Synergy with Translation Artifacts for Training and Inference in Multilingual Tasks

Jaehoon Oh*
Graduate School of DS, KAIST
jhoon.oh@kaist.ac.kr

Jongwoo Ko*, **Se-Young Yun**
Graduate School of AI, KAIST
{jongwoo.ko, yunseyoung}@kaist.ac.kr

Abstract

Translation has played a crucial role in improving the performance on multilingual tasks: (1) to generate the target language data from the source language data for training and (2) to generate the source language data from the target language data for inference. However, prior works have not considered the use of both translations simultaneously. This paper shows that combining them can synergize the results on various multilingual sentence classification tasks. We empirically find that translation artifacts stylized by translators are the main factor of the performance gain. Based on this analysis, we adopt two training methods, SupCon and MixUp, considering translation artifacts. Furthermore, we propose a cross-lingual fine-tuning algorithm called MUSC, which uses SupCon and MixUp jointly and improves the performance. Our code is available at https://github.com/jongwooko/MUSC.

1 Introduction

Large-scale pre-trained multilingual language models (Devlin et al., 2019; Conneau and Lample, 2019; Huang et al., 2019; Conneau et al., 2020; Luo et al., 2021) have shown promising transferability in zero-shot cross-lingual transfer (ZSXLT), where pre-trained language models (PLMs) are fine-tuned using a labeled task-specific dataset from a *rich-resource* source language (e.g., English or Spanish) and then evaluated on *zero-resource* target languages. Multilingual PLMs yield a universal representation space across different languages, thereby improving multilingual task performance (Pires et al., 2019; Chen et al., 2019). Recent work has enhanced cross-lingual transferability by reducing the discrepancies between languages based on translation approaches during fine-tuning (Fang et al., 2021; Zheng et al., 2021; Yang et al., 2022). Our paper focuses on when translated datasets are available for cross-lingual transfer (XLT).

Conneau et al. (2018) provided two translation-based XLT baselines: `translate-train` and `translate-test`. The former fine-tunes a multilingual PLM (e.g., multilingual BERT) using the *original* source language and *machine-translated* target languages simultaneously and then evaluates it on the target languages. Meanwhile, the latter fine-tunes a source language-based PLM (e.g., English BERT) using the *original* source language and then evaluates it on the *machine-translated* source language. Both baselines improve the performance compared to ZSXLT; however, they are sensitive to the translator, including translation artifacts, which are characteristics stylized by the translator (Conneau et al., 2018; Artetxe et al., 2020).

Artetxe et al. (2020) showed that matching the types of text (i.e., origin or translationese[1]) between training and inference is essential due to the presence of translation artifacts under `translate-test`. Recently, Yu et al. (2022) proposed a training method that projects the original and translated texts into the same representation space under `translate-train`. However, prior works have not considered the two baselines simultaneously.

In this paper, we combine `translate-train` and `translate-test` using a pre-trained multilingual BERT, to improve the performance. Next, we identify that fine-tuning using the *translated target* dataset is required to improve the performance on the *translated source* dataset due to translation artifacts even if the languages for training and inference are different. Finally, to consider translation artifacts during fine-tuning, we adopt two training methods, supervised contrastive learning (SupCon; Khosla et al. 2020) and MixUp (Zhang et al., 2018) and propose MUSC, which combines them and improves the performance for multilingual sentence classification tasks.

*Equal contribution

[1]Original text is directly written by humans. Translationese includes both human-translated and machine-translated texts.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6747–6754
December 7-11, 2022 ©2022 Association for Computational Linguistics

Table 1: Notations of datasets.

Notation	Description
$\mathcal{S}_{\mathsf{trn}}$	given source dataset for training
$\mathcal{T}_{\mathsf{trn}}$	given target dataset for training
$\mathcal{T}^{\mathsf{MT}}_{\mathsf{trn}}$	machine-translated target dataset from $\mathcal{S}_{\mathsf{trn}}$ for training
$\mathcal{T}^{\mathsf{BT}}_{\mathsf{trn}}$	back-translated target dataset from $\mathcal{T}_{\mathsf{trn}}$ for training
$\mathcal{T}_{\mathsf{tst}}$	given target dataset for inference
$\mathcal{S}^{\mathsf{MT}}_{\mathsf{tst}}$	machine-translated source dataset from $\mathcal{T}_{\mathsf{tst}}$ for inference

Table 2: Algorithm comparison.

Algorithm	PLM	Training	Inference
ZSXLT	Multilingual	$\mathcal{S}_{\mathsf{trn}}$	$\mathcal{T}_{\mathsf{tst}}$
`translate-train`	Multilingual	$\mathcal{S}_{\mathsf{trn}}$ & $\mathcal{T}^{\mathsf{MT}}_{\mathsf{trn}}$	$\mathcal{T}_{\mathsf{tst}}$
`translate-test`	English	$\mathcal{S}_{\mathsf{trn}}$	$\mathcal{S}^{\mathsf{MT}}_{\mathsf{tst}}$
`translate-all`	Multilingual	$\mathcal{S}_{\mathsf{trn}}$ & $\mathcal{T}^{\mathsf{MT}}_{\mathsf{trn}}$	$\mathcal{T}_{\mathsf{tst}}$ & $\mathcal{S}^{\mathsf{MT}}_{\mathsf{tst}}$

2 Scope of the Study

In this study, four datasets are used: MARC and MLDoc for single sentence classification, and PAWSX and XNLI from XTREME (Hu et al., 2020) for sentence pair classification. The details of datasets are provided in Appendix A. Each dataset consists of the source dataset for training $\mathcal{S}_{\mathsf{trn}}$ and the target dataset for inference $\mathcal{T}_{\mathsf{tst}}$, where $\mathcal{S}_{\mathsf{trn}}$ is original and $\mathcal{T}_{\mathsf{tst}}$ is original (for MARC and MLDoc) or human-translated (for PAWSX and XNLI). For MARC and MLDoc, the original target dataset for training $\mathcal{T}_{\mathsf{trn}}$ is additionally given.

We use the given translated datasets $\mathcal{T}^{\mathsf{MT}}_{\mathsf{trn}}$ for PAWSX and XNLI. However, for MARC and MLDoc, the translated datasets are not given. Therefore, we use an m2m_100_418M translator (Fan et al., 2021) from the open-source library `EasyNMT`[2] to create the translated datasets. $\mathcal{T}^{\mathsf{MT}}_{\mathsf{trn}}$ is translated from $\mathcal{S}_{\mathsf{trn}}$ (i.e., $\mathcal{S}_{\mathsf{trn}} \rightarrow \mathcal{T}^{\mathsf{MT}}_{\mathsf{trn}}$), and $\mathcal{T}^{\mathsf{BT}}_{\mathsf{trn}}$ is back-translated from $\mathcal{T}_{\mathsf{trn}}$ (i.e., $\mathcal{T}_{\mathsf{trn}} \rightarrow \mathcal{S}^{\mathsf{MT}}_{\mathsf{trn}} \rightarrow \mathcal{T}^{\mathsf{BT}}_{\mathsf{trn}}$; Sennrich et al. 2016). Similarly, for inference, $\mathcal{S}^{\mathsf{MT}}_{\mathsf{tst}}$ is translated from $\mathcal{T}_{\mathsf{tst}}$. The notations used in this paper are listed in Table 1.

We use the pre-trained cased multilingual BERT (Devlin et al., 2019) from HuggingFace Transformers (Wolf et al., 2020) and use accuracy as a metric. Detailed information for fine-tuning is provided in Appendix B.

[2]https://github.com/UKPLab/EasyNMT

Table 3: Results according to the inference datasets (Acc. in %). $\mathcal{S}_{\mathsf{trn}}$ and $\mathcal{T}^{\mathsf{MT}}_{\mathsf{trn}}$ are used for training. The number in the parenthesis of MLDoc is the number of training samples. 'Ens.' indicates the ensemble of results on the two different test datasets in the inference. XNLI results are reported in Appendix C.

Dataset	Inference	EN	ZH	FR	DE	RU	ES	IT	KO	JA	Avg.
MARC	$\mathcal{T}_{\mathsf{tst}}$	65.2	47.8	55.4	59.1	-	55.8	-	-	47.8	55.1
	$\mathcal{S}^{\mathsf{MT}}_{\mathsf{tst}}$	65.2	44.9	54.4	59.8	-	55.4	-	-	44.9	54.5
	Ens.	65.2	49.3	56.1	61.2	-	56.2	-	-	48.8	**56.1**
MLDoc (1000)	$\mathcal{T}_{\mathsf{tst}}$	91.1	77.4	74.5	84.0	67.9	74.4	65.0	-	74.4	76.1
	$\mathcal{S}^{\mathsf{MT}}_{\mathsf{tst}}$	91.1	77.6	79.0	88.1	61.3	76.4	72.3	-	67.3	76.6
	Ens.	91.1	78.9	78.3	87.9	66.1	76.2	71.2	-	74.9	**78.1**
MLDoc (10000)	$\mathcal{T}_{\mathsf{tst}}$	97.4	82.6	91.1	91.0	72.2	85.9	78.0	-	72.6	83.8
	$\mathcal{S}^{\mathsf{MT}}_{\mathsf{tst}}$	97.4	86.4	92.0	92.6	72.4	88.2	79.0	-	71.0	84.9
	Ens.	97.4	87.7	92.2	92.6	72.1	88.0	80.6	-	75.9	**85.8**
PAWSX	$\mathcal{T}_{\mathsf{tst}}$	94.5	85.0	91.2	89.0	-	90.5	-	83.1	83.3	88.1
	$\mathcal{S}^{\mathsf{MT}}_{\mathsf{tst}}$	94.5	84.5	91.7	90.6	-	91.3	-	83.1	80.9	88.1
	Ens.	94.5	86.1	92.0	91.2	-	91.6	-	85.3	82.8	**89.1**

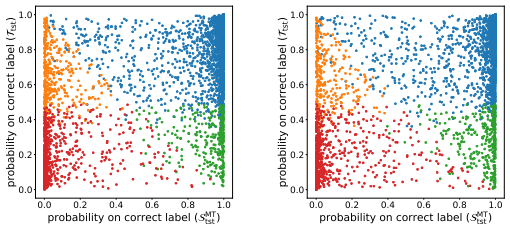
3 Original and Translationese Ensemble

In this section, we demonstrate that the two baselines, `translate-train` and `translate-test`, are easily combined to improve performance, which we call it `translate-all`. Table 2 describes the differences between algorithms.

Table 3 presents the results according to the inference dataset when the models are fine-tuned using $\mathcal{S}_{\mathsf{trn}}$ and $\mathcal{T}^{\mathsf{MT}}_{\mathsf{trn}}$. Inference on $\mathcal{T}_{\mathsf{tst}}$ is a general way to evaluate the models, i.e., `translate-train`. In addition, we evaluate the models on $\mathcal{S}^{\mathsf{MT}}_{\mathsf{tst}}$ like `translate-test`. Furthermore, we ensemble the two results from different test datasets by averaging the predicted predictions, i.e., `translate-all`, because averaging the predictions over models or data points is widely used to improve predictive performance and uncertainty estimation of models (Gontijo-Lopes et al., 2022; Kim et al., 2020a).

From Table 3, it is shown that even if the multilingual PLMs are fine-tuned with $\mathcal{S}_{\mathsf{trn}}$ and $\mathcal{T}^{\mathsf{MT}}_{\mathsf{trn}}$, the performance on the translated source data $\mathcal{S}^{\mathsf{MT}}_{\mathsf{tst}}$ is competitive with that on the target data $\mathcal{T}_{\mathsf{tst}}$. Furthermore, ensemble inference increases the performance on all datasets. This can be interpreted as the effectiveness of the test time augmentation (Kim et al., 2020a; Ashukha et al., 2021), because the results on the two *test* datasets, $\mathcal{T}_{\mathsf{tst}}$ and $\mathcal{S}^{\mathsf{MT}}_{\mathsf{tst}}$ (augmented from $\mathcal{T}_{\mathsf{tst}}$), are combined.

To explain the changes in inferences via test time augmentation, we describe the predicted probability values on the correct label when the models are evaluated on $\mathcal{T}_{\mathsf{tst}}$ and $\mathcal{S}^{\mathsf{MT}}_{\mathsf{tst}}$, as depicted in Figure 1. The green and orange dots represent the benefits



(a) ZH on XNLI (b) DE on XNLI

Figure 1: Predicted probability values on correct label when the models are evaluated on $\mathcal{T}_{\text{tst}}$ and $\mathcal{S}^{\text{MT}}_{\text{tst}}$. The colors indicate right or wrong predictions: right on $\mathcal{T}_{\text{tst}}$ and right on Ens. (blue), right on $\mathcal{T}_{\text{tst}}$ and wrong on Ens. (orange), wrong on $\mathcal{T}_{\text{tst}}$ and right on Ens. (green), and wrong on $\mathcal{T}_{\text{tst}}$ and wrong on Ens. (red).

Table 4: Results according to the matching between types of text for training and inference (Acc. in %). $\mathcal{S}_{\text{trn}}$ is also used for training.

Dataset	Training	Inference	EN	ZH	FR	DE	RU	ES	IT	JA	Avg.
MARC	$\mathcal{T}_{\text{trn}}$	$\mathcal{T}_{\text{tst}}$	65.3	57.9	61.4	65.5	-	62.0	-	60.1	**62.0**
	$\mathcal{T}^{\text{BT}}_{\text{trn}}$		65.7	55.7	60.1	63.9	-	60.3	-	56.7	60.4
	$\mathcal{T}_{\text{trn}}$	$\mathcal{S}^{\text{MT}}_{\text{tst}}$	65.3	48.2	57.1	61.8	-	57.7	-	47.1	56.2
	$\mathcal{T}^{\text{BT}}_{\text{trn}}$		65.7	49.2	57.7	62.4	-	57.2	-	48.5	**56.8**
MLDoc (1000)	$\mathcal{T}_{\text{trn}}$	$\mathcal{T}_{\text{tst}}$	93.7	91.9	93.6	95.6	87.2	95.5	86.8	89.3	**91.7**
	$\mathcal{T}^{\text{BT}}_{\text{trn}}$		93.4	90.6	93.5	95.1	87.1	92.7	86.4	86.4	90.6
	$\mathcal{T}_{\text{trn}}$	$\mathcal{S}^{\text{MT}}_{\text{tst}}$	93.7	86.4	92.5	93.9	83.8	93.1	80.6	73.3	87.2
	$\mathcal{T}^{\text{BT}}_{\text{trn}}$		93.4	87.2	93.0	94.8	84.0	93.2	80.5	75.9	**87.7**
MLDoc (10000)	$\mathcal{T}_{\text{trn}}$	$\mathcal{T}_{\text{tst}}$	96.8	93.9	96.7	97.5	89.5	96.8	92.2	92.5	**94.5**
	$\mathcal{T}^{\text{BT}}_{\text{trn}}$		97.0	93.3	96.1	97.2	87.9	95.7	90.8	89.5	93.4
	$\mathcal{T}_{\text{trn}}$	$\mathcal{S}^{\text{MT}}_{\text{tst}}$	96.8	88.9	94.9	96.4	84.3	94.0	83.5	75.7	89.3
	$\mathcal{T}^{\text{BT}}_{\text{trn}}$		97.0	87.6	94.6	95.4	84.2	93.8	85.7	77.2	**89.4**

and losses via the ensemble, respectively. The improved performance through the ensemble means that the number of green samples is greater than the number of orange samples in Figure 1.

To analyze where the performance gain comes from, we focus on the green samples. The green samples are concentrated around the right down corner, which implies that wrong predictions on $\mathcal{T}_{\text{tst}}$ can be right predictions with high confidence on $\mathcal{S}^{\text{MT}}_{\text{tst}}$. In fact, this phenomenon is the opposite of what we expected; the samples are expected to be concentrated around the $y = x$ line, because the semantic meaning between $\mathcal{T}_{\text{tst}}$ and $\mathcal{S}^{\text{MT}}_{\text{tst}}$ is similar even though the languages are different. This implies that semantic meaning is not the main factor explaining the performance gain of the ensemble.

4 Translation Artifacts for Training

To find the main factor of performance gain, we hypothesize that matching the types of text (i.e., original or translated) between training and inference is important *even if the languages used for training and inference are different*, by expanding on Artetxe et al. (2020). For the analysis, we use MARC and MLDoc because they provide $\mathcal{T}_{\text{trn}}$, which has no artifacts.

Table 4 describes the results according to the matching between texts for training and inference. Well-matched texts are better than badly matched ones. In particular, the results that $\mathcal{T}^{\text{BT}}_{\text{trn}}$–$\mathcal{S}^{\text{MT}}_{\text{tst}}$ is better than $\mathcal{T}_{\text{trn}}$–$\mathcal{S}^{\text{MT}}_{\text{tst}}$ support our hypothesis. This implies that biasing training and inference datasets using the same translator can lead to performance improvement, and that translation artifacts can change wrong predictions on $\mathcal{T}_{\text{tst}}$ into right predictions on $\mathcal{S}^{\text{MT}}_{\text{tst}}$ when the models are trained using $\mathcal{T}^{\text{MT}}_{\text{trn}}$, as shown in Section 3.

4.1 Proposed Method: MUSC

We propose an XLT method called MUSC, by applying SupCon (Khosla et al., 2020) and MixUp (Zhang et al., 2018) jointly. Namely, our method is contrastive learning with mixture sentences in *supervised* settings. Several works have attempted to employ the idea of mixtures on unsupervised contrastive learning (Kim et al., 2020b; Shen et al., 2022); however, ours is the first to leverage the label information in a mixture. In this section, the loss functions are formulated at batch level with a batch size of N, and $\uparrow$ and $\downarrow$ indicate the normal and reverse order, respectively, in a batch. All methods are designed upon the `translate-all`.

SupCon. We adopt SupCon, which makes the samples in the same class closer (Gunel et al., 2021), to reduce the discrepancies between original and translated texts. Namely, SupCon helps models to learn both originality of $\mathcal{S}_{\text{trn}}$ and artifacts of $\mathcal{T}^{\text{MT}}_{\text{trn}}$ comprehensively. The loss function of SupCon ($\mathcal{L}_{sc}$) with $I \equiv [1, \ldots, 2N]$ is as follows:

$$\mathcal{L}_{sc}(\mathbf{Z}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{T}}, \mathbf{y}) = \sum_{i \in I} -\log \left\{ \frac{1}{|P(i)|} \sum_{p \in P(i)} \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_p / \tau)}{\sum_{\mathbf{z}_a \in \mathbf{Z} \setminus \{\mathbf{z}_i\}} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)} \right\},$$

where $\mathbf{Z} = [\mathbf{Z}_{\mathcal{S}}; \mathbf{Z}_{\mathcal{T}}] \in \mathbb{R}^{2N \times d_p}$ is the projections of [CLS] token representations through an encoder f and a projector g, i.e., $g(f(\mathbf{E})^{[\text{CLS}]})$, and $\mathbf{z}_i$ indicates the i-th row of $\mathbf{Z}$. $\mathbf{Z}$ is concatenated along with the batch dimension and d_p is the dimension of projections. The positive set of the sample i, $P(i)$, is defined as $\{j | y'_j = y'_i, j \in I \setminus \{i\}\}$, where $[y'_1, \ldots, y'_N] = [y'_{N+1}, \ldots, y'_{2N}] = \mathbf{y}$.

MixUp. We adopt MixUp to densify original and translated texts, respectively. MixUp is performed on the word embeddings by following Chen et al.

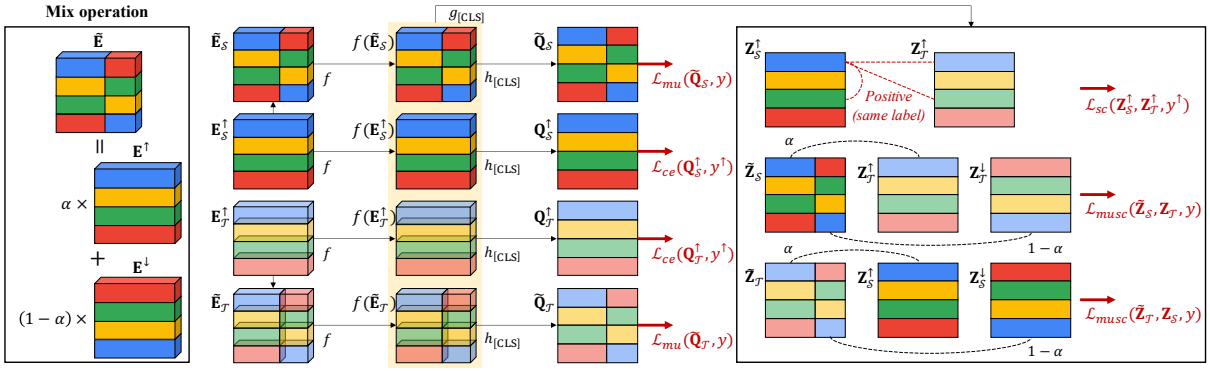


Figure 2: Overview of MUSC. $\mathbf{E}_{\mathcal{S}}$ and $\mathbf{E}_{\mathcal{T}}$ are the embeddings of the paired source and target languages, and each row indicates one sentence. Note that in the mix operation, addition and multiplication are operated elementwisely. f, g, and h are the encoder, projector, and classifier, respectively. $g_{[\mathrm{CLS}]}(f(\mathbf{E}))$ and $h_{[\mathrm{CLS}]}(f(\mathbf{E}))$ means $g(f(\mathbf{E})^{[\mathrm{CLS}]})$ and $h(f(\mathbf{E})^{[\mathrm{CLS}]})$, respectively. $g(f(\mathbf{E})^{[\mathrm{CLS}]})$ is expressed as $\mathbf{Z}$. In this figure, it is assumed that the batch size is four and that the blue- and green-colored samples have the same class.

(2020), because it is infeasible to directly apply MixUp to discrete word tokens. MixUp with $\alpha \in [0, 1]$ is as follows:

$$\tilde{\mathbf{E}} = \mathtt{Mix}_{\alpha}(\mathbf{E}^{\uparrow}, \mathbf{E}^{\downarrow}) = \alpha \mathbf{E}^{\uparrow} + (1-\alpha)\mathbf{E}^{\downarrow},$$

where $\mathbf{E}^{\uparrow} = \mathbf{X}\mathbf{W}_e \in \mathbb{R}^{N \times L \times d}$ is the output of the embedding layer for a given batch $\mathbf{X} \in \mathbb{R}^{N \times L \times |V|}$ with weight matrix $\mathbf{W}_e \in \mathbb{R}^{|V| \times d}$. L, $|V|$, and d indicate maximum sequence length, vocab size, and dimension of word embeddings, respectively. $\mathbf{E}^{\downarrow}$ is reversed along with the batch dimension. We apply MixUp between the same language to densify each type of text. For convenience in implementation, we mix a normal batch ($\uparrow$) and a reversed batch ($\downarrow$), following Shen et al. (2022). The mixing process is conducted elementwisely. The loss function of MixUp ($\mathcal{L}_{mu}$) with cross-entropy ($\mathcal{L}_{ce}$) is as follows:

$$\mathcal{L}_{mu}(\tilde{\mathbf{Q}}, \mathbf{y}) = \alpha \mathcal{L}_{ce}(\tilde{\mathbf{Q}}, \mathbf{y}^{\uparrow}) + (1-\alpha)\mathcal{L}_{ce}(\tilde{\mathbf{Q}}, \mathbf{y}^{\downarrow}),$$

where $\tilde{\mathbf{Q}} = h(f(\tilde{\mathbf{E}})^{[\mathrm{CLS}]})$ is the logits of [CLS] token for the mixed embeddings, with an encoder f and a classifier h. $\mathbf{y}$ is a set of labels in the same batch.

MUSC. We replace the original projected representations in $\mathcal{L}_{sc}$ with mixture ones, i.e., $\mathbf{Z}_{\mathcal{S}} \rightarrow \tilde{\mathbf{Z}}_{\mathcal{S}}$ or $\mathbf{Z}_{\mathcal{T}} \rightarrow \tilde{\mathbf{Z}}_{\mathcal{T}}$, to use MixUp and SupCon jointly. The loss functions of MUSC ($\mathcal{L}_{musc}$) are as follows:

$$\mathcal{L}_{musc}(\tilde{\mathbf{Z}}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{T}}, \mathbf{y}) = \alpha \mathcal{L}_{sc}(\tilde{\mathbf{Z}}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{T}}^{\uparrow}, \mathbf{y}^{\uparrow}) + (1-\alpha)\mathcal{L}_{sc}(\tilde{\mathbf{Z}}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{T}}^{\downarrow}, \mathbf{y}^{\downarrow}),$$

$$\mathcal{L}_{musc}(\tilde{\mathbf{Z}}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{S}}, \mathbf{y}) = \alpha \mathcal{L}_{sc}(\tilde{\mathbf{Z}}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{S}}^{\uparrow}, \mathbf{y}^{\uparrow}) + (1-\alpha)\mathcal{L}_{sc}(\tilde{\mathbf{Z}}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{S}}^{\downarrow}, \mathbf{y}^{\downarrow}).$$

We calculate $\mathcal{L}_{musc}$ by decomposing it in two opposite orders, similar to $\mathcal{L}_{mu}$. Finally, the total loss function ($\mathcal{L}$), descried in Figure 2, is as follows:

Table 5: Results according to the losses. $\mathcal{S}_{\mathsf{trn}}$ and $\mathcal{T}_{\mathsf{trn}}^{\mathsf{MT}}$ are used for training and $\mathcal{T}_{\mathsf{tst}}$ and $\mathcal{S}_{\mathsf{tst}}^{\mathsf{MT}}$ are used for ensemble inference, i.e., under `translate-all`. $-$ denotes baseline which only applies $\mathcal{L}_{ce}$. $\mathcal{L}_{ce}$ is basically added for all methods. XNLI results are reported in Appendix C.

Dataset	Method	EN	ZH	FR	DE	RU	ES	IT	KO	JA	Avg.
MARC	$-$	65.2	49.3	56.1	61.2	-	56.2	-	-	48.8	56.1
	$\mathcal{L}_{sc}$	64.9	49.1	56.1	61.4	-	55.7	-	-	49.3	56.1
	$\mathcal{L}_{mu}$	64.5	49.4	55.5	61.5	-	55.9	-	-	48.7	55.9
	$\mathcal{L}_{sc} + \mathcal{L}_{mu}$	65.1	49.5	56.1	61.5	-	56.1	-	-	49.9	**56.4**
	$\mathcal{L}$	65.5	49.4	56.4	61.6	-	56.0	-	-	48.5	56.2
MLDoc (1000)	$-$	91.1	78.9	78.3	87.9	66.1	76.2	71.2	-	74.9	78.1
	$\mathcal{L}_{sc}$	95.0	86.0	85.0	91.4	67.3	84.2	75.0	-	72.3	82.0
	$\mathcal{L}_{mu}$	94.0	83.7	84.2	90.5	73.4	82.4	75.5	-	71.2	81.9
	$\mathcal{L}_{sc} + \mathcal{L}_{mu}$	91.7	85.0	88.6	90.4	71.0	82.9	76.7	-	75.3	82.7
	$\mathcal{L}$	94.8	86.7	86.2	90.2	73.3	80.8	74.8	-	77.5	**83.1**
MLDoc (2000)	$-$	95.7	87.3	85.9	91.3	80.5	81.4	76.7	-	78.2	84.6
	$\mathcal{L}_{sc}$	95.8	89.0	90.3	92.2	80.9	83.4	79.4	-	77.7	86.1
	$\mathcal{L}_{mu}$	95.9	88.8	92.3	92.3	81.1	85.9	78.5	-	75.7	86.3
	$\mathcal{L}_{sc} + \mathcal{L}_{mu}$	95.3	88.4	92.0	93.1	80.4	85.8	79.1	-	77.2	86.4
	$\mathcal{L}$	94.8	88.7	89.8	92.7	82.3	86.8	80.2	-	78.4	**86.7**
MLDoc (5000)	$-$	96.8	89.7	92.2	92.7	73.6	82.6	78.8	-	77.2	85.4
	$\mathcal{L}_{sc}$	96.7	89.0	93.0	93.7	71.5	88.2	81.1	-	77.6	86.3
	$\mathcal{L}_{mu}$	96.9	88.9	91.3	93.7	72.0	86.5	81.0	-	76.3	85.8
	$\mathcal{L}_{sc} + \mathcal{L}_{mu}$	96.6	88.1	92.2	92.2	78.7	85.4	80.2	-	76.3	86.2
	$\mathcal{L}$	97.0	88.6	92.9	95.0	77.7	89.1	81.1	-	72.9	**86.8**
MLDoc (10000)	$-$	97.4	87.7	92.2	92.6	72.1	88.0	80.6	-	75.9	85.8
	$\mathcal{L}_{sc}$	97.3	90.6	92.0	93.0	71.1	88.5	80.9	-	78.8	86.5
	$\mathcal{L}_{mu}$	97.4	88.7	92.5	94.8	71.6	89.9	79.4	-	72.7	85.9
	$\mathcal{L}_{sc} + \mathcal{L}_{mu}$	97.4	89.4	93.6	93.8	72.1	91.2	79.7	-	76.3	86.7
	$\mathcal{L}$	97.4	89.8	94.1	94.6	71.7	88.9	80.3	-	78.2	**86.9**
PAWSX	$-$	94.5	86.1	92.0	91.2	-	91.6	-	85.3	82.8	89.1
	$\mathcal{L}_{sc}$	94.5	87.3	92.6	91.3	-	92.2	-	85.6	84.0	89.6
	$\mathcal{L}_{mu}$	94.5	86.3	92.3	92.2	-	92.4	-	85.3	84.8	89.7
	$\mathcal{L}_{sc} + \mathcal{L}_{mu}$	95.1	87.0	92.2	92.2	-	91.8	-	85.5	84.8	**89.8**
	$\mathcal{L}$	94.9	87.0	92.4	92.0	-	91.9	-	86.0	84.7	**89.8**

$$\mathcal{L} = (1-\lambda)\left[\sum_{i \in \{\mathcal{S},\mathcal{T}\}} \mathcal{L}_{ce}(\mathbf{Q}_i, \mathbf{y}) + \sum_{i \in \{\mathcal{S},\mathcal{T}\}} \mathcal{L}_{mu}(\tilde{\mathbf{Q}}_i, \mathbf{y})\right]$$
$$+ \lambda\left[\mathcal{L}_{sc}(\mathbf{Z}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{T}}, \mathbf{y}) + \mathcal{L}_{musc}(\tilde{\mathbf{Z}}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{T}}, \mathbf{y}) + \mathcal{L}_{musc}(\tilde{\mathbf{Z}}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{S}}, \mathbf{y})\right].$$

Table 5 describes the ablation study according to the applied loss functions. – denotes baseline which only applies $\mathcal{L}_{ce}$. Other methods include $\mathcal{L}_{ce}$ and additionally apply the corresponding loss, respectively. It is shown that SupCon ($\mathcal{L}_{sc}$) and MixUp ($\mathcal{L}_{mu}$) improve performance on most datasets even when they are used separately. The effectiveness of these losses is powerful when dataset size is small. Moreover, our total loss ($\mathcal{L}$), which includes learning a model using SupCon and MixUp jointly ($\mathcal{L}_{musc}$), outperforms both SupCon and MixUp on all datasets. In addition, our total loss ($\mathcal{L}$) brings more performance gains than the simple conjunction of SupCon and MixUp ($\mathcal{L}_{sc} + \mathcal{L}_{mu}$) for all datasets except for MARC dataset. These results demonstrate that our proposed MUSC effectively collaborates the SupCon and MixUp. The optimized hyperparameters are reported in Appendix B.

5 Conclusion

In this paper, we showed that `translate-train` and `translate-test` are easily synergized from the test time augmentation perspective and found that the improved performance is based on translation artifacts. Based on our analysis, we propose MUSC, which is supervised contrastive learning with mixture sentences, to enhance the generalizability on translation artifacts. Our work highlighted the role of translation artifacts for XLT.

Limitations

Our work addressed the role of translation artifacts for cross-lingual transfer. Limitation of our work is that we experimented for sentence classification tasks using multilingual BERT, because it is almost impossible to get token-level ground truths using translator.

Ethics Statement

Our work does not violate the ethical issues. Furthermore, we showed that a new baseline, `translate-all`, could achieve higher performance, and proposed MUSC designed upon the `translate-all` approach. We believe that various algorithms can be developed based on the `translate-all` for multilingual tasks.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST), 10%) and Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2022-0-00641, XVoice: Multi-Modal Voice Meta Learning, 90%)

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2020. Translation artifacts in cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7674–7684, Online. Association for Computational Linguistics.

Arsenii Ashukha, Andrei Atanov, and Dmitry Vetrov. 2021. Mean embeddings with test-time data augmentation for ensembling of representations. *arXiv preprint arXiv:2106.08038*.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. 2019. Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3098–3112, Florence, Italy. Association for Computational Linguistics.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel 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*, pages 2475–2485, Brussels, Belgium. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *The Journal of Machine Learning Research*, 22(107):1–48.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2021. Filter: An enhanced fusion method for cross-lingual language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12776–12784.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. Association for Computational Linguistics.

Raphael Gontijo-Lopes, Yann Dauphin, and Ekin Dogus Cubuk. 2022. No one representation to rule them all: Overlapping features of training methods. In *International Conference on Learning Representations*.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2020. Statistical power and translationese in machine translation evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 72–81, Online. Association for Computational Linguistics.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*.

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*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. 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)*, pages 2485–2494, Hong Kong, China. Association for Computational Linguistics.

Masahiro Kaneko and Danushka Bollegala. 2021. Debiasing pre-trained contextualised embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1256–1266, Online. Association for Computational Linguistics.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The multilingual Amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4563–4568, Online. Association for Computational Linguistics.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673.

Ildoo Kim, Younghoon Kim, and Sungwoong Kim. 2020a. Learning loss for test-time augmentation. *Advances in Neural Information Processing Systems*, 33:4163–4174.

Sungnyun Kim, Gihun Lee, Sangmin Bae, and Se-Young Yun. 2020b. Mixco: Mix-up contrastive learning for visual representation. *arXiv preprint arXiv:2010.06300*.

Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. 2020. i-mix: A domain-agnostic strategy for contrastive representation learning. *arXiv preprint arXiv:2010.08887*.

Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2020. Mixkd: Towards efficient distillation of large-scale language models. *arXiv preprint arXiv:2011.00593*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Fuli Luo, Wei Wang, Jiahao Liu, Yijia Liu, Bin Bi, Songfang Huang, Fei Huang, and Luo Si. 2021. VECO: Variable and flexible cross-lingual pre-training for language understanding and generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3980–3994, Online. Association for Computational Linguistics.

Kazuhisa Matsunaga, Akira Hamada, Akane Minagawa, and Hiroshi Koga. 2017. Image classification of melanoma, nevus and seborrheic keratosis by deep neural network ensemble. *arXiv preprint arXiv:1703.03108*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for*

Computational Linguistics, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. 2018. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8571–8580.

Peter Prettenhofer and Benno Stein. 2010. Cross-Language Text Classification using Structural Correspondence Learning. In *48th Annual Meeting of the Association of Computational Linguistics (ACL 2010)*, pages 1118–1127. Association for Computational Linguistics.

Holger Schwenk and Xian Li. 2018. A corpus for multilingual document classification in eight languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Zhiqiang Shen, Zechun Liu, Zhuang Liu, Marios Savvides, Trevor Darrell, and Eric Xing. 2022. Un-mix: Rethinking image mixtures for unsupervised visual representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Lewis Smith and Yarin Gal. 2018. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Huiyun Yang, Huadong Chen, Hao Zhou, and Lei Li. 2022. Enhancing cross-lingual transfer by manifold mixup. In *International Conference on Learning Representations*.

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)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Sicheng Yu, Qianru Sun, Hao Zhang, and Jing Jiang. 2022. Translate-train embracing translationese artifacts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 362–370, Dublin, Ireland. Association for Computational Linguistics.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Michael Zhang, Nimit Sharad Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Ré. 2021. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*.

Bo Zheng, Li Dong, Shaohan Huang, Wenhui Wang, Zewen Chi, Saksham Singhal, Wanxiang Che, Ting Liu, Xia Song, and Furu Wei. 2021. Consistency regularization for cross-lingual fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3403–3417, Online. Association for Computational Linguistics.

A Dataset Description

MARC (Keung et al., 2020) is Amazon review classification dataset. **MLDoc** (Schwenk and Li, 2018) is news article classification dataset. **PAWSX** (Yang et al., 2019) is paraphrase identification dataset[3]. **XNLI** (Conneau et al., 2018) is natural language inference dataset[4].

Table 6: Dataset description

Dataset	# of languages	# of classes	# of train	# of val	# of test
MARC	6	5	200,000	5,000	5,000
MLDoc	8	4	1,000-10,000	1,000	4,000
PAWSX	7	2	49,401	2,000	2,000
XNLI	15	3	392,702	2,490	5,010

B Implementation Detail

Learning rate and λ are searched by grid from [1e-5, 3e-5, 5e-5] and from [0.1, 0.5, 0.9], respectively. Fine-tuning epochs are 4, 10, 4, and 2 on MARC, MLDoc, PAWSX, and XNLI, respectively. The batch size is 32 for all datasets. The evaluation is executed every 300 batches on all languages. Table 7 describes the optimized hyperparameters.

Table 7: Optimized hyperparameters

		MARC	MLDoc	PAWSX	XNLI
$\mathcal{L}_{sc}$	lr	3e-5	3e-5	1e-5	3e-5
	λ	0.5	0.5/0.9	0.9	0.1
$\mathcal{L}_{mu}$	lr	3e-5	3e-5	1e-5	1e-5
$\mathcal{L}$	lr	3e-5	3e-5	1e-5	1e-5
	λ	0.9	0.5/0.9	0.9	0.9

C XNLI results

Table 8: XNLI results according to the inference datasets.

Inference	EN	AR	BG	ZH	FR	DE	EL	HI	RU	ES	SW	TH	TR	UR	VI	Avg.
$\mathcal{T}_{\text{tst}}$	82.2	73.1	77.8	77.6	78.2	77.3	75.0	71.0	76.0	79.3	67.6	67.8	73.3	67.2	76.8	74.7
$\mathcal{S}^{\text{MT}}_{\text{tst}}$	82.2	74.9	78.4	75.2	77.9	78.4	77.5	71.5	75.3	78.6	69.5	71.3	75.7	67.3	75.0	75.3
Ens.	82.2	76.7	79.6	77.3	79.1	79.8	78.8	73.1	77.3	79.9	71.1	73.0	77.6	68.8	77.1	**76.8**

Table 9: XNLI results according to the training methods.

Method	EN	AR	BG	ZH	FR	DE	EL	HI	RU	ES	SW	TH	TR	UR	VI	Avg.
–	82.2	76.7	79.6	77.3	79.1	79.8	78.8	73.1	77.3	79.9	71.1	73.0	77.6	68.8	77.1	76.8
$\mathcal{L}_{sc}$	82.7	77.2	79.8	77.7	80.0	80.4	79.7	73.7	76.6	80.7	72.7	73.5	77.7	69.8	77.0	77.3
$\mathcal{L}_{mu}$	82.9	77.3	80.0	78.2	80.4	80.0	79.3	73.4	77.8	80.6	71.7	73.9	77.8	69.4	78.0	77.4
$\mathcal{L}_{sc} + \mathcal{L}_{mu}$	83.5	77.6	79.8	78.4	80.6	80.5	80.0	73.5	78.1	80.9	72.5	73.7	77.8	69.9	77.6	77.6
$\mathcal{L}$	83.6	77.9	80.2	78.1	80.8	80.3	79.7	73.6	78.0	81.0	72.2	73.6	77.8	70.4	77.8	**77.7**

[3]https://console.cloud.google.com/storage/browser/xtreme_translations/PAWSX

[4]https://console.cloud.google.com/storage/browser/xtreme_translations/XNLI

D Additional Related Works

Cross-lingual Transfer. As the recent advances in NLP demonstrate the effectiveness of pre-trained language models (PLMs) like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), the performances of XLT rapidly improve by extending the monolingual PLMs to the multilingual settings (Conneau and Lample, 2019; Conneau et al., 2020). While these multilingual PLMs show state-of-the-art performances in ZSXLT, one promising approach for improving the cross-lingual transferability is instance-based transfer by translation such as `translate-train` and `translate-test` (Conneau et al., 2018). Due to the effectiveness and acceptability of translation, most recent works (Fang et al., 2021; Zheng et al., 2021; Yang et al., 2022) focus on better utilization of translation.

Test-time augmentation. Data augmentation, which expands a dataset by adding transformed copies of each example, is a common practice in supervised learning. While the data augmentation is also widely used in XLT (Zheng et al., 2021) during training models, it can also be used at the test time to obtain greater robustness (Prakash et al., 2018), improved accuracy (Matsunaga et al., 2017), and estimates of uncertainty (Smith and Gal, 2018). Test time augmentation (TTA) combines predictions from a multi-viewed version of a single input to get a "smoothed" prediction. We also point out that using translation with XLT can be viewed as TTA, which can get performance gain from a different view of original and translation sentences. In this direction of the necessity of study for TTA (Kim et al., 2020a), we propose better utilization of translation artifacts in XLT.

Translation artifacts. "Translationese" can be referred to as characteristics in a translated text that differentiate it from the original text in the same language. While the effect of translationese has been widely studied in translation tasks (Graham et al., 2020; Freitag et al., 2020), the efficacy of translationese in XLT is under-explored. Artetxe et al. (2020) and Kaneko and Bollegala (2021) investigate the effect of translationese in `translate-test` and ZSXLT settings, however, these are apart from general training approach of XLT. Recently, Yu et al. (2022) firstly attempt to study `translate-train`, which focuses on single QA task.