Learning from Multiple Noisy Augmented Data Sets for Better Cross-Lingual Spoken Language Understanding

Yingmei Guo[1]* Linjun Shou[2] Jian Pei[3] Ming Gong[2] Mingxing Xu[1]
Zhiyong Wu[1] Daxin Jiang[2]†

[1]Department of Computer Science and Technology, Tsinghua University
[2]NLP Group, Microsoft STCA
[3]School of Computing Science, Simon Fraser University
{guoym18@mails.,xumx@,zywu@sz.}tsinghua.edu.cn
{lisho,migon,djiang}@microsoft.com jpei@cs.sfu.ca

Abstract

Lack of training data presents a grand challenge to scaling out spoken language understanding (SLU) to low-resource languages. Although various data augmentation approaches have been proposed to synthesize training data in low-resource target languages, the augmented data sets are often noisy, and thus impede the performance of SLU models. In this paper we focus on mitigating noise in augmented data. We develop a denoising training approach. Multiple models are trained with data produced by various augmented methods. Those models provide supervision signals to each other. The experimental results show that our method outperforms the existing state of the art by 3.05 and 4.24 percentage points on two benchmark datasets, respectively. The code will be made open sourced on github.

1 Introduction

Spoken language understanding (SLU) is a key component in task-oriented dialogue systems. SLU consists of two subtasks: intent detection and slot tagging (Wang et al., 2005; Tur and De Mori, 2011). Although promising progress has been achieved on SLU in English (Liu and Lane, 2016; Peng et al., 2020; Huang et al., 2020), those methods need large amounts of training data, and thus cannot be applied to low-resource languages where zero or few training data is available.

In this paper, we target at the extreme setting for cross-lingual SLU where no labeled data in target languages is assumed, which is critical for industry practice, since annotating a large SLU dataset with high quality for every language is simply infeasible.

Existing cross-lingual transfer learning methods mainly build on pre-trained cross-lingual word embeddings (Ruder et al., 2019) or contextual models (Wu and Dredze, 2019; Huang et al., 2019; Lample and Conneau, 2019; Conneau et al., 2020), which represent texts with similar meaning in different languages close to each other in a shared vector space. Those approaches often show good performance on intent detection. The results on slot tagging, however, are often unsatisfactory, especially for distant languages, which are dramatically different from English in scripts, morphology, or syntax (Upadhyay et al., 2018; Schuster et al., 2019; Li et al., 2020a).

Several studies (Schuster et al., 2019; Liu et al., 2020b; Li et al., 2020a) show that adding translated data into the fine-tuning process of pretrained models can improve the results of cross-lingual SLU substantially when no golden-labeled training data in target languages is available. For example, machine translation can be employed to translate the training data in English into target languages, and some alignment methods, such as attention weights (Schuster et al., 2019), fastalign (Dyer et al., 2013) or giza++ (Och and Ney, 2003), can be further applied to label the translated data. Another approach to alleviate the problem of data scarcity is to automatically generate training data. Recently, some methods for monolingual SLU (Anaby-Tavor et al., 2020; Kumar et al., 2020; Zhao et al., 2019; Peng et al., 2020) automatically label domain-specific data or use pre-trained language models to generate additional data.

However, the synthesized training data derived from both the translation approach and the data generation approach may be quite noisy and may contain errors in label. For the translation approach, both the translation process and the alignment process may generate errors (Xu et al., 2020; Li et al., 2020b). For the data generation approach, it is often hard to control a right tradeoff between generating correct but less diverse data and generalizing more diverse data but with more noise. Moreover, generating synthetic training data across languages further adds challenges to the robustness of the

*Work is done during internship at NLP Group, Microsoft STCA.

†Corresponding author.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3226–3237
November 7–11, 2021. ©2021 Association for Computational Linguistics

generation methods.

To filter out noise in the synthesized training data, a few methods are proposed, such as the mask mechanism (Li et al., 2020a), the soft alignment method (Xu et al., 2020), the unsupervised adaptation method (Li et al., 2020b), the rule-based filtering method (Peng et al., 2020), the classifier based filtering method (Anaby-Tavor et al., 2020) and the language model score based filtering method (Shakeri et al., 2020). These methods rely on either adhoc rules or extra models. Although they have shown promising results, each of them considers only a single source for data augmentation. It is still challenging to differentiate noisy instances from useful ones, since all those instances are sampled from the same distribution generated by the same method.

In this paper, we regard both the translation approach and the generation approach as data augmentation methods. We tackle the problem of reducing the impact of noise in augmented data sets. We develop a principled method to learn from multiple noisy augmented data sets for cross-lingual SLU, where no golden-labeled target language data exists. Our major technical contribution consists of a series of denoising methods including *instance relabeling*, *co-training* and *instance re-weighting*.

First, motivated by the self-learning method (Zoph et al., 2020), we design a model-ensemble-based instance relabeling approach to correct the noisy labels of augmented training data in low resource languages. As teacher models can not always generate correct labels, the original self-learning method tends to suffer from accumulated errors caused by model predictions. To alleviate the problem of accumulated errors, in our instance relabeling approach, we use crowd-intelligence from multiple models to derive the more reliable labels of pseudo training instances. Besides, we filter out noisy instances based on co-training and instance re-weighting strategies to reduce the impact of incorrect predictions on subsequent training. Our training strategy does not follow a traditional teacher-student manner. Instead, we use model predictions in the last epoch as pseudo labels in the current epoch to compute loss which saves training time.

In order to filter out noisy instances, we adopt a co-training mechanism, which uses selected instances from the other models to train the current model. Different from the co-teaching method (Han et al., 2018) where two models are trained with the same data, we propose multiple models should be trained with multiple different noisy augmented data where noise may be largely independent. It is because deep neural networks have a high capacity to fit noisy labels. When two models are trained with the same data, we tend to obtain two similar models. The co-teaching method gradually becomes a naive selection method with two models ensembled. In our co-training method, by employing very different data generation methods, we hope to attain that the noise from different sources may be largely independent and models can learn different knowledge from them. Therefore, the instances that pass the screening process of the other models can serve as the supervision signals to the current model which alleviates the problem of accumulated errors caused by selection bias.

Last, we further propose an instance reweighting technique to adjust the weights of training instances adaptively. As we do not have real training data in target languages, we can use the consistency among the soft labels predicted by different models to predict the reliability of the instances. Intuitively, if the predictions of different models are highly inconsistent on an instance, the instance may contain much noise. The larger deviation, the more uncertainty, and the less weight. This idea further increases the robustness of the selected training instances.

We conduct extensive experiments on two public datasets. The experiment results clearly indicate that, by consciously considering multiple noisy data sources derived from very different augmentation methods, our approach is more effective than using any single source. Our methods improve the state of the art (SOTA) by 3.05 and 4.24 percentage points on the two benchmark datasets, respectively.

2 Related Work

The cross-lingual spoken language understanding methods can be divided into two main categories: the model transfer methods and the data transfer methods.

The model transfer methods build on pre-trained cross-lingual models to learn language agnostic representations, such as MUSE (Lample et al., 2017), CoVE (McCann et al., 2017), mBERT (Wu and Dredze, 2019), XLM (Lample and Conneau, 2019), Unicoder (Huang et al., 2019), and XLM-R (Conneau et al., 2020). The English training data is applied to fine-tune the pre-trained models and then the fine-tuned models are directly applied to target languages (Liu et al., 2020b; Schuster et al., 2019; Qin et al., 2020; Upadhyay et al., 2018; Li et al., 2020a). To better align embeddings between source and target languages, Liu et al. (2019) use domain-related word pairs and employ a latent variable model to cope with the variance of similar sentences across different languages. Liu et al. (2020b) and Qin et al. (2020) use parallel word pairs to construct code-switching data for fine-tuning. Their methods encourage the model to align similar words in different languages into the same space and attend to keywords. Liu et al. (2020c) propose a regularization approach to align word-level and sentence-level representations across languages without any external resource.

The data transfer methods construct pseudo-labeled data in target languages. These methods usually employ machine translators to translate training instances in a source language into target languages and then apply alignment methods, such as attention weights (Schuster et al., 2019), fastalign (Dyer et al., 2013), or giza++ (Och and Ney, 2003), to project slot labels to the target language side. The derived training instances are combined with the training data in the source language to fine-tune the pre-trained cross-lingual models. Previous studies (Upadhyay et al., 2018; Schuster et al., 2019; Li et al., 2020a) show that adding translated training data can significantly improve the model performance, especially on languages which are distant from the source language.

The data generation approaches can also construct additional training data. Some methods (Wang and Yang, 2015; Marivate and Sefara, 2020; Gao et al., 2020) make slight changes to the original training instances through word replacement or paraphrases. More sophisticated methods generate training data through large-scale neural networks, such as generative adversarial networks (Goodfellow et al., 2020), variational autoencoders (Doersch, 2016; dos Santos Tanaka and Aranha, 2019; Russo et al., 2020), and pre-trained language models (Wu et al., 2019; Anaby-Tavor et al., 2020; Kumar et al., 2020; Peng et al., 2020).

3 Method

In this section, we define the problem and then propose our method.

3.1 Problem Definition and Solution Framework

The SLU task aims to parse user queries into a predefined semantic representation format. Formally, given an utterance $\boldsymbol{x} = \{x_i\}_{i=1}^{L}$ with a sequence of L tokens, a SLU model targets to produce an intent label $\boldsymbol{y}^I$ for the whole utterance and a sequence of slot labels $\boldsymbol{y}^S = \{\boldsymbol{y}_i^S\}_{i=1}^{L}$, where $\boldsymbol{y}_i^S$ is the slot label for the i^{th} token x_i. Here, we target at the extreme cross-lingual setting where only some training data $\mathcal{D}_{src}$ in English (or, in general, a rich-resource source language) and some development data $\mathcal{D}_{dev}$ in English exist. Besides, some annotated data $\mathcal{D}_{test}$ in target languages is used as the test set. Cross-lingual SLU is to learn a model by leveraging $\mathcal{D}_{src}$ to perform well on $\mathcal{D}_{test}$, using $\mathcal{D}_{dev}$ for parameter tuning.

We add a special token [CLS] in front of each input sequence. Then we feed $\boldsymbol{x}$ into an encoder $\mathcal{M}_{enc}$ to obtain the contextual hidden representation $\boldsymbol{H} = \{\boldsymbol{h}_i\}_{i=0}^{L}$, that is, $\boldsymbol{H} = \mathcal{M}_{enc}(\boldsymbol{x}; \Theta)$, where Θ denotes the parameters of the encoder.

We take $\boldsymbol{h}_0$ as the sentence representation for intent classification and take $\boldsymbol{h}_i$ $(1 \leq i \leq L)$ as the token representations for slot filling. We apply linear transformation and the *softmax* operation to obtain the intent probability distribution $\boldsymbol{p}^I(\boldsymbol{x}; \Theta)$ and the slot probability distribution $\boldsymbol{p}_i^S(\boldsymbol{x}; \Theta)$, that is,

$$\begin{aligned} \boldsymbol{p}^I(\boldsymbol{x}; \Theta) &= \textit{softmax}(\boldsymbol{W}^I \cdot \boldsymbol{h}_0 + \boldsymbol{b}^I) \\ \boldsymbol{p}_i^S(\boldsymbol{x}; \Theta) &= \textit{softmax}(\boldsymbol{W}^S \cdot \boldsymbol{h}_i + \boldsymbol{b}^S) \end{aligned} \tag{1}$$

where $\boldsymbol{p}^I \in \mathbb{R}^{1\times|C^I|}$, $\boldsymbol{p}_i^S \in \mathbb{R}^{1\times|C^S|}$, C^I is the set of intent labels, C^S is the set of slot labels under the BIO annotation schema (Ramshaw and Marcus, 1999), $\boldsymbol{W}^I \in \mathbb{R}^{|C^I|\times d}$ and $\boldsymbol{W}^S \in \mathbb{R}^{|C^S|\times d}$ are the output matrices, and $\boldsymbol{b}^I$ and $\boldsymbol{b}^S$ are the biases.

The overall architecture of our proposed method is shown in Figure 1. It consists of two major modules, the data augmentation module and the denoising module.

3.2 The Data Augmentation Module

In this module, we augment training data in target languages via translation and generation. The left part of Figure 1 shows the details.

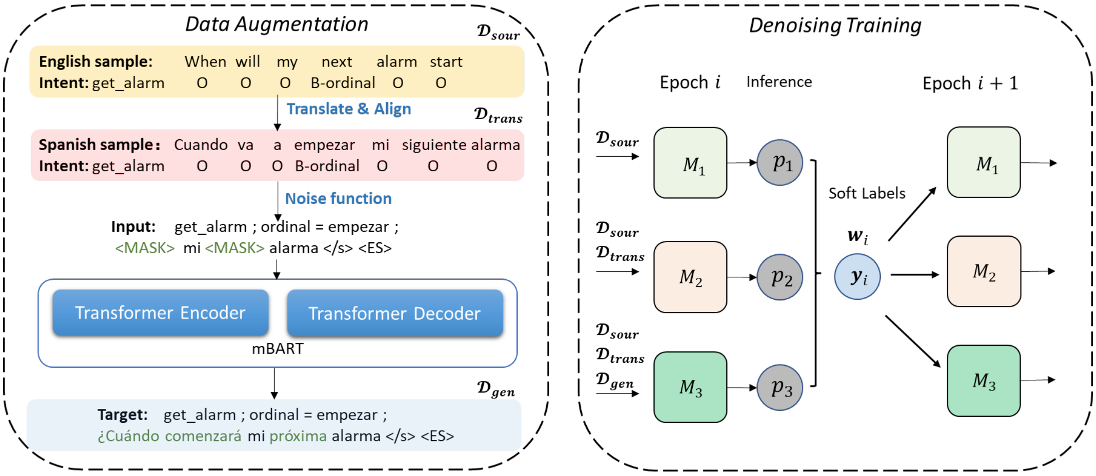

Figure 1: The overall framework of data augmentation module (left) and denoising module (right).

3.2.1 Translation

We use Google Translator to translate the training corpus $\mathcal{D}_{src}$ in the source language (English) to the target languages. In addition to translation, we also need some word alignment methods to project the slot labels to the target language side. We try giza++ (Och and Ney, 2003) and fastalign (Dyer et al., 2013) to obtain word alignment information and find that the pseudo slot labels projected by giza++ generally lead to better performance (about 2% increase in F1 on the SNIPS dataset). Thus, in the rest of the paper, we use Google translator and giza++ to produce translated data. Denote by $\mathcal{D}_{trans}$ the translated training corpus.

3.2.2 Generation

To further increase the diversity of synthesized training data, we leverage multilingual BART (mBART) (Liu et al., 2020a) as the generator to synthesize additional target language training corpus. Specifically, we first fine-tune the pretrained mBART model on the translated training data $\mathcal{D}_{trans}$ by adopting the denoising objective (Liu et al., 2020a) - the cross-entropy loss between the decoder's output and the original input. The input to mBART consists of the dialog act and the utterance in $\mathcal{D}_{trans}$, defined by

$$\boldsymbol{X} = [I; (s_1, v_1), ..., (s_p, v_p); (x_1, ..., x_L)] \quad (2)$$

where I is the intent and $(s_i, v_i)_{i=1}^{p}$ are the slot-value pairs. Here, v_i in the target language is obtained by word alignment between utterances in the source and the target language. Following Liu et al. (2020a), we apply text infilling as the injected noise in the fine-tuning stage.

After fine-tuning, we apply the same noise function to the input data $\boldsymbol{X}$ and leverage the fine-tuned mBART to generate m candidates for each instance. To increase diversity of generated data, the top-p sampling strategy (Fan et al., 2018) is adopted. Each generated instance consists of "an utterance, the corresponding intent and slot-value pairs". Then, we perform preliminary data selection by filtering out the generated utterances not containing the required slot values. Last, we randomly sample a instances from the candidate set for each input to construct the generated corpus $\mathcal{D}_{gen}$.

3.3 The Denoising Module

To tackle the noisy label problem introduced by the data augmentation module, we design a denoising module shown in Algorithm 1.

At the initialization stage, we first train K models using the augmented data derived from K different augmentation methods. All models are optimized by the cross-entropy loss function computed using the original labels of intent and slots. For the k-th model ($k \leq K$),

$$\mathcal{L}^k(\boldsymbol{x}) = -\frac{1}{L}\sum_{j=1}^{L}\boldsymbol{y}_j^S log \boldsymbol{p}_j^S(\boldsymbol{x};\Theta_k) - \boldsymbol{y}^I log \boldsymbol{p}^I(\boldsymbol{x};\Theta_k) \quad (3)$$

where $\boldsymbol{x}$ is a training utterance, $\boldsymbol{y}^I$ is the intent label, $\boldsymbol{y}_j^S$ is the slot label of the j-th word in the utterance, and $\boldsymbol{p}^I(\boldsymbol{x};\Theta_k)$ and $\boldsymbol{p}_j^S(\boldsymbol{x};\Theta_k)$ are the predicted probability distributions of the intent and the slot, respectively.

To keep our discussion simple, in this paper, we mainly consider using the training corpora derived from machine translation and generation. Thus, we maintain $K = 2$ SLU networks with the same structure in the training process. $\mathcal{M}_1$ and $\mathcal{M}_2$ are trained using different training corpora $\mathcal{D}_1 = \{\mathcal{D}_{src}, \mathcal{D}_{trans}\}$ and $\mathcal{D}_2 = \{\mathcal{D}_{src}, \mathcal{D}_{trans}, \mathcal{D}_{gen}\}$, respectively. Based on our epxeriments, $\mathcal{D}_{gen}$ only is too noisy thus we combine with $\mathcal{D}_{trans}$. Our training framework in general can handle more than 2 models. We present experimental results with more than 2 models in Section 4.4.

After the initialization stage, each model has learned some knowledge from each augmented training data. Since there exists noise in the augmented training data (such as $\mathcal{D}_{trans}$ and $\mathcal{D}_{gen}$), we step into the relabeling stage, which combines a series of strategies: *instance relabeling*, *co-training* and *instance re-weighting* to reduce the impact of the noise.

Algorithm 1: Denoising Module

```
Data: $\mathcal{M}_k(1 \leq k \leq K)$: SLU networks; $E_{all}$: the
      number of fine-tuning epochs; $E$: the number
      of initialization epochs; $\mathcal{D}_{src}, \mathcal{D}_{trans}, \mathcal{D}_{gen}$:
      training corpus;
Result: $\mathcal{M}_k(1 \leq k \leq K)$
// Initialization Stage
for $e = 1; e < E; e++$ do
    shuffle training corpus $\mathcal{D}_k$ from
      $\{\mathcal{D}_{src}, \mathcal{D}_{trans}, \mathcal{D}_{gen}, ...\}$
    update $\mathcal{M}_k$ with $\mathcal{L}^k(\mathcal{D}_k)$
end
// Relabeling Stage
for $e = E; e \leq E_{all}; e++$ do
    shuffle training corpus
      $\mathcal{D} = \{\mathcal{D}_{src}, \mathcal{D}_{trans}, \mathcal{D}_{gen}, ...\}$ into $B$
      mini-batches;
    for $j = 1; j <= B; j++$ do
        Fetch $j^{th}$ mini-batch $\mathcal{D}'$ from $\mathcal{D}$
        // Co-Training
        $\mathcal{D}'_k =$
          $argmin_{\mathcal{D}':|\mathcal{D}'| \geq (1-\delta)|\mathcal{D}'|} \sum_i^{K \backslash k} \mathcal{L}^i(\mathcal{D}')$
        // Instance Re-weighting
        update $\mathcal{M}_k$ with $\mathcal{L}^{k'}(\mathcal{D}'_k)$
        // Instance Relabeling
        update labels of instances in $\mathcal{D} \setminus \mathcal{D}_{src}$ as
          Equation 4
    end
end
```

3.3.1 Instance Relabeling

In the relabeling stage, model training and instance relabeling are conducted iteratively. Motivated by the idea of model ensemble, we use the ensemble of model predictions to correct label errors in a self-learning manner. Specifically, all models are trained using all training corpora $\mathcal{D} = \{\mathcal{D}_{src}, \mathcal{D}_{trans}, \mathcal{D}_{gen}\}$. The slot labels and the intent labels of the training instances in $\mathcal{D}_{trans}$ and $\mathcal{D}_{gen}$ are modified to the corresponding ensemble predicted probability distributions, which are used as the pseudo-truth labels to compute the loss in the next epoch. That is,

$$\widetilde{\boldsymbol{y}}^I = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{p}^I(\boldsymbol{x};\Theta_k); \quad \widetilde{\boldsymbol{y}}_j^S = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{p}_j^S(\boldsymbol{x};\Theta_k) \quad (4)$$

3.3.2 Co-Training

Heuristically, instances with small losses are more likely to have cleaner labels. When noise from different augmentation methods is more or less independent, each model can learn instances with small cross-entropy losses from the other models.

Specifically, when $K = 2$, in each batch of the training data, each network discards the instances with larger losses computed by Equation 3 by a ratio of δ and then teach the remaining instances to another one.

3.3.3 Instance Re-weighting

Another way to reduce the impact of noisy instances is to assign different weights to different instances, the more noisy an instance, the less weight it is associated. We design a re-weighting mechanism to implement this idea. The intuition is that if the predictions by multiple models are quite inconsistent, the instance may likely be noisy. Technically, we design an uncertainty based weight to re-weight the training instances. The larger the deviation, the more uncertainty and the less the weight. Specifically, the uncertainty of each instance is defined as:

$$u = \frac{1}{K}\sum_{k=1}^{K} Var(\boldsymbol{p}(\boldsymbol{x};\Theta_k), \boldsymbol{m}(x)) \quad (5)$$

where

$$Var(\boldsymbol{p}(\boldsymbol{x};\Theta_k), \boldsymbol{m}(x)) = (\boldsymbol{p}^I(\boldsymbol{x};\Theta_k) - \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{p}^I(\boldsymbol{x};\Theta_k))^2$$
$$+ \frac{1}{L}\sum_{l=1}^{L}(\boldsymbol{p}_l^S(\boldsymbol{x};\Theta_k) - \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{p}_l^S(\boldsymbol{x};\Theta_k))^2 \quad (6)$$

We further compute weight by $w = e^{-u}$ and incorporate this weight into Equation 3 to obtain:

$$\mathcal{L}^{k'}(\boldsymbol{x}) = -w[\frac{1}{L}\sum_{j=1}^{L}\widetilde{\boldsymbol{y}}_j^S log\boldsymbol{p}_j^S(\boldsymbol{x};\Theta_k) + \widetilde{\boldsymbol{y}}^I log\boldsymbol{p}^I(\boldsymbol{x};\Theta_k)] \quad (7)$$

which is the new training objective during the relabeling stage.

4 Experiments

In this section, we report our experiments on two benchmark datasets.

	SNIPS (F1/Intent Accuracy)			**MTOP** (Exact Match Accuracy)					
	es	th	Average	es	fr	de	hi	th	Average
Multi. CoVe	19.25/53.89	35.62/70.70	27.44/62.30	-	-	-	-	-	-
Transferable Latent Variable	65.79/90.20	32.24/73.47	49.02/81.84	-	-	-	-	-	-
Attention-Informed Mixed	73.89/87.88	27.12/73.46	50.51/80.67	-	-	-	-	-	-
CoSDA-ML	80.40/94.80	37.30/76.80	58.85/85.80	-	-	-	-	-	-
LR&ALVM	72.49/92.31	33.28/75.77	52.89/84.04	-	-	-	-	-	-
EN	84.20/97.70	46.00/90.40	65.10/94.05	69.10	65.40	64.00	55.00	43.80	59.46
EN + Trans.	73.40/98.30	50.90/96.60	62.15/97.45	74.50	72.60	64.70	58.30	56.50	65.32
EN + Trans. + mask	83.00/98.00	50.20/96.60	66.60/97.30	74.60	72.20	65.70	62.50	53.20	65.64
EN_{reimp}	84.69/97.67	39.34/89.42	62.02/93.55	68.38	70.48	66.15	54.81	38.36	59.63
EN + $\text{Trans.}_{\text{reimp}}$	83.58/99.34	79.29/97.93	81.44/98.64	70.25	67.27	70.39	64.22	55.80	65.59
EN + Trans. + Gen. + Denoise	86.19/99.37	82.79/98.17	**84.49/98.71**	71.58	71.53	74.05	69.66	62.31	**69.83**

Table 1: Comparison results between our approach and prior SOTA methods. The top block rows are baselines that belong to model transfer methods. The middle block rows are XLM-R large model based approaches, w/ or w/o translated training data. Methods with $_{\text{reimp}}$ are re-implemented in this paper with different translator and alignment tool. EN, Trans. and Gen. denote source language data $\mathcal{D}_{src}$, translated target language data $\mathcal{D}_{trans}$ and generated target language data $\mathcal{D}_{gen}$, respectively. Denoise denotes proposed denoising module.

4.1 Settings

We evaluate the effectiveness of our proposed approach over five languages on two benchmark datasets: **SNIPS** (Schuster et al., 2019) and **MTOP** (Li et al., 2020a). The details of datasets are provided in Appendix.

For generation, we generate $m = 10$ candidates for each input and randomly sample $a = 1$ from each candidate set to construct $\mathcal{D}_{gen}$. Our SLU model is based on the pre-trained XLM-R large model, which has $L = 24$ layers and two additional task-specific linear layers for intent classification and slot filling. More implementation details including hyper-parameters are described in Appendix.

Following the previous works (Schuster et al., 2019; Li et al., 2020a), we use F1 score to measure the slot filling quality and use accuracy score to evaluate the intent classification quality on the SNIPS dataset and use Exact Match Accuracy on the MTOP dataset.

We employ the following SOTA baselines in two groups. The first group is the model transfer methods, including **Multi.CoVe** (Schuster et al., 2019); **Transferable Latent Variable** (Liu et al., 2019); **Attention-Informed Mixed** (Liu et al., 2020b); **CoSDA-ML** (Qin et al., 2020); **LR&ALVM** (Liu et al., 2020c); and **EN** (Li et al., 2020a). The second group is the data transfer methods, including **EN+Trans.** (Li et al., 2020a) and **EN+Trans.+mask** (Li et al., 2020a).

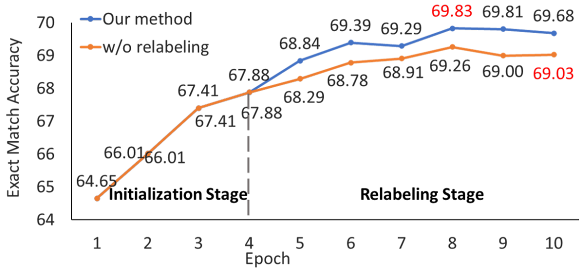

Figure 2: Metric trend per epoch on MTOP w/ or w/o instance relabeling strategy.

4.2 Results

Table 1 reports the results of our approach and the SOTA baselines. As the translator used in Li et al. (2020a) is not publicly available, we use Google translator instead, which leads to some results on some languages slightly different from reported by Li et al. (2020a).

Our method outperforms the SOTA baselines and achieves new SOTA performance. Our method improves the Exact Match Accuracy on MTOP from 65.59 to 69.83, the F1 score on SNIPS from 81.44 to 84.49, and the accuracy on SNIPS from 98.64 to 98.71. These results clearly demonstrate the effectiveness of our proposed method.

One interesting finding is that the performances on Spanish and French become slightly worse after adding the translated data. It is because the noise introduced by the machine translation and alignment processes may hurt the performance. Our method introduces the denoising training approach, which is able to handle the noise of synthesized data.

	es	fr	de	hi	th	Average
Classifier based selection	71.45	69.21	71.68	66.37	60.43	67.83
LM based selection	70.41	68.78	70.95	65.94	58.44	66.90
Classifier based selection w/ relabeling	72.01	70.06	73.29	69.01	60.25	68.92
LM based selection w/ relabeling	71.38	69.28	72.95	67.73	61.19	68.51
EN + Trans. + Gen. + Denoise	71.58	71.53	74.05	69.66	62.31	**69.83**

Table 2: Comparison with other denoising methods on the MTOP dataset.

	es	fr	de	hi	th	Average
EN + Trans.	70.25	67.27	70.39	64.22	55.80	65.59
EN + Gen.	66.71	65.99	70.84	63.55	45.32	62.48
EN + Trans. + Gen.	69.21	66.93	71.46	65.48	56.89	65.99
EN + Trans. + Gen. + Denoise	71.58	71.53	74.05	69.66	62.31	**69.83**
w/o Gen.	71.08	70.28	72.67	70.05	62.71	69.36 ↓
w/o instance relabeling	71.98	70.15	72.78	69.37	60.87	69.03 ↓
w/o co-training	71.71	69.56	73.15	68.94	61.41	68.95 ↓
w/o instance re-weighting	72.18	70.84	73.99	68.55	62.82	69.68 ↓

Table 3: Ablation study on the MTOP dataset.

4.3 Comparison with Other Denoising Methods

To verify the effectiveness of our approach, we conduct experiments with two kinds of denoising approaches used by the previous works. First, we consider the **classifier based selection** approach. Following Anaby-Tavor et al. (2020), we train an extra classifier using the corpus in English and the translated corpus, and filter out noisy data according to the probability scores predicted by the classifier. Second, we consider the **LM based selection** approach. Following Shakeri et al. (2020), we use the language model score as the indicator to select high-quality data.

For fair comparisons, for each noise filtering baseline, we also fine-tune two pre-trained XLM-R large models using different random seeds and take the ensembled model as the final model. Additionally, to remove the effect of instance relabeling method, we also apply it to baselines similar to our approach. Table 2 shows the comparison results on the MTOP dataset. Our approach outperforms those two methods (w/ relabeling) by 0.91 and 1.32 percentage points, respectively. This suggests that the gain of our method is not from the simple ensemble of two models. Instead, our method could indeed effectively remove the noise of synthesized data, outperforming previous noise filtering methods.

4.4 Ablation Study

To validate the contribution of each component in our method, we conduct the ablation study on the MTOP dataset. We consider several ablation options. (1) **w/o generation** removes the generated training data. (2) **w/o instance relabeling** keeps the intent and slot labels of data unchanged throughout the training process. (3) **w/o co-training** trains models using all training data without filtering. (4) **w/o instance re-weighting** skips the instance re-weighting strategy.

As shown in Table 3, compared with the performance of approach using translated and source language data, the performance of approach using generated data and source language data is reduced by 3.11 percentage points without denoising strategies. This is due to the much noise introduced by the generation process, which hurts the performance. When combining with our denoising approach, the approach with generated training corpus is superior to the approach without generated corpus by 0.47 percentage points. We consider that multiple augmented data sets increase data diversity and lead to better supervision signals. Table 3 also shows that removing any of the other components generally leads to clear performance drop. It confirms that all of the proposed techniques contribute to the cross-lingual setting.

4.5 Effect of Instance Relabeling

To better understand the effect of the instance relabeling strategy, in Figure 2, we record the Exact Match Accuracy of our method with or without the relabeling strategy on the MTOP test set after each training epoch. The performance of our method with relabeling strategy keeps improving and is consistently better than the baseline during

Type	Source	Lang	Utterance	Original Labels	Modified Labels
Intent	Gen.	fr	je suis tout disponible	get_available	set_available
Intent	Gen.	de	verschieben sie die erinnerung in den alarm	update_reminder	delete_reminder
Slot	Trans.	es	¿cómo es el clima en colorado en esta época del año?	None	location
Boundary	Trans.	es	resumen de noticias [de tayer]	data_time	data_time
Slot&Boundary	Gen.	de	lesen sie mir [die heutigen] schlagzeilen	news_source	data_time

Table 4: Cases of relabeling results on the MTOP dataset.

	One model	Two models	Three models
EN+Trans.+Gen.+Denoise	67.57	69.83	**70.04**

Table 5: Performance of approaches with different number of models on the MTOP dataset.

	0%	10%	20%	30%	40%
EN+Trans.+Denoise	68.32	68.69	**69.36**	69.07	68.75
EN+Trans.+Gen.+Denoise	68.95	68.77	69.12	**69.83**	69.07

Table 6: Performance with different filtering rates and training corpus.

the relabeling stage. It demonstrates that the relabeling method indeed corrects many label errors in the noisy training data and the corrected labels contribute to the performance improvement.

4.6 Effect of Number of Models

We explore the effect of the number K of models (XLM-R). Specifically, we conduct experiments using one or three models. In the setting of one model, that is $K = 1$, we only train one network with all training corpora $\mathcal{D} = \{\mathcal{D}_{src}, \mathcal{D}_{trans}, \mathcal{D}_{gen}\}$, and adopt the instance relabeling and instance filtering strategies. In the setting of $K = 3$, three models are trained using $\{\mathcal{D}_{src}, \mathcal{D}_{trans}\}$, $\{\mathcal{D}_{src}, \mathcal{D}_{gen}\}$ and $\{\mathcal{D}_{src}, \mathcal{D}_{trans}, \mathcal{D}_{gen}\}$, respectively.

The results shown in Table 5 indicate that our method can effectively extend beyond two models. When the number of networks increases, the performance improves. The intuition is that more models can produce more reliable predictions, and thus can lead to better instance relabeling as well as instance filtering.

4.7 Effect of Filtering Rate in Co-Training

To study the impact of the co-training strategy, we conduct experiments with different filtering rates on the MTOP dataset. Table 6 shows the results with regard to different filtering rates and different training corpora. For both approaches using or not using the generated corpus, as the filtering rate increases, the performance improves as well. This demonstrates that the filtering strategy can indeed filter out noisy instances effectively. However, further increasing the filtering rate degrades the performance. It is mainly because of the excessive drop of useful information contained in the training data. Another finding is that the best filtering rate for training corpus $\{\mathcal{D}_{src}, \mathcal{D}_{trans}, \mathcal{D}_{gen}\}$ is larger than that for $\{\mathcal{D}_{src}, \mathcal{D}_{trans}\}$. The explanation may be that the generated corpus $\mathcal{D}_{gen}$ has more diverse data than the translated corpus $\mathcal{D}_{trans}$, but may also contain more noise.

5 Case Study

We conduct case analysis of the instance relabeling results on the MTOP dataset to examine the capability of our approach. We statistically analyze the differences between the original labels and the modified labels after relabeling stage.

We find that our instance relabeling method effectively corrects wrong labels of the synthesized data, including intent label and slot label as shown in Table 4. Specifically, there are four types of label modifications: 1) *Intent Change*: the intent label of an utterance is modified; 2) *Slot Change*: the slot type of a text span is modified; 3) *Boundary Change*: the BIO boundaries of a slot are modified; and 4) *Slot and Boundary Change*: both the slot type and the BIO boundaries are modified. For the MTOP dataset, intent labels of 4.99% of the translated and generated data are modified and the slot labels of 33.10% of those data sets are modified.

From the case study, we can see that the synthesized data indeed contains much noise and our relabeling strategy is able to greatly reduce the negative impact of the noise by correcting different types of label errors.

6 Conclusions

In this paper, we propose a denoising training approach where multiple models trained from various augmented methods provide supervision signals to each other. Extensive experimental results show that the proposed method outperforms the previous approaches, and can certainly alleviate the noisy label problem. Our proposed method is independent of the backbone network (e.g., XLM-R model) and the task. As future work, we plan to investigate the performance of our method on different cross-lingual tasks.

Acknowledgments

Jian Pei's research is supported in part by the NSERC Discovery Grant program. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *AAAI*, pages 7383–7390.

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, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Carl Doersch. 2016. Tutorial on variational autoencoders. *CoRR*, abs/1606.05908.

Fabio Henrique Kiyoiti dos Santos Tanaka and Claus Aranha. 2019. Data augmentation using gans. *CoRR*, abs/1904.09135.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. 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 889–898. Association for Computational Linguistics.

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase augmented task-oriented dialog generation.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 2485–2494. Association for Computational Linguistics.

Zhiqi Huang, Fenglin Liu, and Yuexian Zou. 2020. Federated learning for spoken language understanding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3467–3478, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *CoRR*, abs/2003.02245.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2020a. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. *arXiv preprint arXiv:2008.09335*.

Xin Li, Lidong Bing, Wenxuan Zhang, Zheng Li, and Wai Lam. 2020b. Unsupervised cross-lingual adaptation for sequence tagging and beyond.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020a. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:726–742.

Zihan Liu, Jamin Shin, Yan Xu, Genta Indra Winata, Peng Xu, Andrea Madotto, and Pascale Fung. 2019. Zero-shot cross-lingual dialogue systems with transferable latent variables. 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 1297–1303. Association for Computational Linguistics.

Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020b. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8433–8440. AAAI Press.

Zihan Liu, Genta Indra Winata, Peng Xu, Zhaojiang Lin, and Pascale Fung. 2020c. Cross-lingual spoken language understanding with regularized representation alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7241–7251. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Vukosi Marivate and Tshephisho Sefara. 2020. Improving short text classification through global augmentation methods. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 385–399. Springer.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in neural information processing systems*, pages 6294–6305.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Baolin Peng, Chenguang Zhu, Michael Zeng, and Jianfeng Gao. 2020. Data augmentation for spoken language understanding via pretrained models. *CoRR*, abs/2004.13952.

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.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Giuseppe Russo, Nora Hollenstein, Claudiu Cristian Musat, and Ce Zhang. 2020. Control, generate, augment: A scalable framework for multi-attribute text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 351–366. Association for Computational Linguistics.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. 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 3795–3805. Association for Computational Linguistics.

Siamak Shakeri, Noah Constant, Mihir Sanjay Kale, and Linting Xue. 2020. Multilingual synthetic question and answer generation for cross-lingual reading comprehension. *CoRR*, abs/2010.12008.

Gokhan Tur and Renato De Mori, editors. 2011. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. Wiley.

S. Upadhyay, M. Faruqui, G. Tur, H. Dilek, and L. Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2557–2563.

Ye-Yi Wang, Li Deng, and Alex Acero. 2005. Spoken language understanding "" an introduction to the statistical framework. *IEEE Signal Processing Magazine*, 22(5):16–31.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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 833–844. Association for Computational Linguistics.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *International Conference on Computational Science*, pages 84–95. Springer.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5052–5063.

Zijian Zhao, Su Zhu, and Kai Yu. 2019. Data augmentation with atomic templates for spoken language understanding. 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 3635–3641. Association for Computational Linguistics.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. 2020. Rethinking pre-training and self-training. *arXiv preprint arXiv:2006.06882*.

A Experimental Setting

A.1 Datasets

We evaluate the effectiveness of our proposed approach over five target languages on two benchmark datasets: **SNIPS** (Schuster et al., 2019) and **MTOP** (Li et al., 2020a). Statistics of used data are detailed in Table 7.

	SNIPS	**MTOP**
Language	en,es,th	en,es,fr,de,hi,th
Intent types	12	117
Slot types	11	78
Train	30,521	13,151
Dev	4,181	1,878
Test	4,735	16,442

Table 7: Statistics of datasets.

A.2 Implementation Details

For generation, we fine-tune mBART pre-trained on 25 languages with 0.3 dropout, 0.2 label smoothing, 2500 warm-up steps, 3×10^{-5} maximum learning rate, and 1024 tokens in each batch. For text filling, we mask 35% of the words in each instance by randomly sampling a span length according to a Poisson distribution ($\lambda = 3.5$). Then we append to each instance an end-of-sentence token ($</S>$) and the corresponding language id symbol ($< LID >$). We don't search the best parameters for generation but use the default values in open-source code[*]. The final models are selected based on validation likelihood.

For SLU, we use XLM-R large model with about 550M parameters as the backbone network. In the fine-tuning process, we set the batch size as 128, fine-tuning epochs $E_{all} = 10$, initialization epochs $E = 4$ and 0.1 dropout for two benchmark datasets. The maximum filtering rates are $\delta = 0.2$ and $\delta = 0.3$ for the SNIPS and MTOP datasets, respectively. The learning rates are 2×10^{-5} and 5×10^{-5} for SNIPS and MTOP datasets, respectively. We select the best hyper-parameters by searching a combination of batch size, learning rate, the number of fine-tuning epochs, the number of initialization epochs and the filtering ratio with the following range: batch size $\{32, 64, 128\}$, learning rate $\{1, 2, 3, 4, 5\} \times 10^{-5}$, fine-tuning epochs $\{5, 10, 15\}$, initialization epochs $\{2, 3, 4\}$, filtering ratio $\{10\%, 20\%, 30\%, 40\%\}$. The models are saved by performance on the English development corpus and translated target language development corpus. The models are trained using mini-batch back-propagation, and the AdamW (Loshchilov and Hutter, 2019) optimizer is used for optimization. We fine-tune the models on two V100-32GB GPUs which lasts about 4 hours.

	0	1	2	3
EN+Trans.+Gen.+Denoise	69.36	69.83	**70.52**	69.98

Table 8: Performance with different sizes of generated data on the MTOP dataset.

						Mean	Stdev
Baseline	65.59	66.02	65.59	65.46	65.12	65.56	0.29
Our method	69.83	69.68	69.51	69.46	69.02	69.50	0.27

Table 9: Mean and Standard deviation (Stdev) values across five different runs for our approach and the baseline on the MTOP dataset.

B More Experimental Results and Discussions

B.1 Effect of Generated Data Size

To further analyze the effect of generated data, we randomly sample $a = \{0, 1, 2, 3\}$ instances from the candidate set for each input to construct the generated corpus. As shown in Table 8, by increasing the size of the generated corpus, the performance improves. However, when the data size reaches a certain scale, e.g., $a = 3$ in our experiments, the performance slightly regresses, but still outperforms the baseline without generated data. This suggests that the augmentation module indeed increases the diversity of training data and then improves the performance. Though increasing noise limits the growth of the improvement, our approach is robust enough to achieve comparable performance.

B.2 Variance Analysis

We conduct 5 runs of training and calculate the mean and standard deviation (Stdev) values for our approach and the baseline on the MTOP dataset. The results are listed in Table 9. Besides, we also conduct a two-sided statistically significant t-test with the significance threshold 0.05 to compare the baseline with our method. The results show that the variance of our approach is similar to that of the baseline. Moreover, with p-value = 4.5×10^{-8}, our method outperforms the baseline with statistical significance.

[*] https://github.com/pytorch/fairseq/tree/master/examples/mbart