mBERT. MVR, on the other hand, generally brings improvements for English across both mBERT and XLM-R large models. This is likely because MVR also utilizes English inputs with standard segmentation, the method used at pretraining time, which allows it to take full advantage of the information encoded during pretraining.

7 Related work

Several works propose to optimize subword-sensitive word encoding methods for pretrained language models. Ma et al. (2020) uses convolutional neural networks (Kim, 2014) on characters to calculate word representations. Zhang and Li (2020) propose to add phrases into the vocabulary for Chinese pretrained language models. However, they focus on improving the vocabulary of pretrained representations of a single language, and they require modification to the model pretraining stage. Chung et al. (2020) propose to cluster related languages together and run subword vocabulary construction on each language cluster when constructing vocabularies for mBERT. Their method is also applied at the pretraining stage and could be combined with our method for potential additional improvements.

Our method is also related to prior work that optimize word representations for NMT and language modeling. Character level embeddings have been utilized instead of subword segmentation for NMT (Cherry et al., 2018; Lee et al., 2017; Ataman and Federico, 2018) and language modeling (Kim et al., 2016; Józefowicz et al., 2016). Wang et al. (2019) propose a multilingual word embedding method for NMT that relies on character n-gram embedding and a latent semantic embedding shared between different languages. Ataman and Federico (2018) show that character n-gram based embedding performs better than BPE for morphologically rich languages.

Our method is inspired by semi-supervised learning methods that enforce model consistency on unlabeled data. Several self-training methods utilize unlabeled examples to minimize the distance between the model predictions based on the unlabeled example and a noised version of the same input (Miyato et al., 2017b,a; Xu and Yang, 2017; Clark et al., 2018; Xie et al., 2018). Xu and Yang (2017) use knowledge distillation on unlabeled data to adapt models to a new language. Clark et al. (2018) propose to mask out different parts of the unlabeled input and encourage the model to make consistent prediction given these different inputs. These methods all focus on semi-supervised learning, while our method regulates model consistency to mitigate the subword segmentation discrepancy between different languages.

8 Conclusion

We believe that the results in this paper convincingly demonstrate that standard deterministic subword segmentation is sub-optimal for multilingual pretrained representations. Even incorporating simple methods for subword regularization such as BPE-dropout at fine-tuning can improve the cross-lingual transfer of pretrained models, and our proposed Multi-view Subword Regularization method further shows consistent and strong improvements over a variety of tasks for models built upon different subword segmentation algorithms. Going forward, we suggest that some variety of subword regularization, MVR or otherwise, should be a standard component of the fine-tuning of pre-trained representations that use subword segmentation.

References

Judit Ács. 2019. Exploring bert's vocabulary.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the Cross-lingual Transferability of Monolingual Representations. In *ACL*.

Duygu Ataman and Marcello Federico. 2018. Compositional representation of morphologically-rich input for neural machine translation. *ACL*.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of EMNLP*.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of EMNLP 2020*.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. *CoRR*.

Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving multilingual models with language-clustered vocabularies. In *EMNLP*.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *EMNLP*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.

Alexis Conneau and Guillaume Lample. 2019. Crosslingual language model pretraining. In *NeurIPS*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *EMNLP*, pages 2475–2485.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014. Distilling the knowledge in a neural network. In *NIPS Deep Learning Workshop*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *ICML*.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhoun. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. In *EMNLP*.

Rafal Józefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *Arxiv*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. *AAAI*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *TACL*.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating Cross-lingual Extractive Question Answering. In *ACL*.

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. Charbert: Character-aware pre-trained language model. In *COLING*.

Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2017a. Adversarial training methods for semi-supervised text classification. In *ICLR*.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2017b. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *ACL*, pages 1946–1958.

Jonas Pfeiffer, Ivan Vuli, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-based Framework for Multi-task Cross-lingual Transfer. In *Proceedings of EMNLP 2020*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *ACL*, Florence, Italy.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *ACL*.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *ICASSP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision. In *CVPR*.

Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. 2019. Multilingual neural machine translation with soft decoupled encoding. In *ICLR*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *EMNLP*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc Le. 2018. Unsupervised data augmentation for consistency training. In *EMNLP*.

Ruochen Xu and Yiming Yang. 2017. Cross-lingual distillation for text classification. In *ACL*, Vancouver, Canada.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *EMNLP*, pages 3685–3690.

Li Yuan, Francis E.H.Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisit knowledge distillation: a teacher-free framework. In *CVPR*.

Xinsong Zhang and Hang Li. 2020. Ambert: A pre-trained language model with multi-grained tokenization. In *arxiv*.

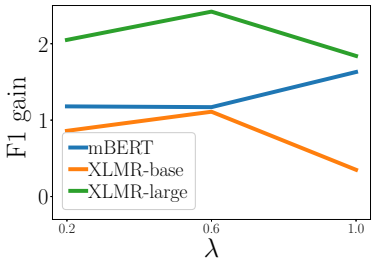

Figure 9: Average F1 gains over the baseline on NER task.

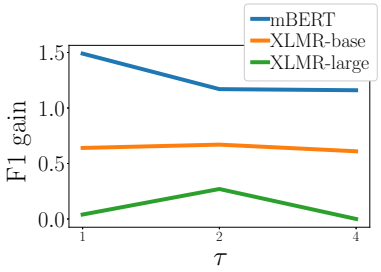

Figure 10: Average F1 gains over the baseline on QA tasks.

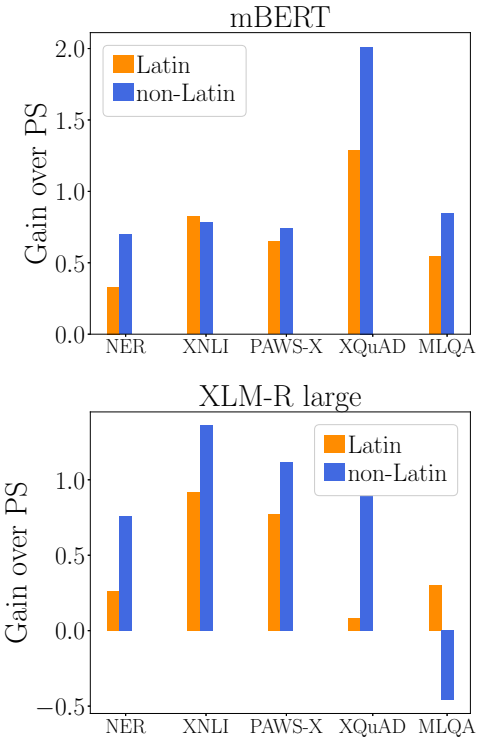

Figure 11: Gains of MVR over SR for languages with Latin vs. non-Latin script.

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.

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.