Consistency Regularization for Cross-Lingual Fine-Tuning

Bo Zheng[†]*, **Li Dong**[‡], **Shaohan Huang**[‡], **Wenhui Wang**[‡], **Zewen Chi**[‡]*
Saksham Singhal[‡], **Wanxiang Che**[†], **Ting Liu**[†], **Xia Song**[‡], **Furu Wei**[‡]
[†]Harbin Institute of Technology
[‡]Microsoft Corporation
{bzheng,car,tliu}@ir.hit.edu.cn
{lidong1,shaohanh,wenwan,saksingh,xiaso,fuwei}@microsoft.com

Abstract

Fine-tuning pre-trained cross-lingual language models can transfer task-specific supervision from one language to the others. In this work, we propose to improve cross-lingual fine-tuning with consistency regularization. Specifically, we use *example consistency* regularization to penalize the prediction sensitivity to four types of data augmentations, i.e., subword sampling, Gaussian noise, code-switch substitution, and machine translation. In addition, we employ *model consistency* to regularize the models trained with two augmented versions of the same training set. Experimental results on the XTREME benchmark show that our method[1] significantly improves cross-lingual fine-tuning across various tasks, including text classification, question answering, and sequence labeling.

1 Introduction

Pre-trained cross-lingual language models (Conneau and Lample, 2019; Conneau et al., 2020a; Chi et al., 2020) have shown great transferability across languages. By fine-tuning on labeled data in a source language, the models can generalize to other target languages, even without any additional training. Such generalization ability reduces the required annotation efforts, which is prohibitively expensive for low-resource languages.

Recent work has demonstrated that data augmentation is helpful for cross-lingual transfer, e.g., translating source language training data into target languages (Singh et al., 2019), and generating code-switch data by randomly replacing input words in the source language with translated words in target languages (Qin et al., 2020). By populating the dataset, their fine-tuning still treats training instances independently, without considering the inherent correlations between the original input and its augmented example. In contrast, we propose to utilize consistency regularization to better leverage data augmentation for cross-lingual fine-tuning. Intuitively, for a semantic-preserving augmentation strategy, the predicted result of the original input should be similar to its augmented one. For example, the classification predictions of an English sentence and its translation tend to remain consistent.

In this work, we introduce a cross-lingual fine-tuning method XTUNE that is enhanced by consistency regularization and data augmentation. First, *example consistency* regularization enforces the model predictions to be more consistent for semantic-preserving augmentations. The regularizer penalizes the model sensitivity to different surface forms of the same example (e.g., texts written in different languages), which implicitly encourages cross-lingual transferability. Second, we introduce *model consistency* to regularize the models trained with various augmentation strategies. Specifically, given two augmented versions of the same training set, we encourage the models trained on these two datasets to make consistent predictions for the same example. The method enforces the corpus-level consistency between the distributions learned by two models.

Under the proposed fine-tuning framework, we study four strategies of data augmentation, i.e., subword sampling (Kudo, 2018), code-switch substitution (Qin et al., 2020), Gaussian noise (Aghajanyan et al., 2020), and machine translation. We evaluate XTUNE on the XTREME benchmark (Hu et al., 2020), including three different tasks on seven datasets. Experimental results show that our method outperforms conventional fine-tuning with data augmentation. We also demonstrate that XTUNE is flexible to be plugged in various

*Contribution during internship at Microsoft Research.
[1]The code is available at https://github.com/bozheng-hit/xTune.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 3403–3417
August 1–6, 2021.

tasks, such as classification, span extraction, and sequence labeling.

We summarize our contributions as follows:

- We propose XTUNE, a cross-lingual fine-tuning method to better utilize data augmentations based on consistency regularization.
- We study four types of data augmentations that can be easily plugged into cross-lingual fine-tuning.
- We give instructions on how to apply XTUNE to various downstream tasks, such as classification, span extraction, and sequence labeling.
- We conduct extensive experiments to show that XTUNE consistently improves the performance of cross-lingual fine-tuning.

2 Related Work

Cross-Lingual Transfer Besides learning cross-lingual word embeddings (Mikolov et al., 2013; Faruqui and Dyer, 2014; Guo et al., 2015; Xu et al., 2018; Wang et al., 2019), most recent work of cross-lingual transfer is based on pre-trained cross-lingual language models (Conneau and Lample, 2019; Conneau et al., 2020a; Chi et al., 2020). These models generate multilingual contextualized word representations for different languages with a shared encoder and show promising cross-lingual transferability.

Cross-Lingual Data Augmentation Machine translation has been successfully applied to the cross-lingual scenario as data augmentation. A common way to use machine translation is to fine-tune models on both source language training data and translated data in all target languages. Furthermore, Singh et al. (2019) proposed to replace a segment of source language input text with its translation in another language. However, it is usually impossible to map the labels in source language data into target language translations for token-level tasks. Zhang et al. (2019) used code-mixing to perform the syntactic transfer in cross-lingual dependency parsing. Fei et al. (2020) constructed pseudo translated target corpora from the gold-standard annotations of the source languages for cross-lingual semantic role labeling. Fang et al. (2020) proposed an additional Kullback-Leibler divergence self-teaching loss for model training, based on auto-generated soft pseudo-labels for translated text in the target language. Besides, Qin et al. (2020) fine-tuned models on multilingual code-switch data, which achieves considerable improvements.

Consistency Regularization One strand of work in consistency regularization focused on regularizing model predictions to be invariant to small perturbations on image data. The small perturbations can be random noise (Zheng et al., 2016), adversarial noise (Miyato et al., 2019; Carmon et al., 2019) and various data augmentation approaches (Hu et al., 2017; Ye et al., 2019; Xie et al., 2020). Similar ideas are used in the natural language processing area. Both adversarial noise (Zhu et al., 2020; Jiang et al., 2020; Liu et al., 2020) and sampled Gaussian noise (Aghajanyan et al., 2020) are adopted to augment input word embeddings. Another strand of work focused on consistency under different model parameters (Tarvainen and Valpola, 2017; Athiwaratkun et al., 2019), which is complementary to the first strand. We focus on the cross-lingual setting, where consistency regularization has not been fully explored.

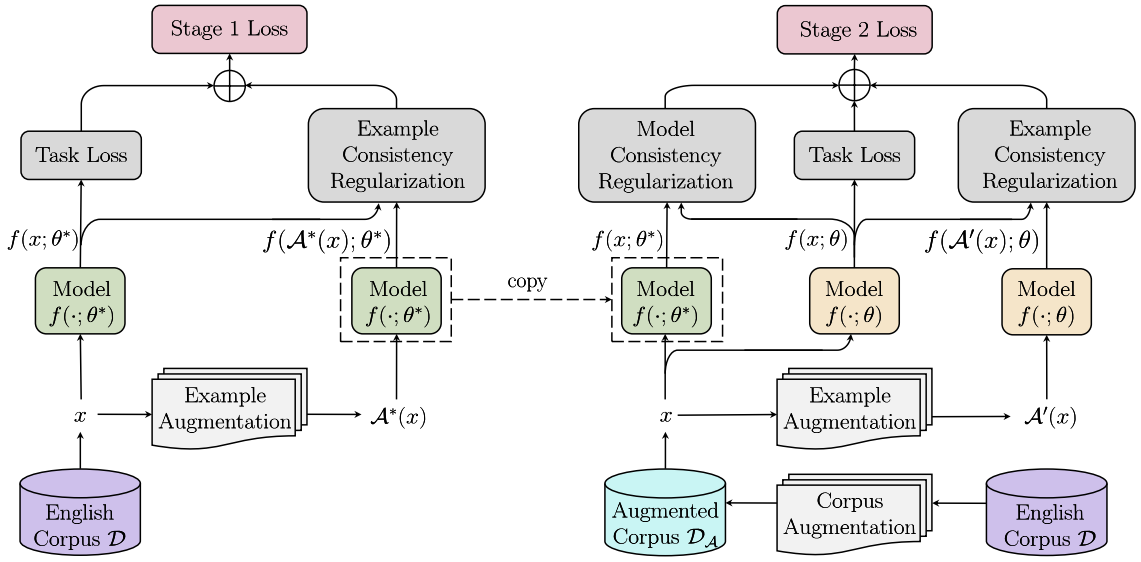
3 Methods

Conventional cross-lingual fine-tuning trains a pre-trained language model on the source language and directly evaluates it on other languages, which is also known as the setting of zero-shot cross-lingual fine-tuning. Specifically, given a training corpus $\mathcal{D}$ in the source language (typically in English), and a model $f(\cdot;\theta)$ that predicts task-specific probability distributions, we define the loss of cross-lingual fine-tuning as:

$$\mathcal{L}^{\text{task}}(\mathcal{D},\theta) = \sum_{x\in\mathcal{D}} \ell(f(x;\theta), G(x)),$$

where $G(x)$ denotes the ground-truth label of example x, $\ell(\cdot,\cdot)$ is the loss function depending on the downstream task.

Apart from vanilla cross-lingual fine-tuning on the source language, recent work shows that data augmentation is helpful to improve performance on the target languages. For example, Conneau and Lample (2019) add translated examples to the training set for better cross-lingual transfer. Let $\mathcal{A}(\cdot)$ be a cross-lingual data augmentation strategy (such as code-switch substitution), and $\mathcal{D}_{\mathcal{A}} = \mathcal{D} \cup \{\mathcal{A}(x) \mid x \in \mathcal{D}\}$ be the augmented training corpus, the fine-tuning loss is $\mathcal{L}^{\text{task}}(\mathcal{D}_{\mathcal{A}},\theta)$. Notice that it is non-trivial to apply some augmentations for token-level tasks directly. For instance, in part-of-speech



(a) The first stage in XTUNE (b) The second stage in XTUNE

Figure 1: Overview of our two-stage fine-tuning algorithm. The model parameters $f(\cdot;\theta^*)$ in the second stage are copied from the first stage.

tagging, the labels of source language examples can not be mapped to the translated examples because of the lack of explicit alignments.

3.1 XTUNE: Cross-Lingual Fine-Tuning with Consistency Regularization

We propose to improve cross-lingual fine-tuning with two consistency regularization methods, so that we can effectively leverage cross-lingual data augmentations.

3.1.1 Example Consistency Regularization

In order to encourage consistent predictions for an example and its semantically equivalent augmentation, we introduce *example consistency* regularization, which is defined as follows:

$$\mathcal{R}_1(\mathcal{D},\theta,\mathcal{A}) = \sum_{x\in\mathcal{D}} \mathrm{KL_S}(f(x;\theta)\|f(\mathcal{A}(x);\theta)),$$
$$\mathrm{KL_S}(P,Q) = \mathrm{KL}(\mathrm{stopgrad}(P)\|Q) + \mathrm{KL}(\mathrm{stopgrad}(Q)\|P)$$

where $\mathrm{KL_S}(\cdot)$ is the symmertrical Kullback-Leibler divergence. The regularizer encourages the predicted distributions $f(x;\theta)$ and $f(\mathcal{A}(x);\theta)$ to agree with each other. The $\mathrm{stopgrad}(\cdot)$ operation[2] is used to stop back-propagating gradients, which is also employed in (Jiang et al., 2020; Liu et al., 2020). The ablation studies in Section 4.2 empirically show that the operation improves fine-tuning performance.

[2]Implemented by `.detach()` in PyTorch.

3.1.2 Model Consistency Regularization

While the example consistency regularization is conducted at the example level, we propose the *model consistency* to further regularize the model training at the corpus level. The regularization is conducted at two stages. First, we obtain a fine-tuned model θ^* on the training corpus $\mathcal{D}$:

$$\theta^* = \arg\min_{\theta_1} \mathcal{L}^{\mathrm{task}}(\mathcal{D},\theta_1).$$

In the second stage, we keep the parameters θ^* fixed. The regularization term is defined as:

$$\mathcal{R}_2(\mathcal{D}_\mathcal{A},\theta,\theta^*) = \sum_{x\in\mathcal{D}_\mathcal{A}} \mathrm{KL}(f(x;\theta^*)\|f(x;\theta))$$

where $\mathcal{D}_\mathcal{A}$ is the augmented training corpus, and $\mathrm{KL}(\cdot)$ is Kullback-Leibler divergence. For each example x of the augmented training corpus $\mathcal{D}_\mathcal{A}$, the model consistency regularization encourages the prediction $f(x;\theta)$ to be consistent with $f(x;\theta^*)$. The regularizer enforces the corpus-level consistency between the distributions learned by two models.

An unobvious advantage of model consistency regularization is the flexibility with respect to data augmentation strategies. For the example of part-of-speech tagging, even though the labels can not be directly projected from an English sentence to its translation, we are still able to employ the regularizer. Because the term $\mathcal{R}_2$ is put on the same example $x \in \mathcal{D}_\mathcal{A}$, we can always align the token-level predictions of the models θ and θ^*.

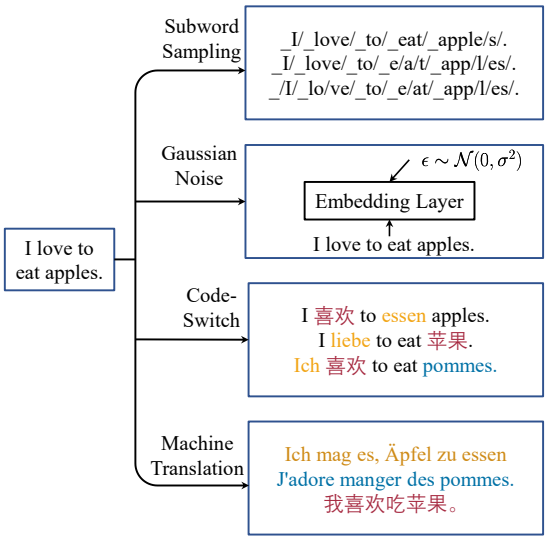


Figure 2: Cross-lingual data augmentation strategies.

3.1.3 Full xTune Fine-Tuning

As shown in Figure 1, we combine example consistency regularization $\mathcal{R}_1$ and model consistency regularization $\mathcal{R}_2$ as a two-stage fine-tuning process. Formally, we fine-tune a model with $\mathcal{R}_1$ in the first stage:

$$\theta^* = \arg\min_{\theta_1} \mathcal{L}^{\text{task}}(\mathcal{D}, \theta_1) + \mathcal{R}_1(\mathcal{D}, \theta_1, \mathcal{A}^*)$$

where the parameters θ^* are kept fixed for $\mathcal{R}_2$ in the second stage. Then the final loss is computed via:

$$\begin{aligned}\mathcal{L}^{\text{xTune}} = \; & \mathcal{L}^{\text{task}}(\mathcal{D}_{\mathcal{A}}, \theta) \\ & + \lambda_1 \mathcal{R}_1(\mathcal{D}_{\mathcal{A}}, \theta, \mathcal{A}') \\ & + \lambda_2 \mathcal{R}_2(\mathcal{D}_{\mathcal{A}}, \theta, \theta^*)\end{aligned}$$

where λ_1 and λ_2 are the corresponding weights of two regularization methods. Notice that the data augmentation strategies $\mathcal{A}$, $\mathcal{A}'$, and $\mathcal{A}^*$ can be either different or the same, which are tuned as hyper-parameters.

3.2 Data Augmentation

We consider four types of data augmentation strategies in this work, which are shown in Figure 2. We aim to study the impact of different data augmentation strategies on cross-lingual transferability.

3.2.1 Subword Sampling

Representing a sentence in different subword sequences can be viewed as a data augmentation strategy (Kudo, 2018; Provilkov et al., 2020). We utilize XLM-R (Conneau et al., 2020a) as our pre-trained cross-lingual language model, while it applies subword tokenization directly on raw text data using SentencePiece (Kudo and Richardson, 2018) with a unigram language model (Kudo, 2018). As one of our data augmentation strategies, we apply the on-the-fly subword sampling algorithm in the unigram language model to generate multiple subword sequences.

3.2.2 Gaussian Noise

Most data augmentation strategies in NLP change input text discretely, while we directly add random perturbation noise sampled from Gaussian distribution on the input embedding layer to conduct data augmentation. When combining this data augmentation with example consistency $\mathcal{R}_1$, the method is similar to the stability training (Zheng et al., 2016), random perturbation training (Miyato et al., 2019) and the R3F method (Aghajanyan et al., 2020). We also explore Gaussian noise's capability to generate new examples on continuous input space for conventional fine-tuning.

3.2.3 Code-Switch Substitution

Anchor points have been shown useful to improve cross-lingual transferability. Conneau et al. (2020b) analyzed the impact of anchor points in pre-training cross-lingual language models. Following Qin et al. (2020), we generate code-switch data in multiple languages as data augmentation. We randomly select words in the original text in the source language and replace them with target language words in the bilingual dictionaries to obtain code-switch data. Intuitively, this type of data augmentation explicitly helps pre-trained cross-lingual models align the multilingual vector space by the replaced anchor points.

3.2.4 Machine Translation

Machine translation has been proved to be an effective data augmentation strategy (Singh et al., 2019) under the cross-lingual scenario. However, the ground-truth labels of translated data can be unavailable for token-level tasks (see Section 3), which disables conventional fine-tuning on the augmented data. Meanwhile, our proposed model consistency $\mathcal{R}_2$ can not only serve as consistency regularization but also can be viewed as a self-training objective to enable semi-supervised training on the unlabeled target language translations.

3.3 Task Adaptation

We give instructions on how to apply xTune to various downstream tasks, i.e., classification, span extraction, and sequence labeling. By default, we use model consistency $\mathcal{R}_2$ in full xTune. We describe the usage of example consistency $\mathcal{R}_1$ as follows.

3.3.1 Classification

For classification task, the model is expected to predict one distribution per example on n_{label} types, i.e., model $f(\cdot;\theta)$ should predict a probability distribution $p_{\text{cls}} \in \mathbb{R}^{n_{\text{label}}}$. Thus we can directly use example consistency $\mathcal{R}_1$ to regularize the consistency of the two distributions for all four types of our data augmentation strategies.

3.3.2 Span Extraction

For span extraction task, the model is expected to predict two distributions per example $p_{\text{start}}, p_{\text{end}} \in \mathbb{R}^{n_{\text{subword}}}$, indicating the probability distribution of where the answer span starts and ends, n_{subword} denotes the length of the tokenized input text. For Gaussian noise, the subword sequence remains unchanged so that example consistency $\mathcal{R}_1$ can be directly applied to the two distributions. Since subword sampling and code-switch substitution will change n_{subword}, we control the ratio of words to be modified and utilize example consistency $\mathcal{R}_1$ on unchanged positions only. We do not use the example consistency $\mathcal{R}_1$ for machine translation because it is impossible to explicitly align the two distributions.

3.3.3 Sequence Labeling

Recent pre-trained language models generate representations at the subword-level. For sequence labeling tasks, these models predict label distributions on each word's first subword. Therefore, the model is expected to predict n_{word} probability distributions per example on n_{label} types. Unlike span extraction, subword sampling, code-switch substitution, and Gaussian noise do not change n_{word}. Thus the three data augmentation strategies will not affect the usage of example consistency $\mathcal{R}_1$. Although word alignment is a possible solution to map the predicted label distributions between translation pairs, the word alignment process will introduce more noise. Therefore, we do not employ machine translation as data augmentation for the example consistency $\mathcal{R}_1$.

4 Experiments

4.1 Experiment Setup

Datasets For our experiments, we select three types of cross-lingual understanding tasks from XTREME benchmark (Hu et al., 2020), including two classification datasets: XNLI (Conneau et al., 2018), PAWS-X (Yang et al., 2019), three span extraction datasets: XQuAD (Artetxe et al., 2020), MLQA (Lewis et al., 2020), TyDiQA-GoldP (Clark et al., 2020), and two sequence labeling datasets: NER (Pan et al., 2017), POS (Nivre et al., 2018). The statistics of the datasets are shown in the supplementary document.

Fine-Tuning Settings We consider two typical fine-tuning settings from Conneau et al. (2020a) and Hu et al. (2020) in our experiments, which are (1) *cross-lingual transfer*: the models are fine-tuned on English training data without translation available, and directly evaluated on different target languages; (2) *translate-train-all*: translation-based augmentation is available, and the models are fine-tuned on the concatenation of English training data and its translated data on all target languages. Since the official XTREME repository[3] does not provide translated target language data for POS and NER, we use Google Translate to obtain translations for these two datasets.

Implementation Details We utilize XLM-R (Conneau et al., 2020a) as our pre-trained cross-lingual language model. The bilingual dictionaries we used for code-switch substitution are from MUSE (Lample et al., 2018).[4] For languages that cannot be found in MUSE, we ignore these languages since other bilingual dictionaries might be of poorer quality. For the POS dataset, we use the average-pooling strategy on subwords to obtain word representation since part-of-speech is related to different parts of words, depending on the language. We tune the hyper-parameter and select the model with the best average results over all the languages' development set. There are two datasets without development set in multi-languages. For XQuAD, we tune the hyper-parameters with the development set of MLQA since they share the same training set and have a higher degree of overlap in languages. For TyDiQA-GoldP, we use the English test set

[3] github.com/google-research/xtreme
[4] github.com/facebookresearch/MUSE

Model	Pair Sentence		Structure Prediction		Question Answering			
	XNLI	PAWS-X	POS	NER	XQuAD	MLQA	TyDiQA	
Metrics	Acc.	Acc.	F1	F1	F1/EM	F1/EM	F1/EM	Avg.
Cross-lingual-transfer (models are fine-tuned on English training data without translation available)								
mBERT	65.4	81.9	70.3	62.2	64.5/49.4	61.4/44.2	59.7/43.9	63.1
XLM	69.1	80.9	70.1	61.2	59.8/44.3	48.5/32.6	43.6/29.1	58.6
X-STILTs (Phang et al., 2020)	80.4	87.7	74.4	63.4	77.2/61.3	72.3/53.5	76.0/59.5	72.3
VECO (Luo et al., 2020)	79.9	88.7	75.1	65.7	77.3/61.8	71.7/53.2	67.6/49.1	71.4
XLM-R_{large}	79.2	86.4	72.6	65.4	76.6/60.8	71.6/53.2	65.1/45.0	70.0
XTUNE	**82.6**	**89.8**	**78.5**	**69.3**	**79.4/64.4**	**74.4/56.2**	**74.8/59.4**	**74.9**
Translate-train-all (translation-based augmentation is available for English training data)								
VECO (Luo et al., 2020)	83.0	91.1	75.1	65.7	79.9/66.3	73.1/54.9	75.0/58.9	74.1
FILTER (Fang et al., 2020)	83.9	91.4	76.2	67.7	82.4/68.0	**76.2/57.7**	68.3/50.9	74.4
XLM-R_{large}	82.6	90.4	-	-	80.2/65.9	72.8/54.3	66.5/47.7	-
XTUNE	**84.8**	**91.6**	**79.3**	**69.9**	**82.5/69.0**	75.0/57.1	**75.4/60.8**	**76.5**

Table 1: Evaluation results on the XTREME benchmark. Results of mBERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019) and XLM-R_{large} (Conneau et al., 2020a) are taken from (Hu et al., 2020). Results of XLM-R_{large} under the translate-train-all setting are from FILTER (Fang et al., 2020). The results of XTUNE are from the best models selected with the performance on the corresponding development set.

Model	Pair Sentence		Structure Prediction		Question Answering		
	XNLI	PAWS-X	POS	NER	XQuAD	MLQA	TyDiQA
Metrics	Acc.	Acc.	F1	F1	F1/EM	F1/EM	F1/EM
Cross-lingual-transfer (models are fine-tuned on English training data without translation available)							
XLM-R_{base}	74.9	84.9	75.6	61.8	71.9/56.4	65.0/47.1	55.4/38.3
XTUNE	**77.7**	**87.5**	**76.5**	**63.0**	**73.9/59.0**	**68.1/50.2**	**61.2/45.2**
with only *example consistency* $\mathcal{R}_1$	77.6	87.2	76.3	62.4	73.6/58.6	67.6/49.7	60.7/44.4
with only *model consistency* $\mathcal{R}_2$	76.6	86.3	76.3	**63.0**	73.2/58.1	66.7/49.0	59.2/42.3
Translate-train-all (translation-based augmentation is available for English training data)							
XLM-R_{base}	78.8	88.4	-	-	75.2/61.4	67.8/50.1	63.7/47.7
XTUNE	**80.6**	**89.4**	**77.8**	**63.7**	**78.1/64.4**	69.7/**52.1**	**65.9/51.1**
with only *example consistency* $\mathcal{R}_1$	80.5	89.3	-	-	76.1/62.5	69.1/51.6	65.1/50.3
with only *model consistency* $\mathcal{R}_2$	78.9	88.5	76.6	63.5	77.4/63.4	68.7/51.1	64.5/48.7
remove stopgrad in $\mathcal{R}_1$	80.2	89.1	76.8	63.4	77.3/63.4	**69.9/52.1**	65.1/50.5

Table 2: Ablation studies on the XTREME benchmark. All numbers are averaged over five random seeds.

as the development set. In order to make a fair comparison, the ratio of data augmentation in $\mathcal{D}_{\mathcal{A}}$ is all set to 1.0. The detailed hyper-parameters are shown in the supplementary document.

4.2 Results

Table 1 shows our results on XTREME. For the cross-lingual transfer setting, we outperform previous works on all seven cross-lingual language understanding datasets.[5] Compared to XLM-R_{large} baseline, we achieve an absolute 4.9-point improvement (70.0 vs. 74.9) on average over seven datasets. For the translate-train-all setting, we achieved state-of-the-art results on six of the seven datasets. Compared to FILTER,[6] we achieve an absolute 2.1-point improvement (74.4 vs. 76.5), and we do not need English translations during inference.

Table 2 shows how the two regularization methods affect the model performance separately. For the cross-lingual transfer setting, XTUNE achieves an absolute 2.8-point improvement compared to our implemented XLM-R_{base} baseline. Meanwhile, fine-tuning with only example consistency $\mathcal{R}_1$ and model consistency $\mathcal{R}_2$ degrades the averaged results by 0.4 and 1.0 points, respectively.

For the translate-train-all setting, our proposed model consistency $\mathcal{R}_2$ enables training on POS and NER even if labels of target language translations

[5] X-STILTs (Phang et al., 2020) uses additional SQuAD v1.1 English training data for the TyDiQA-GoldP dataset, while we prefer a cleaner setting here.

[6] FILTER directly selects the best model on the test set of XQuAD and TyDiQA-GoldP. Under this setting, we can obtain 83.1/69.7 for XQuAD, 75.5/61.1 for TyDiQA-GoldP.

Model	en	ar	bg	de	el	es	fr	hi	ru	sw	th	tr	ur	vi	zh	Avg.
Cross-lingual-transfer (models are fine-tuned on English training data without translation available)																
R3F (Aghajanyan et al., 2020)	89.4	80.6	84.6	83.7	83.6	85.1	84.2	77.3	82.3	72.6	79.4	80.7	74.2	81.1	80.1	81.2
R4F (Aghajanyan et al., 2020)	**89.6**	80.5	84.6	84.2	83.6	85.2	84.7	78.2	82.5	72.7	79.2	80.3	73.9	80.9	80.6	81.4
XLM-R_{large}	88.7	77.2	83.0	82.5	80.8	83.7	82.2	75.6	79.1	71.2	77.4	78.0	71.7	79.3	78.2	79.2
XTUNE	**89.6**	**81.6**	**85.9**	**84.8**	**84.3**	**86.5**	**85.4**	**80.5**	**82.8**	**73.3**	**80.3**	**82.1**	**77.1**	**83.0**	**82.3**	**82.6**
Translate-train-all (translation-based augmentation is available for English training data)																
FILTER (Fang et al., 2020)	89.5	83.6	86.4	85.6	85.4	86.6	85.7	81.1	83.7	78.7	81.7	83.2	79.1	83.9	83.8	83.9
XLM-R_{large}	88.6	82.2	85.2	84.5	84.5	85.7	84.2	80.8	81.8	77.0	80.2	82.1	77.7	82.6	82.7	82.6
XTUNE	**89.9**	**84.0**	**87.0**	**86.5**	**86.2**	**87.4**	**86.6**	**83.2**	**85.2**	**80.0**	**82.7**	**84.1**	**79.6**	**84.8**	**84.3**	**84.8**

Table 3: XNLI accuracy scores for each language. XLM-R_{large} under the cross-lingual transfer setting are from (Hu et al., 2020). Results of XLM-R_{large} under the translate-train-all setting are from (Fang et al., 2020).

Method	Model	XNLI	POS	MLQA
Baseline	XLM-R_{base}	74.9	75.6	65.0/47.1
Subword Sampling	Data Aug.	75.3	75.8	64.7/46.7
	XTUNE$_{\mathcal{R}_1}$	76.5	76.3	67.4/49.5
	XTUNE$_{\mathcal{R}_2}$	75.8	76.3	66.7/49.0
Gaussian Noise	Data Aug.	74.7	75.6	64.2/46.1
	XTUNE$_{\mathcal{R}_1}$	76.3	75.7	66.7/48.9
	XTUNE$_{\mathcal{R}_2}$	75.5	76.2	66.3/48.5
Code-Switch	Data Aug.	76.5	75.1	63.8/45.9
	XTUNE$_{\mathcal{R}_1}$	77.6	75.8	67.6/49.7
	XTUNE$_{\mathcal{R}_2}$	76.8	76.1	66.3/48.6
Machine Translation	Data Aug.	78.8	-	67.8/50.1
	XTUNE$_{\mathcal{R}_1}$	**79.7**	-	-
	XTUNE$_{\mathcal{R}_2}$	78.9	**76.6**	**68.7/51.1**

Table 4: Comparison between different data augmentation strategies. "Data Aug." uses data augmentation for conventional fine-tuning. "XTUNE$_{\mathcal{R}_1}$" denotes fine-tuning with only example consistency $\mathcal{R}_1$. "XTUNE$_{\mathcal{R}_2}$" denotes fine-tuning with only model consistency $\mathcal{R}_2$.

Model	Tatoeba	BUCC
XLM-R_{base} (*cross-lingual transfer*)	74.2	78.2
XLM-R_{base} (*translate-train-all*)	79.7	79.7
XTUNE (*translate-train-all*)	**82.3**	**82.2**
with only *example consistency* $\mathcal{R}_1$	82.0	82.1
with only *model consistency* $\mathcal{R}_2$	79.5	79.0

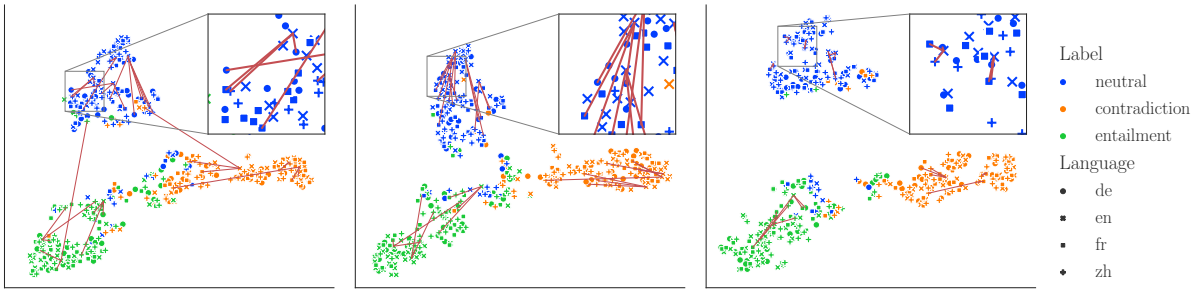
Table 5: Results of cross-lingual retrieval with the models fine-tuned on XNLI.

are unavailable in these two datasets. To make a fair comparison in the translate-train-all setting, we augment the English training corpus with target language translations when fine-tuning with only example consistency $\mathcal{R}_1$. Otherwise, we only use the English training corpus in the first stage, as shown in Figure 1(a). Compared to XTUNE, the performance drop on two classification datasets under this setting is relatively small since $\mathcal{R}_1$ can be directly applied between translation-pairs in any languages. However, the performance is significantly degraded in three question answering datasets, where we can not align the predicted distributions between translation-pairs in $\mathcal{R}_1$. We use subword sampling as the data augmentation strategy in $\mathcal{R}_1$ for this situation. Fine-tuning with only model consistency $\mathcal{R}_2$ degrades the overall performance by 1.1 points. These results demonstrate that the two consistency regularization methods complement each other. Besides, we observe that removing stopgrad degrades the overall performance by 0.5 points.

Table 3 provides results of each language on the XNLI dataset. For the cross-lingual transfer setting, we utilize code-switch substitution as data augmentation for both example consistency $\mathcal{R}_1$ and model consistency $\mathcal{R}_2$. We utilize all the bilingual dictionaries, except for English to Swahili and English to Urdu, which MUSE does not provide. Results show that our method outperforms all baselines on each language, even on Swahili (+2.2 points) and Urdu (+5.4 points), indicating our method can be generalized to low-resource languages even without corresponding machine translation systems or bilingual dictionaries. For translate-train-all setting, we utilize machine translation as data augmentation for both example consistency $\mathcal{R}_1$ and model consistency $\mathcal{R}_2$. We improve the XLM-R_{large} baseline by +2.2 points on average, while we still have +0.9 points on average compared to FILTER. It is worth mentioning that we do not need corresponding English translations during inference. Complete results on other datasets are provided in the supplementary document.

4.3 Analysis

It is better to employ data augmentation for consistency regularization than for conventional fine-tuning. As shown in Table 4, com-



(a) *cross-lingual transfer* (b) *translate-train-all* (c) XTUNE

Figure 3: t-SNE visualization of 100 examples in four languages from the XNLI development set (best viewed in color). We fine-tune the XLM-R_{base} model on XNLI and use the hidden states of `[CLS]` symbol in the last layer. Examples with different labels are represented with different colors. Examples in different languages are represented with different markers. The red lines connect English examples and their translations in target languages.

pared to employing data augmentation for conventional fine-tuning (Data Aug.), our regularization methods (XTUNE$_{\mathcal{R}_1}$, XTUNE$_{\mathcal{R}_2}$) consistently improve the model performance under all four data augmentation strategies. Since there is no labeled data on translations in POS and the issue of distribution alignment in example consistency $\mathcal{R}_1$, when machine translation is utilized as data augmentation, the results for Data Aug. and XTUNE$_{\mathcal{R}_1}$ in POS, as well as XTUNE$_{\mathcal{R}_1}$ in MLQA, are unavailable. We observe that Data Aug. can enhance the overall performance for coarse-grained tasks like XNLI, while our methods can further improve the results. However, Data Aug. even causes the performance to degrade for fine-grained tasks like MLQA and POS. In contrast, our proposed two consistency regularization methods improve the performance by a large margin (e.g., for MLQA under code-switch data augmentation, Data Aug. decreases baseline by 1.2 points, while XTUNE$_{\mathcal{R}_1}$ increases baseline by 2.6 points). We give detailed instructions on how to choose data augmentation strategies for XTUNE in the supplementary document.

XTUNE improves cross-lingual retrieval. We fine-tune the models on XNLI with different settings and compare their performance on two cross-lingual retrieval datasets. Following Chi et al. (2020) and Hu et al. (2020), we utilize representations averaged with hidden-states on the layer 8 of XLM-R_{base}. As shown in Table 5, we observe significant improvement from the translate-train-all baseline to fine-tuning with only example consistency $\mathcal{R}_1$, this suggests regularizing the task-specific output of translation-pairs to be consistent also encourages the model to generate language-invariant representations. XTUNE only slightly improves upon this setting, indicating $\mathcal{R}_1$ between translation-pairs is the most important factor to improve cross-lingual retrieval task.

XTUNE improves decision boundaries as well as the ability to generate language-invariant representations. As shown in Figure 3, we present t-SNE visualization of examples from the XNLI development set under three different settings. We observe the model fine-tuned with XTUNE significantly improves the decision boundaries of different labels. Besides, for an English example and its translations in other languages, the model fine-tuned with XTUNE generates more similar representations compared to the two baseline models. This observation is also consistent with the cross-lingual retrieval results in Table 5.

5 Conclusion

In this work, we present a cross-lingual fine-tuning framework XTUNE to make better use of data augmentation. We propose two consistency regularization methods that encourage the model to make consistent predictions for an example and its semantically equivalent data augmentation. We explore four types of cross-lingual data augmentation strategies. We show that both example and model consistency regularization considerably boost the performance compared to directly fine-tuning on data augmentations. Meanwhile, model consistency regularization enables semi-supervised training on the unlabeled target language translations. XTUNE combines the two regularization methods, and the experiments show that it can improve the performance by a large margin on the XTREME benchmark.

Acknowledgments

Wanxiang Che is the corresponding author. This work was supported by the National Key R&D Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grant 61976072 and 61772153.

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2020. Better fine-tuning by reducing representational collapse. *CoRR*, abs/2008.03156.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4623–4637. Association for Computational Linguistics.

Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. 2019. There are many consistent explanations of unlabeled data: Why you should average. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C. Duchi, and Percy Liang. 2019. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 11190–11201.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. *CoRR*, abs/2007.07834.

Jonathan H. Clark, Jennimaria Palomaki, Vitaly Nikolaev, Eunsol Choi, Dan Garrette, Michael Collins, and Tom Kwiatkowski. 2020. Tydi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Trans. Assoc. Comput. Linguistics*, 8:454–470.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7057–7067.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2475–2485. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6022–6034. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2020. FILTER: An enhanced fusion method for cross-lingual language understanding. *CoRR*, abs/2009.05166.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 462–471. The Association for Computer Linguistics.

Hao Fei, Meishan Zhang, and Donghong Ji. 2020. Cross-lingual semantic role labeling with high-quality translated training corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7014–7026. Association for Computational Linguistics.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1234–1244. The Association for Computer Linguistics.

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 generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. 2017. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1558–1567. PMLR.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2177–2190. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Patrick S. H. Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7315–7330. Association for Computational Linguistics.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. *CoRR*, abs/2004.08994.

Fuli Luo, W. Wang, Jiahao Liu, Yijia Liu, Bin Bi, Songfang Huang, Fei Huang, and L. Si. 2020. VECO: Variable encoder-decoder pre-training for cross-lingual understanding and generation. *ArXiv*, abs/2010.16046.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2019. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1979–1993.

Joakim Nivre, Rogier Blokland, Niko Partanen, and Michael Rießler. 2018. Universal dependencies 2.2.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1946–1958. Association for Computational Linguistics.

Jason Phang, Phu Mon Htut, Yada Pruksachatkun, Haokun Liu, Clara Vania, Katharina Kann, Iacer Calixto, and Samuel R. Bowman. 2020. English intermediate-task training improves zero-shot cross-lingual transfer too. *CoRR*, abs/2005.13013.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-Dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1882–1892. Association for Computational Linguistics.

Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. CoSDA-ML: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3853–3860. ijcai.org.

Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. XLDA: cross-lingual data augmentation for natural language inference and question answering. *CoRR*, abs/1905.11471.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. 2019. Cross-lingual BERT transformation for zero-shot dependency parsing. In

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 5720–5726. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ruochen Xu, Yiming Yang, Naoki Otani, and Yuexin Wu. 2018. Unsupervised cross-lingual transfer of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2465–2474. Association for Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3685–3690. Association for Computational Linguistics.

Mang Ye, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang. 2019. Unsupervised embedding learning via invariant and spreading instance feature. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6210–6219. Computer Vision Foundation / IEEE.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2019. Cross-lingual dependency parsing using code-mixed treebank. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 997–1006. Association for Computational Linguistics.

Stephan Zheng, Yang Song, Thomas Leung, and Ian J. Goodfellow. 2016. Improving the robustness of deep neural networks via stability training. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4480–4488. IEEE Computer Society.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. FreeLB: Enhanced adversarial training for natural language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Appendix

A Statistics of XTREME Datasets

Task	Dataset	|Train|	|Lang|
Classification	XNLI	392K	15
	PAWS-X	49.4K	7
Structured Prediction	POS	21K	33
	NER	20K	40
Question Answering	XQuAD	87K	11
	MLQA	87K	7
	TyDiQA	3.7K	9

Table 6: Statistics for the datasets in the XTREME benchmark. we report the number of training examples (|Train|), and the number of languages (|Lang|).

B Hyper-Parameters

For XNLI, PAWS-X, POS and NER, we fine-tune 10 epochs. For XQuAD and MLQA, we fine-tune 4 epochs. For TyDiQA-GoldP, we fine-tune 20 epochs and 10 epochs for base and large model, respectively. We select λ_1 in [1.0, 2.0, 5.0], λ_2 in [0.3, 0.5, 1.0, 2.0, 5.0]. For learning rate, we select in [5e-6, 7e-6, 1e-5, 1.5e-5] for large models, [7e-6, 1e-5, 2e-5, 3e-5] for base models. We use batch size 32 for all datasets and 10% of total training steps for warmup with a linear learning rate schedule. Our experiments are conducted with a single 32GB Nvidia V100 GPU, and we use gradient accumulation for large-size models. The other hyper-parameters for the two-stage XTUNE training are shown in Table 7 and Table 8.

C Results for Each Dataset and Language

We provide detailed results for each dataset and language below. We compare our method against XLM-R_{large} for cross-lingual transfer setting, FILTER (Fang et al., 2020) for translate-train-all setting.

D How to Select Data Augmentation Strategies in XTUNE

We give instructions on selecting a proper data augmentation strategy depending on the corresponding task.

	Variable	XNLI	PAWS-X	POS	NER	XQuAD	MLQA	TyDiQA
Stage 1	$\mathcal{A}^*$	CS	CS	SS	SS	CS	CS	SS
Stage 2	$\mathcal{A}$	CS	CS	SS	SS	SS	SS	SS
	$\mathcal{A}'$	CS	CS	SS	SS	SS	SS	SS
Hyper-parameters	λ_1	5.0	5.0	5.0	5.0	5.0	5.0	5.0
	λ_2	5.0	2.0	0.3	5.0	5.0	5.0	5.0

Table 7: The best hyper-parameters used for XTUNE under the cross-lingual transfer setting. "SS", "CS", "MT" denote the data augmentation methods: subword sampling, code-switch substitution, and machine translation, respectively.

	Variable	XNLI	PAWS-X	POS	NER	XQuAD	MLQA	TyDiQA
Stage 1	$\mathcal{A}^*$	MT	MT	SS	SS	CS	CS	SS
Stage 2	$\mathcal{A}$	MT	MT	MT	MT	MT	MT	MT
	$\mathcal{A}'$	MT	MT	SS	SS	SS	SS	SS
Hyper-parameters	λ_1	5.0	5.0	5.0	5.0	5.0	5.0	5.0
	λ_2	1.0	1.0	0.3	1.0	0.1	0.5	0.3

Table 8: The best hyper-parameters used for XTUNE under the translate-train-all setting. "SS", "CS", "MT" denote the data augmentation methods subword sampling, code-switch substitution, and machine translation, respectively.

Method	Model	XNLI	POS	MLQA	Avg.
-	XLM-R_{base}	10.6	20.8	20.3	17.2
Subword Sampling	Data Aug.	10.5	20.5	20.2	17.1
	XTUNE$_{\mathcal{R}_1}$	10.2	20.2	19.6	16.7
	XTUNE$_{\mathcal{R}_2}$	10.6	**20.1**	19.8	16.8
Gaussian Noise	Data Aug.	10.8	20.6	19.8	17.1
	XTUNE$_{\mathcal{R}_1}$	10.5	20.7	19.8	17.0
	XTUNE$_{\mathcal{R}_2}$	10.8	20.2	19.7	16.9
Code-Switch	Data Aug.	9.2	21.1	20.5	16.9
	XTUNE$_{\mathcal{R}_1}$	9.1	20.7	**19.4**	16.4
	XTUNE$_{\mathcal{R}_2}$	**8.8**	20.2	20.0	**16.3**
Machine Translation	Data Aug.	7.2	-	17.9	-
	XTUNE$_{\mathcal{R}_1}$	**6.9**	-	-	-
	XTUNE$_{\mathcal{R}_2}$	7.2	**19.6**	**17.1**	**14.6**

Table 9: Cross-lingual transfer gap, i.e., averaged performance drop between English and other languages in zero-shot transfer. A smaller gap indicates better transferability. For MLQA, we report the average of F1-scores and exact match scores.

D.1 Classification

The two distribution in example consistency $\mathcal{R}_1$ can always be aligned. Therefore, we recommend using machine translation as data augmentation if the machine translation systems are available. Otherwise, the priority of our data augmentation strategies is code-switch substitution, subword sampling and Gaussian noise.

D.2 Span Extraction

The two distribution in example consistency $\mathcal{R}_1$ can not be aligned in translation-pairs. Therefore, it is impossible to use machine translation as data augmentation in example consistency $\mathcal{R}_1$. We prefer to use code-switch when applying example consistency $\mathcal{R}_1$ individually. However, when the training corpus is augmented with translations, since the bilingual dictionaries between arbitrary language pairs may not be available, we recommend using subword sampling in example consistency $\mathcal{R}_1$.

D.3 Sequence Labeling

Similar to span extraction, the two distribution in example consistency $\mathcal{R}_1$ can not be aligned in translation-pairs. Therefore, we do not use machine translation in example consistency $\mathcal{R}_1$. Unlike classification and span extraction, sequence labeling requires finer-grained information and is more sensitive to noise. We found code-switch is worse than subword sampling as data augmentation in both example consistency $\mathcal{R}_1$ and model consistency $\mathcal{R}_2$, it will even degrade performance for certain hyper-parameters. Thus we recommend using subword sampling in example consistency $\mathcal{R}_1$, and use machine translation to augment the English training corpus if machine translation systems are available, otherwise subword sampling.

E Cross-Lingual Transfer Gap

As shown in Table 9, the cross-lingual transfer gap can be reduced under all four data augmentation strategies. Meanwhile, we observe machine translation and code-switch substitution achieve a smaller cross-lingual transfer gap than the other two data augmentation methods. This suggests the data augmentation methods with cross-lingual knowledge have a greater improvement in cross-lingual transferability. Although code-switch significantly reduces the transfer gap on XNLI, the improvement is relatively small on POS and MLQA under the cross-lingual transfer setting, indicating the noisy code-switch substitution will harm the cross-lingual transferability on finer-grained tasks.

Model	en	de	es	fr	ja	ko	zh	Avg.
Cross-lingual-transfer (models are fine-tuned on English training data without translation available)								
XLM-R_{large}	94.7	89.7	90.1	90.4	78.7	79.0	82.3	86.4
XTUNE	96.0	92.5	92.2	92.7	84.9	84.2	86.6	89.8
Translate-train-all (translation-based augmentation is available for English training data)								
FILTER (Fang et al., 2020)	95.9	92.8	93.0	93.7	87.4	87.6	89.6	91.5
XTUNE	96.1	92.6	93.1	93.9	87.8	89.0	88.8	91.6

Table 10: PAWSX results (accuracy scores) for each language.

Model	en	ar	de	el	es	hi	ru	th	tr	vi	zh	Avg.
Cross-lingual-transfer (models are fine-tuned on English training data without translation available)												
XLM-R_{large}	86.5/75.7	68.6/49.0	80.4/63.4	79.8/61.7	82.0/63.9	76.7/59.7	80.1/64.3	74.2/62.8	75.9/59.3	79.1/59.0	59.3/50.0	76.6/60.8
XTUNE	88.9/78.6	77.1/60.0	83.1/67.2	82.6/66.0	83.0/65.1	77.8/61.8	80.8/64.8	73.5/62.1	77.6/62.0	81.8/62.5	67.7/58.4	79.4/64.4
Translate-train-all (translation-based augmentation is available for English training data)												
FILTER (Fang et al., 2020)	86.4/74.6	79.5/60.7	83.2/67.0	83.0/64.6	85.0/67.9	83.1/66.6	82.8/67.4	79.6/73.2	80.4/64.4	83.8/64.7	79.9/77.0	82.4/68.0
XTUNE	88.8/78.1	79.7/63.9	83.7/68.2	83.0/65.7	84.7/68.3	80.7/64.9	82.2/66.6	81.9/76.1	79.3/65.0	82.7/64.5	81.3/78.0	82.5/69.0

Table 11: XQuAD results (F1/EM scores) for each language.

Model	en	ar	de	es	hi	vi	zh	Avg.
Cross-lingual-transfer (models are fine-tuned on English training data without translation available)								
XLM-R_{large}	83.5/70.6	66.6/47.1	70.1/54.9	74.1/56.6	70.6/53.1	74.0/52.9	62.1/37.0	71.6/53.2
XTUNE	85.2/72.6	67.9/47.7	72.2/56.8	75.5/57.9	73.2/55.1	75.9/54.7	71.1/48.6	74.4/56.2
Translate-train-all (translation-based augmentation is available for English training data)								
FILTER (Fang et al., 2020)	84.0/70.8	72.1/51.1	74.8/60.0	78.1/60.1	76.0/57.6	78.1/57.5	70.5/47.0	76.2/57.7
XTUNE	85.3/72.9	69.7/50.1	72.3/57.3	76.3/58.8	74.0/56.0	76.5/55.9	70.8/48.3	75.0/57.1

Table 12: MLQA results (F1/EM scores) for each language.

Model	en	ar	bn	fi	id	ko	ru	sw	te	Avg.
Cross-lingual-transfer (models are fine-tuned on English training data without translation available)										
XLM-R_{large}	71.5/56.8	67.6/40.4	64.0/47.8	70.5/53.2	77.4/61.9	31.9/10.9	67.0/42.1	66.1/48.1	70.1/43.6	65.1/45.0
XTUNE	75.3/63.6	77.4/60.3	72.4/58.4	75.5/60.2	81.5/68.5	68.6/58.3	71.1/48.8	73.3/56.7	78.4/60.1	74.8/59.4
Translate-train-all (translation-based augmentation is available for English training data)										
FILTER (Fang et al., 2020)	72.4/59.1	72.8/50.8	70.5/56.6	73.3/57.2	76.8/59.8	33.1/12.3	68.9/46.6	77.4/65.7	69.9/50.4	68.3/50.9
XTUNE	73.8/61.6	77.8/60.2	73.5/61.1	77.0/62.2	80.8/68.1	66.9/56.5	72.1/51.9	77.9/65.3	77.6/60.7	75.3/60.8

Table 13: TyDiQA-GolP results (F1/EM scores) for each language.

Model	af	ar	bg	de	el	en	es	et	eu	fa	fi	fr	he	hi	hu	id	it
Cross-lingual-transfer (models are fine-tuned on English training data without translation available)																	
XLM-R$_{\text{large}}$	89.8	67.5	88.1	88.5	86.3	96.1	88.3	86.5	72.5	70.6	85.8	87.2	68.3	76.4	82.6	72.4	89.4
XTUNE	90.4	72.8	89.0	89.4	87.0	96.1	88.8	88.1	73.1	74.7	87.2	89.5	83.5	77.7	83.6	73.2	90.5
Translate-train-all (translation-based augmentation is available for English training data)																	
FILTER (Fang et al., 2020)	88.7	66.1	88.5	89.2	88.3	96.0	89.1	86.3	78.0	70.8	86.1	88.9	64.9	76.7	82.6	72.6	89.8
XTUNE	90.7	74.2	89.9	90.2	87.4	96.1	90.5	88.4	75.9	74.2	87.9	90.2	85.9	79.3	83.2	73.3	91.0

Model	ja	kk	ko	mr	nl	pt	ru	ta	te	th	tl	tr	ur	vi	yo	zh	Avg.
Cross-lingual-transfer (models are fine-tuned on English training data without translation available)																	
XLM-R$_{\text{large}}$	15.9	78.1	53.9	80.8	89.5	87.6	89.5	65.2	86.6	47.2	92.2	76.3	70.3	56.8	24.6	25.7	73.8
XTUNE	62.7	78.3	55.7	82.4	90.2	88.5	90.5	63.6	88.3	61.8	94.5	76.9	72.0	57.8	24.4	69.4	78.5
Fine-tune multilingual model on all target language target language training sets (translate-train-all)																	
FILTER (Fang et al., 2020)	40.4	80.4	53.3	86.4	89.4	88.3	90.5	65.3	87.3	57.2	94.1	77.0	70.9	58.0	43.1	53.1	76.9
XTUNE	65.3	79.8	56.0	85.5	89.7	89.3	90.8	65.7	85.5	61.4	93.8	78.3	74.0	57.5	27.9	68.8	79.3

Table 14: POS results (accuracy) for each language.

Model	en	af	ar	bg	bn	de	el	es	et	eu	fa	fi	fr	he	hi	hu	id	it	ja	jv
Cross-lingual-transfer (models are fine-tuned on English training data without translation available)																				
XLM-R$_{\text{large}}$	84.7	78.9	53.0	81.4	78.8	78.8	79.5	79.6	79.1	60.9	61.9	79.2	80.5	56.8	73.0	79.8	53.0	81.3	23.2	62.5
XTUNE	85.0	80.4	59.1	84.8	79.1	80.5	82.0	78.1	81.5	64.5	65.9	82.2	81.9	62.0	75.0	82.8	55.8	83.1	30.5	65.9
Translate-train-all (translation-based augmentation is available for English training data)																				
FILTER (Fang et al., 2020)	83.5	80.4	60.7	83.5	78.4	80.4	80.7	74.0	81.0	66.9	71.3	80.2	79.9	57.4	74.3	82.2	54.0	81.9	24.3	63.5
XTUNE	84.4	81.7	59.7	85.3	80.8	80.9	82.0	74.1	83.4	69.9	63.6	82.5	80.6	64.0	76.3	83.8	57.9	83.3	26.5	69.8

Model	ka	kk	ko	ml	mr	ms	my	nl	pt	ru	sw	ta	te	th	tl	tr	ur	vi	yo	zh
Cross-lingual-transfer (models are fine-tuned on English training data without translation available)																				
XLM-R$_{\text{large}}$	71.6	56.2	60.0	67.8	68.1	57.1	54.3	84.0	81.9	69.1	70.5	59.5	55.8	1.3	73.2	76.1	56.4	79.4	33.6	33.1
XTUNE	76.7	57.5	65.9	68.1	73.3	67.2	63.7	85.3	84.0	73.6	70.1	66.1	60.1	1.8	76.9	83.6	76.0	80.3	44.4	38.7
Translate-train-all (translation-based augmentation is available for English training data)																				
FILTER (Fang et al., 2020)	71.0	51.1	63.8	70.2	69.8	69.3	59.0	84.6	82.1	71.1	70.6	64.3	58.7	2.4	74.4	83.0	73.4	75.8	42.9	35.4
XTUNE	76.3	56.9	67.1	72.6	71.5	72.5	66.7	85.8	82.1	75.2	72.4	66.0	61.8	1.1	77.5	83.7	75.6	80.8	44.9	36.5

Table 15: NER results (F1 scores) for each language.