# Improving Zero-Shot Cross-Lingual Transfer via Progressive Code-Switching

**Zhuoran Li**[1], **Chunming Hu**[1,2], **Junfan Chen**[1,2], **Zhijun Chen**[1], **Xiaohui Guo**[3] and **Richong Zhang**[1*]
[1]SKLSDE, School of Computer Science and Engineering, Beihang University, Beijing, China
[2]School of Software, Beihang University, Beijing, China
[3]Hangzhou Innovation Institute, Beihang University, Hangzhou, China
{lizhuoranget, hucm, zhijunchen}@buaa.edu.cn, {chenjf, guoxh, zhangrc}@act.buaa.edu.cn

## Abstract

Code-switching is a data augmentation scheme mixing words from multiple languages into source lingual text. It has achieved considerable generalization performance of cross-lingual transfer tasks by aligning cross-lingual contextual word representations. However, uncontrolled and over-replaced code-switching would augment dirty samples to model training. In other words, the excessive code-switching text samples will negatively hurt the models' cross-lingual transferability. To this end, we propose a **P**rogressive **C**ode-**S**witching (PCS) method to gradually generate moderately difficult code-switching examples for the model to discriminate from easy to hard. The idea is to incorporate progressively the preceding learned multilingual knowledge using easier code-switching data to guide model optimization on succeeding harder code-switching data. Specifically, we first design a difficulty measurer to measure the impact of replacing each word in a sentence based on the word relevance score. Then a code-switcher generates the code-switching data of increasing difficulty via a controllable temperature variable. In addition, a training scheduler decides when to sample harder code-switching data for model training. Experiments show our model achieves state-of-the-art results on three different zero-shot cross-lingual transfer tasks across ten languages.

## 1 Introduction

Zero-shot cross-lingual transfer learning aims to train an adaptable model on a source language that can effectively perform on others without labelled data in the target languages. This study is particularly valuable in circumstances where there are limited or no annotations available for the target languages. In recent years, the multilingual pre-trained language models, such as mBERT [Devlin *et al.*, 2019], XLM [Conneau and Lample, 2019] and XLM-R [Conneau *et al.*, 2020] have achieved significant performance improvements through fine-tuning on source language data and direct application to target language data (as shown in Figure 1(a)). Furthermore, it has been discovered that a further multilingual contextualized representation alignment improves the zero-shot cross-lingual transfer performance by exploiting a bilingual dictionary to replace some tokens in the source text with target-lingual translated words. This strategy, named *Code-Switching* (CS), usually randomly chooses substitution words and has been shown improvements in many zero-shot cross-lingual tasks [Liu *et al.*, 2020; Qin *et al.*, 2021; Zheng *et al.*, 2021; Ma *et al.*, 2022; Wang *et al.*, 2023b] (as shown in Figure 1(b)).

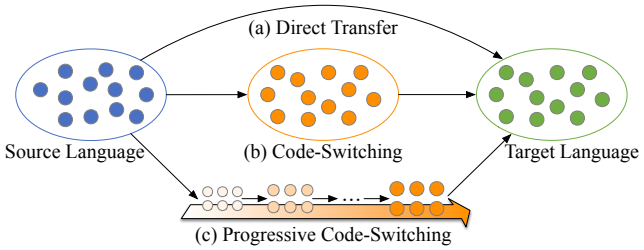



Figure 1: Illustration of our progressive code-switching cross-lingual idea. (a) Direct transfer from source to target. (b) Randomly generating code-switching data. (c) The proposed progressive code-switching method generates code-switching data for the model to discriminate from easy to hard. Larger and darker dots indicate harder code-switching data.

Code-Switching, as a data augmentation technique, on one hand, inevitably leads to losing original contextual information when over-replacing words with other lingual synonyms within a sentence. On the other hand, under-replaced code-switching results in insufficient cross-lingual alignment and limited data variation may limit the model's ability to learn and transfer knowledge across languages. Existing studies have indicated that such uncontrolled samples might not necessarily benefit model learning [Qin *et al.*, 2021; Yang *et al.*, 2021]. For example, an original English sentence is "*All the services were great.*" and its corresponding multilingual code-switching sentence is "*todas (ES) les (FR) services waren (DE) great.*", wherein capital letters in parentheses represent target language abbreviations (e.g. *ES* indicates *Spanish*). Because of very different contextual sentence expressions, such code-switched sentences may not contribute

* Corresponding author

to multilingual word representation alignment, but in some extreme cases, hurt the trained models' cross-lingual generalization ability. Therefore, we should devise some kind of switching scheme to govern the switching extent subtly, e.g., the number of substitutions or the model discrimination ability impact of code-switched sentences.

In this study, we assume that easy code-switching samples could act as pre-training knowledge, which guides the model optimization on harder code-switching data. For instance, we first consider the easy code-switching sentence "*All les (FR) services were great.*", which shares substantial contextual overlap with the original sentence. In this case, the model can correctly align the word pair "*the (EN) - les (FR)*". Then for a harder code-switching sentence, "*All les (FR) services waren (DE) great.*", the previously aligned "*the (EN) - les (FR)*" can serve as a pivot for aligning the new word pair "*were (EN) - waren (DE)*". By adopting this progressive strategy, new word pairs can be aligned based on the previously identified pairs. Consequently, even the hard code-switching data such as "*todas (ES) les (FR) services waren (DE) great.*" would become easier to be handled and thus progressively improving multilingual alignment.

To pursue both effective utilization of code-switching data and model generalization, we borrow the idea of curriculum learning [Bengio *et al.*, 2009] and propose a progressive code-switching framework termed PCS (as shown in Figure 1(c)). However, determining the difficulty of code-switching data is challenging, as the importance of each word varies for different tasks. Drawing inspiration from explanation learning methods, we develop a difficulty measurer to estimate the difficulty of code-switching sentences based on the contribution of substitution words toward the prediction. We then introduce a code-switcher with an adjustable temperature parameter to generate appropriate code-switching sentences that align with the current curriculum difficulty. Furthermore, to mitigate the problem of catastrophic forgetting in curriculum learning, we design a scheduler that dynamically adapts the difficulty level to revisit the previously acquired knowledge.

In summary, we make the following key contributions:

- We propose a progressive code-switching method for zero-shot cross-lingual transfer, which mitigates the negative impacts of uncontrolled code-switching data and improves the multilingual representation alignment.
- We introduce a word relevance score-guided difficulty measurer, a temperature-adjustable code-switcher, and a dynamic scheduler. They collaboratively regulate the switched samples' difficulty and gradually generate code-switching samples in a controlled manner.
- We comprehensively evaluate our proposed approach on three different cross-lingual tasks covering ten different languages. The results demonstrate that PCS substantially enhances performance compared to some strong code-switching baselines.

## 2 Related Work

**Zero-shot cross-lingual transfer** aims to learn a model with labelled source language data and perform well on other target languages. In recent years, there have been some pre-trained multilingual language models for cross-lingual transfer, such as mBERT [Devlin *et al.*, 2019], XLM [Conneau and Lample, 2019] and XLM-R [Conneau *et al.*, 2020; Goyal *et al.*, 2021]. Some studies further improve the alignment of multiple different languages by parallel corpora [Artetxe and Schwenk, 2019; Cao *et al.*, 2020; Pan *et al.*, 2021; Chi *et al.*, 2021; Wei *et al.*, 2021b]. Recently, code-switching leverages low-resource bilingual dictionary to align multilingual contextual representations and achieve the state-of-the-art performance in many cross-lingual tasks, such as text classification [Lee *et al.*, 2021], dialogue system [Liu *et al.*, 2020; Ma *et al.*, 2022], sequence tagging tasks [Feng *et al.*, 2022], and question answering [Nooralahzadeh and Sennrich, 2023]. There are additional attempts to avoid the original signal loss in code-switching [Lee *et al.*, 2021; Feng *et al.*, 2022]. Most of the code-switching work in word substitution is random and our work considers the negative impacts of excessive code-switching data. To the best of our knowledge, only a few studies have focused on word selection in code-switching, and none of them investigates progressive code-switching.

**Curriculum learning** is proposed as a machine learning strategy by feeding training examples to the model by a meaning order, which is inspired by the learning process of humans and animals [Bengio *et al.*, 2009]. In general, curriculum learning contains a difficulty evaluator used to evaluate the difficulty score of instances, and a scheduler used to decide how examples should be fed to the model. Curriculum learning has been successfully applied to many areas in natural language processing, such as question answering [Sachan and Xing, 2016], reading comprehension [Tay *et al.*, 2019], dialogue system [Shen and Feng, 2020; Zhu *et al.*, 2021] and text classification [Lalor and Yu, 2020; Xu *et al.*, 2020]. Another line of research aims at providing a theoretical guarantee of curriculum learning, including transfer learning method [Xu *et al.*, 2020] and optimization methods [Kumar *et al.*, 2010; Graves *et al.*, 2017]. In this work, we adopt curriculum learning into code-switching and address the negative impacts of code-switching in cross-lingual transfer.

## 3 Progressive Code-Switching

We introduce **P**rogressive **C**ode-**S**witching (PCS) method in this section, which can be applied to various zero-shot cross-lingual transfer learning downstream tasks. Figure 2 depicts an overview of the PCS framework. We will describe the details of our approach from the following four components: difficulty measurer, code-switcher, scheduler, and model trainer.

**Problem Formulation.** Formally in a zero-shot cross-lingual transfer task, given a source language sentence $\mathbf{x} = \{x_1, x_2, ..., x_L\}$ with $L$ words, our code-switcher module generates the augmented code-switching sentence $\mathbf{x}^a = \{x_1^a, x_2^a, ..., x_L^a\}$ by replacing or not $x_i$ with $x_i^a$ from the predefined bilingual dictionary according to word relevance and difficulty temperature. The label $\mathbf{y}$ is kept the same as the original sentence. With the word relevance score of each word $x_i$ measured in a sentence, the code-switcher gradually

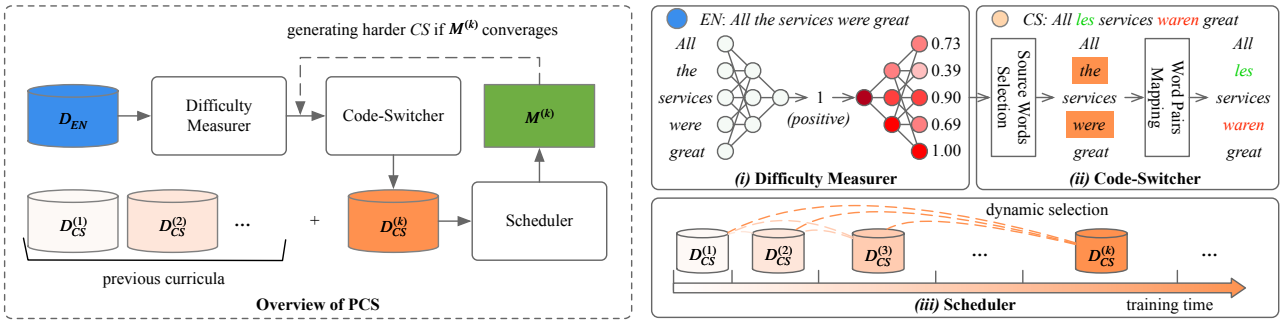



Figure 2: The left subfigure provides an overview of our proposed progressive code-switching framework, while the right subfigure illustrates the three key components. ***(i)*** The difficulty measurer calculates the relevance scores to estimate the contribution of each word in the source language data towards the prediction; ***(ii)*** The code-switcher selects substitution words based on the relevance score to generate suitable code-switching data; ***(iii)*** The scheduler decides when to sample harder code-switching examples for model training. $D_{EN}$: the labelled data in the source language; $D_{CS}^{(k)}$: the generated code-switching data in the $k$-th curriculum; $M^{(k)}$: the learned model for target languages in the $k$-th curriculum.

generates harder code-switching sentences. A pre-trained difficulty measurer with the source language data determines the word relevance. Our goal is to learn a model with source language data $D_S^{train} = \{(\mathbf{x}, \mathbf{y})\}$ and augmented code-switching data $D_{CS}^{train} = \{(\mathbf{x}^a, \mathbf{y})\}$ to perform zero-shot prediction on target languages $D_T^{test} = \{\mathbf{x}\}$.

### 3.1 Difficulty Measurer

The idea of our PCS lies in the strategy of "training from easier code-switching data to harder code-switching data". Due to the code-switching data being augmented from the original data without a predefined difficulty score, we first need to measure what kinds of data are harder than others. Existing popular difficulty measurers in natural language tasks include sentence length [Spitkovsky *et al.*, 2010], word rarity [Platanios *et al.*, 2019], and replacement ratio [Wei *et al.*, 2021a]. However, these measures ignore that replacing the same words produces different degrees of change under different tasks. Therefore, this cannot guarantee the 'easy-to-hard' order, because a code-switching sentence with a few important words being replaced is more difficult than one with a large of irrelevant words being replaced for the model. Here, we argue that code-switching sentences having more important words being replaced have bigger semantic distortion with the original sentences than others, and are treated as more difficult instances. Inspired by explanation methods [Arras *et al.*, 2019], we use Layer-Wise Relevance Propagation (LRP) [Bach *et al.*, 2015] to assign each word a relevance score indicating to which extent it contributed to a particular prediction, as the basis for estimating the difficulty level of CS. In other words, LRP can quantify whether a token is important in the model's decisions to the prediction we are interested in.

We suppose given a trained model $f$, which has learned a scalar-valued prediction function, e.g. $f_c(\mathbf{x})$ means prediction probability of an input sequence $\mathbf{x}$ being class $c$ in the classification task. We adopt a BERT-based LRP [Wu and Ong, 2021] consisting of a standard forward pass, followed by a specific backward pass. For a linear layer of the form as Eq. (1), and given the relevance of the output neurons $r_j$, the relevance of the input neurons $r_i$ are computed through the following Eq. (2), where $\epsilon$ is a stabilizer.

$$z_j^{(l+1)} = \sum_i x_i^{(l)} \cdot w_{ij}^{(l)} + b_j^{(l+1)} \tag{1}$$

$$r_i^{(l)} = \sum_j \frac{x_i^{(l)} \cdot w_{ij}^{(l)}}{z_j^{(l+1)} + \epsilon} \cdot r_j^{(l+1)} \tag{2}$$

In practice, starting from the output neuron whose relevance is set to the value of the prediction function, i.e. $f_c(\mathbf{x})$, LRP uses Eq. (2) to iteratively redistribute the relevance from the last layer $f_c(\mathbf{x})$ down to the input layer $\mathbf{x}$, layer by layer, and verifies a relevance conservation property. We denote $r_{(d)}(x)$ the relevance of the $d$-th dimension ($d \in \{1, 2, ..., D\}$) of the token $x$ and we can derive it as follow Eq. (3):

$$\begin{aligned} r_{(d)}(x) &= f_c(\mathbf{x})\left(\frac{\mathbf{w}^{(l)}x^{(l)}}{\mathbf{z}^{(l+1)}}\right)a'(z_j^{(l+1)})...\left(\frac{\mathbf{w}^{(0)}x^{(0)}}{\mathbf{z}^{(1)}}\right)a'(z_j^{(1)}) \\ &= f_c(\mathbf{x})\left(\prod_l \frac{\mathbf{w}^{(l)}x^{(l)}}{\mathbf{z}^{(l+1)}}\right)\left(\prod_l a'(z_j^{(l+1)})\right) \\ &\approx f_c(\mathbf{x})\left(\prod_l \frac{\mathbf{z}^{(l)}}{\mathbf{z}^{(l+1)}}\right) \end{aligned} \tag{3}$$

where $\mathbf{z}^{(l)}$ is column matrix of hidden states in layer $l$, and derivatives of non-linear activation functions $a'(\cdot)$ are ignored as proposed in [Arras *et al.*, 2019; Wu and Ong, 2021]. For non-linear layers such as the self-attention layer and the residual layer, $z^l$ is approximated by the first term in the Taylor expansion formally as Eq. (4) as proved in [Bach *et al.*, 2015], where $f_\psi$ is an arbitrary differentiable function, and $\hat{x}$ is the Taylor base point where $f_\psi(\hat{x}) = 0$. We derive the relevance score of the token $r(x)$ w.r.t. the class $c$ using absolute sum of $r_{(d)}(x)$, i.e. $r(x) = \sum_d r_{(d)}(x)$.

$$z_j^{(l+1)} = \sum_i f_\psi(x_i^{(l)}) \approx \sum_i \frac{\partial f_\psi(\hat{x}_i^{(l)})}{\partial x_i^{(l)}}(x_i^{(l)} - \hat{x}_i^{(l)}) \tag{4}$$

Via this backward pass, we can observe which words really contributed to the output. An example is shown in Figure 2, "*services*" and "*great*" significantly contribute to the prediction of "*positive*".

### 3.2 Code-Switcher

To generate code-switching sentences that match the difficulty level of the current curriculum, we introduce a code-switcher that incorporates a variable temperature denoted as $\tau$. This temperature parameter represents the proportion of words to be replaced, and it increases linearly as the curriculum stage advances. Given the original source-language sentence and the word relevance score, the code-switcher selects the words in ascending order of word relevance. After that, a target language is chosen randomly based on a bilingual dictionary. It's important to note that source-language words can have multiple translations in the target language. In this case, one of the multiple translations is randomly selected as the target word. While this selection might not guarantee an exact word-to-word translation within the context, we have observed that this scenario is infrequent within most bilingual dictionaries. Consequently, the randomness introduced by this process has a minimal impact on our code-switching.

### 3.3 Scheduler

The scheduler aims to sample the data and send it to the model trainer for training. The scheduler decides when to sample the harder training data with the training progress. For our PCS, we begin with a temperature of $\tau = 0$ equivalent to sampling the source language data. Then, the temperature linearly increases by the increment $\delta$ (e.g. $\delta = 0.1$) every time the validation loss convergence, up to a final temperature of $\tau = 1$. As the temperature increases, we will generate harder code-switching data. To encourage the model to pay more attention to harder data, we set larger early stopping patience for harder curricula than easier ones. However, we found that training the model on a sequence of CS datasets faces the problem of catastrophic forgetting [Kirkpatrick *et al.*, 2017]. As the curriculum stage progresses, code-switching training datasets with varying augmentation levels are sequentially inputted into the model. This results in the modification of weights acquired during the initial curriculum once the model encounters the target of the new curriculum, causing the occurrence of catastrophic forgetting. To mitigate this problem, we design a dynamic curriculum scheduler for the model to revisit previous curricula. Specifically, at the $k$-th curriculum stage, the scheduler selects the code-switching data $D_{CS}^{(i)}$ for the model training on the following probability:

$$P(D_{CS}^{(i)}) = \frac{e^{i-k}}{\sum_{i=1}^{k} e^{i-k}} \tag{5}$$

### 3.4 Model Trainer

The model trainer progressively trains downstream task-specific models with the training data given by the scheduler. And the model has the same network architecture as the pre-trained model in difficulty measurer. We use the conventional fine-tuning method as proposed in the [Devlin *et al.*, 2019]. Specifically, we use a pre-trained multilingual model as an encoder to obtain the representation. Then model $f$ predicts task-specific probability distributions, and we define the loss of cross-lingual fine-tuning as

$$\mathcal{L}_{task} = -\sum_{x} l(f(x), G(x)) \tag{6}$$

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

| Dataset | #Lang. | #Train | #Dev. | #Test | #Labels | Metric |
|---|---|---|---|---|---|---|
| PAWS-X | 7 | 49,401 | 2,000 | 2,000 | 2 | Acc. |
| MLDoc | 8 | 10,000 | 1,000 | 2,000 | 4 | Acc. |
| XTOD | 3 | 30,521 | 4,181 | 2,368 | 12/11 | Acc./F1 |

Table 1: Summary statistics of datasets. Note that XTOD is a joint task dataset that includes 12 intent labels and 11 slot labels.

## 4 Experiments

### 4.1 Setup

**Tasks and Datasets.** To comprehensively evaluate our proposed method, we conduct experiments on three types of cross-lingual transfer tasks with three widely used datasets. (1) For paraphrase identification, we employ **PAWS-X** dataset [Yang *et al.*, 2019] containing seven languages. The label has two possible values: 0 indicates the pair has a different meaning, while 1 indicates the pair is a paraphrase. The evaluation is the classification accuracy (ACC). (2) For document classification, we employ **MLDoc** [Schwenk and Li, 2018] as our document classification dataset, including seven different target languages. The evaluation is the classification accuracy (ACC). (3) For spoken language understanding, we use the cross-lingual task-oriented dialogue dataset (**XTOD**) [Schuster *et al.*, 2019] including English, Spanish, and Thai across three domains. The corpus includes 12 intent types and 11 slot types, and the model has to detect the intent of the user utterance and conduct slot filling for each word of the utterance. The performance of intent detection is evaluated using classification accuracy (ACC), while slot filling can be stated as a sequence labelling task evaluated on the F1 score. The statistics of datasets are summarized in Table 1.

**Implementation Details.** We implement our proposed method based on mBERT and XLM-R-large of HuggingFace Transformer [1] as the backbone model. We set our hyperparameters empirically following previous approaches [Liu *et al.*, 2020; Qin *et al.*, 2021; Lee *et al.*, 2021] with some modifications. We set the batch size to 16 or 64, the maximum sequence length to 128, and the dropout rate to 0.1, and we use AdamW as the optimizer. We select the best learning rate from {5e-6, 1e-5} for the encoder and {1e-3, 1e-5} for the task-specific network layer. As for the scheduler, we initialize $\tau = 0$, which linearly increases as the stage increases. We conduct each experiment 3 times with different random seeds and report the average results of 3 run experiments. To imitate the zero-shot cross-lingual setting, we consider English

[1] https://github.com/huggingface/transformers.

| Model | de | es | fr | ja | ko | zh | Avg. |
|---|---|---|---|---|---|---|---|
| mBERT-based models | | | | | | | |
| mBERT [2019] | 85.7 | 87.4 | 87.0 | 73.0 | 69.6 | 77.0 | 80.0 |
| WS [2021] | 86.7 | 89.8 | 89.4 | 78.9 | 78.1 | 81.7 | 84.1 |
| SCOPA [2021] | 88.7 | 90.3 | 89.7 | **81.5** | 80.1 | 84.3 | 85.8 |
| SALT [2023a] | 87.9 | 89.9 | 89.1 | 78.6 | 77.4 | 81.8 | 84.1 |
| IECC [2023] | 87.9 | 88.9 | 89.3 | 79.4 | 77.9 | 81.8 | 84.2 |
| Macular [2023b] | 88.1 | 90.0 | 89.3 | 80.3 | 79.0 | 83.6 | 85.1 |
| **PCS (Ours)** | **89.5** | **91.4** | **90.9** | 80.8 | **80.4** | **84.6** | **86.3** |
| larger XLM-R-based models | | | | | | | |
| XLM-R [2020] | 89.7 | 90.1 | 90.4 | 78.7 | 79.0 | 82.3 | 85.0 |
| TCS [2023] | 90.8 | 91.6 | 91.4 | 81.8 | 81.7 | 84.7 | 87.0 |
| SCS [2023] | 91.7 | 91.6 | 92.0 | 82.8 | 82.9 | 85.3 | 87.7 |
| IECC [2023] | 92.0 | 92.1 | 92.6 | **83.7** | 84.3 | 85.6 | 88.4 |
| **PCS (Ours)** | **92.4** | **93.0** | **92.8** | 83.6 | **85.1** | **86.6** | **88.9** |

Table 2: Results (Acc.) on natural language inference (PAWS-X). The last '**Avg.**' column denotes the average result for all languages. The best performance is in **bold** (same for Tables 3 and 4).

| Model | de | es | fr | it | ja | ru | zh | Avg. |
|---|---|---|---|---|---|---|---|---|
| mBERT-based models | | | | | | | | |
| mBERT [2019] | 80.2 | 72.6 | 72.6 | 68.9 | 56.5 | 73.7 | 76.9 | 71.6 |
| CoSDA [2021] | 86.3 | 79.2 | 86.7 | 72.6 | 73.7 | 75.1 | 85.5 | 79.9 |
| WS [2021] | 89.1 | 76.7 | 88.1 | 72.0 | 74.4 | 79.0 | 83.0 | 80.3 |
| SCOPA [2021] | 90.7 | 86.1 | **90.5** | 75.1 | 76.7 | **80.4** | 85.5 | 83.6 |
| M-BoE [2022] | 75.5 | 76.9 | 84.0 | 70.0 | 71.1 | 68.9 | 72.2 | 74.1 |
| **PCS (Ours)** | **91.4** | **87.6** | **90.5** | **78.4** | **78.3** | 78.1 | **88.9** | **84.7** |
| larger XLM-R-based models | | | | | | | | |
| XLM-R | 94.9 | 94.5 | 94.7 | **85.6** | 81.9 | 72.0 | 91.8 | 87.9 |
| **PCS (Ours)** | **95.4** | **95.1** | **95.6** | **85.6** | **83.2** | **72.5** | **92.6** | **88.6** |

Table 3: Results (F1) on document classification (MLDoc).

as the source language and others as the target languages. The bilingual dictionaries we use for code-switching are from MUSE [Conneau *et al.*, 2017]. All models are trained on a single Tesla V100 32GB GPU.

## 4.2 Performance Comparison

We compare our method with the following competitive code-switching enhancement models as our baselines:

**mBERT** [Devlin *et al.*, 2019] is a 12-layer transformer model pre-trained on the Wikipedias of 104 languages and is fine-tuned only on the labelled source language training data.

**XLM-R** [Conneau *et al.*, 2020] is a 24-layer transformer-based multilingual masked language model pre-trained on a text in 100 languages around 2.5TB unlabeled text data extracted from CommonCrawl datasets.

**MLT** [Liu *et al.*, 2020] chooses source keywords based on the attention scores computed by a trained source language task-related model to generate code-switching sentences.

| Model | es | th | Avg. |
|---|---|---|---|
| mBERT-based models | | | |
| mBERT [2019] | 73.7/51.7 | 28.2/10.6 | 51.0/31.2 |
| MLT [2020] | 86.5/74.4 | 70.6/28.5 | 78.6/51.5 |
| CoSDA [2021] | 94.8/80.4 | 76.8/37.3 | 85.8/58.9 |
| HCLD [2022] | 84.7/79.5 | **81.0**/32.2 | 82.9/55.9 |
| **PCS (Ours)** | **95.3/81.5** | **81.0/38.5** | **88.2/60.0** |
| larger XLM-R-based models | | | |
| XLM-R | 96.8/85.5 | 95.4/32.8 | 96.1/59.2 |
| **PCS (Ours)** | **98.0/86.6** | **97.0/54.3** | **97.5/70.4** |

Table 4: Results (Intent Acc./Slot F1) on slot filling and intent detection (XTOD).

**CoSDA** [Qin *et al.*, 2021] generates code-switching data with an empirically constant token replacement ratio to enhance the multilingual representations.

**WS** [Lee *et al.*, 2021] simply substitutes words in sentences in every batch during training.

**SCOPA** [Lee *et al.*, 2021] softly mixups the source word embeddings and the switched target word embeddings with an auxiliary pairwise alignment objective.

**M-BoE** [Nishikawa *et al.*, 2022] only mixups the embeddings of Wikipedia entities to boost the performance of cross-lingual text classification.

**HCLD** [Ma *et al.*, 2022] classifies pre-defined intent with code-switching augmentation and then fills the slots under the guidance of intent.

**SALT** [Wang *et al.*, 2023a] incorporates masked language modelling-based offline code-switching and online embedding mixup to enhance the cross-lingual transferability.

**IECC** [Ji *et al.*, 2023] proposes an isotropy enhancement and constrained code-switching method for cross-lingual transfer to alleviate the problem of misalignment.

**TCS** [Lu *et al.*, 2023] encourages cross-lingual interactions via performing token-level code-switched masked language modelling.

**SCS** [Lu *et al.*, 2023] further proposes a semantic-level code-switched masked language modelling based on multiple semantically similar switched tokens in different languages.

**Macular** [Wang *et al.*, 2023b] incorporates code-switching augmentation into multi-task learning to capture the common knowledge across tasks and languages.

As shown in Tables 2, 3 and 4, compared with strong code-switching baselines, PCS shows its superiority and generality across different backbones and tasks at the zero-shot setting. In MLDoc and XTOD, we implement the XLM-R baseline following the reported settings of XLM-R [Conneau *et al.*, 2020]. In Table 2, PCS outperforms SCOPA by 0.5% based on mBERT, and outperforms IECC by 0.5% based on XLM-R. For Table 3, PCS outperforms SCOPA by 1.1% based on mBERT, and outperforms our reproduced XLM-R by 0.7%. In Table 4, compared to random selection (CoSDA) or solely

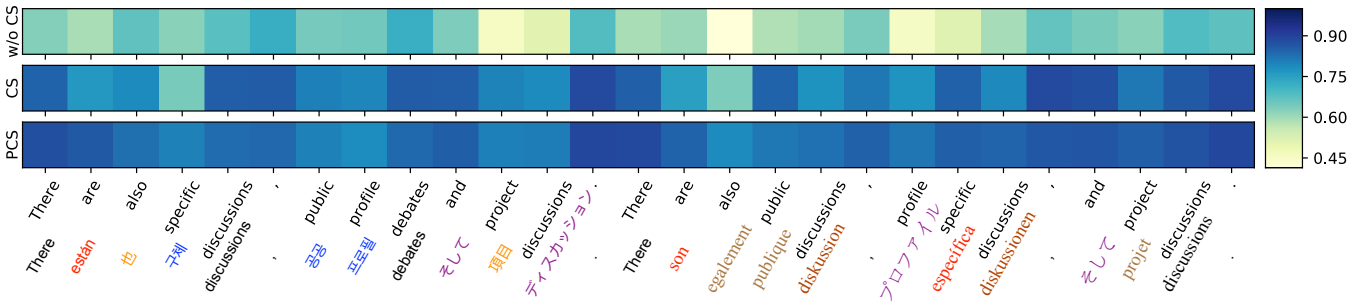



Figure 3: A darker colour indicates a higher cosine similarity score between source words in the original sentence and corresponding target words in the code-switching sentence.

| Model | de | es | fr | ja | ko | zh | Avg. |
|---|---|---|---|---|---|---|---|
| **PCS (Full)** | **89.5** | **91.4** | 90.9 | **80.8** | **80.4** | **84.6** | **86.3** |
| (1) w/o scheduler | 88.7 | 91.2 | 90.5 | 80.1 | 80.1 | 83.5 | 85.7 |
| (2) w/o CL | 87.7 | 90.1 | 89.9 | 79.2 | 78.9 | 82.8 | 84.8 |
| (3) using Ratio-CL | 88.5 | 90.7 | 90.7 | 79.3 | 79.8 | 83.2 | 85.4 |
| (4) using Grad-CL | 89.4 | 91.0 | **91.1** | 80.2 | 80.3 | 84.1 | 86.0 |
| (5) using Anti-CL | 88.1 | 91.0 | 90.5 | 80.5 | 79.6 | 84.4 | 85.7 |
| (6) using TGT-Only | 89.0 | 90.4 | 90.0 | 79.4 | 79.5 | 82.9 | 85.2 |

Table 5: Ablation study (Acc.) on PAWS-X.

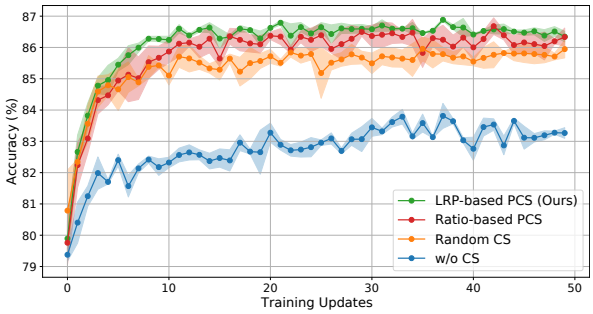



Figure 4: Learning curves of our PCS and three baseline models on PAWS-X based on mBERT.

choosing keywords (MLT) to construct code-switching sentences, our approach demonstrates superior performance.

### 4.3 Ablation Study

To better understand PCS, we conduct ablation studies to analyse the contributions of each component. Table 5 presents the ablation study results for our PCS on PAWS-X. Comparing the full model, we can draw several conclusions: (1) We remove our dynamic scheduler in PCS for the variant w/o scheduler. Results show that our dynamic curriculum selection effectively alleviates the problem of catastrophic forgetting. (2) We remove the curriculum learning strategy for the variant w/o CL, which degrades into the random code-switching model. Results demonstrate that the usage of PCS pushes code-switching by an absolute gain of 1.5% on average. For (3), we use the word replacement ratio as the difficulty measure in PCS. Results show that the performance drops about 0.9% on average because the word replacement ratio cannot flexibly measure the difficulty of a code-switching sentence for different tasks. For (4), as an alternative difficulty measurer in our study, we employ a gradient-based explanation model. Results show that the performance slightly drops about 0.3% on average. This means our LRP-based difficulty measure is superior to the gradient-based method. For (5), we test anti-curriculum learning, which progressively generates code-switching data from hard to easy. Results show that the performance drops about 0.6% on average. This indicates that introducing hard data in the early curriculum negatively impacts learning performance. For (6), we generate code-switching sentences that only mix one target language with the source language. The performance of each language is more degraded than the PCS. This indicates that mixing multiple languages in code-switching can enhance the model's cross-lingual transfer ability.

### 4.4 Case Study

By analyzing the typical cases bettered by PCS, we seek to shed light on the underlying reasons behind its success.

Firstly, PCS can improve the quality of multilingual word representations. Specifically, by leveraging the high-quality word alignment obtained from easier code-switching sentences and their original counterparts, the model gains valuable pivots to comprehend words within over-replaced code-switching sentences. As illustrated in Figure 3, the representations of "specific" and "also" exhibit a relatively low similarity score with their corresponding code-switched words in the first two rows. This is because the model without code-switching (w/o CS) and the code-switching model (CS) simultaneously consider all words within the over-replaced code-switching sentence. On the other hand, our Progressive Code-Switching (PCS) provides a higher similarity score for these word pairs. This is because PCS incorporates other word pairs aligned in the early curricula, allowing it to understand the over-replaced code-switching sentence.

Moreover, PCS helps the model focus on task-relevant keywords, enabling accurate predictions. We calculate the relevance scores of each word for the prediction result, and we notice that PCS makes correct predictions and provides understandable justifications. As illustrated in Table 6, PCS correctly predicts that the sentence pair is semantically different

| Model | Sentence Pair | Prediction |
|---|---|---|
| w/o CS | Un A Khap es un clan o grupo de clanes relacionados, principalmente de los jats del oeste de Uttar Pradesh y del este de Haryana . Un khap es un clan, o grupo de clanes relacionados, principalmente entre los jats del este de Uttar Pradesh y el oeste de Haryana. | same |
| CS | Un A Khap es un clan o grupo de clanes relacionados, principalmente de los jats del oeste de Uttar Pradesh y del este de Haryana. Un khap es un clan, o grupo de clanes relacionados, principalmente entre los jats del este de Uttar Pradesh y el oeste de Haryana. | same |
| PCS | Un A Khap es un clan o grupo de clanes relacionados, principalmente de los jats del oeste de Uttar Pradesh y del este de Haryana. Un khap es un clan, o grupo de clanes relacionados, principalmente entre los jats del este de Uttar Pradesh y el oeste de Haryana. | different |

Table 6: Case study on a Spanish pair having different (the golden label) semantic meaning in paraphrase identification task. The green-highlighted words represent the top five words that contribute the most to the prediction.

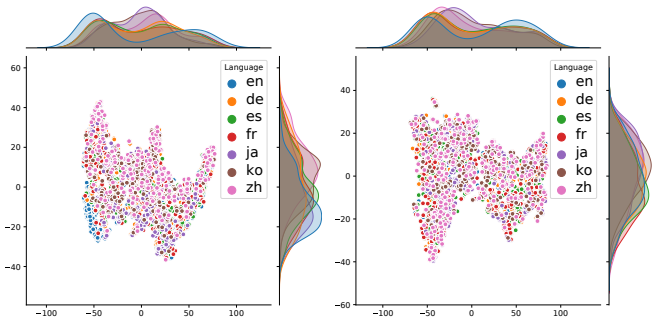
by focusing on the discriminative words (*"este"*, *"Haryana"*, *"oeste"* and *"Haryana"*). In contrast, the model without code-switching (w/o CS) and the code-switching model (CS) fail to do so. We conjecture that our PCS tends to better understand task-related keywords due to its learning process in the later stages of the curriculum, during which the model has already acquired some multilingual knowledge. This further confirms the effectiveness and generalization of the proposed PCS for different tasks.

### 4.5 Learning Curve Analysis

To assess the effectiveness of the progressive code-switching method, we compare the learning curves of our LRP-based PCS with the ratio-based progressive code-switching, the random code-switching and the vanilla fine-tuned mBERT (without code-switching). In Figure 4, we draw the average accuracy and the standard deviation of all target languages based on three experimental runs during the model training updates on PAWS-X. Firstly, all three code-switching models demonstrate significantly improved performance compared to the vanilla mBERT baseline. This indicates that introducing code-switching enhances the model's cross-lingual capabilities. Secondly, both progressive CS models converge to a better solution than the traditional random CS model using the same number of updates. This means that progressive code-switching facilitates the model to converge rapidly to a better minimum. Thirdly, the ratio-based PCS exhibits instability at 15-th and 35-th updates, unlike our LRP-based PCS. This instability arises from the sub-optimal static word replacement rate difficulty measurer, which cannot guarantee that the generated code-switching data satisfies the nature from easy to difficult for various tasks. In contrast, our LRP-based model solves this problem by dynamically generating code-switching data guided by the token-level relevance scores derived from the trained task-related model.

### 4.6 Multilingual Representation Visualization

To examine the alignment of multiple languages, we compare the alignment results of vanilla fine-tuned mBERT and our PCS in terms of sentence representation. We select sentences




(a) mBERT (b) Our PCS

Figure 5: Multilingual alignment t-SNE visualization. Sentence embeddings from fine-tuned mBERT and our PCS.

represented by seven different languages from the PAWS-X datasets respectively to visualize the embedding space. As shown in Figure 5 (a), different languages are distributed in different positions in the embedding space, which indicates that fine-tuned mBERT can distinguish them easily. In contrast, Figure 5 (b) shows that the data distributions of different languages are mixed and overlap each other, which indicates that our model induces language-independent features and boosts the multilingual representation alignment.

## 5 Conclusion

This paper proposes progressive code-switching, which fully mines multilingual knowledge to enhance zero-shot cross-lingual performance. We first adopt a word relevance score calculation method to measure the difficulty of the code-switching data. Then we generate suitable code-switching data controlled by the adoptable temperature. Finally, we introduce a scheduler to decide when to sample harder data for model training. Experimental results on the three zero-shot cross-lingual tasks covering ten languages exhibit the effectiveness and potential of our proposed method.

## Acknowledgements

We thank Beijing Advanced Innovation Center for Big Data and Brain Computing for providing computation resources. We also thank the anonymous reviewers and the area chair for their insightful comments. This research was supported by the Shanxi Province Special Support for Science and Technology Cooperation and Exchange (202204041101020), Key Laboratory of Key Technologies of Major Comprehensive Guarantee of Food Safety for State Market Regulation (BJSJYKFKT202302), and the Zhejiang Provincial Natural Science Foundation of China under Grant (LGG22F020043).

## References

[Arras *et al.*, 2019] Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. Evaluating recurrent neural network explanations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2019.

[Artetxe and Schwenk, 2019] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 2019.

[Bach *et al.*, 2015] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 2015.

[Bengio *et al.*, 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proc. of ICML*, 2009.

[Cao *et al.*, 2020] Steven Cao, Nikita Kitaev, and Dan Klein. Multilingual alignment of contextual word representations, 2020.

[Chi *et al.*, 2021] Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proc. of NAACL*, 2021.

[Conneau and Lample, 2019] Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. In *Proc. of NeuIPS*, 2019.

[Conneau *et al.*, 2017] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data, 2017.

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

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

[Feng *et al.*, 2022] Yukun Feng, Feng Li, and Philipp Koehn. Toward the limitation of code-switching in cross-lingual transfer. In *Proc. of EMNLP*, 2022.

[Goyal *et al.*, 2021] Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. Larger-scale transformers for multilingual masked language modeling. In *Proc. of RepL4NLP*, 2021.

[Graves *et al.*, 2017] Alex Graves, Marc G. Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks, 2017.

[Ji *et al.*, 2023] Yixin Ji, Jikai Wang, Juntao Li, Hai Ye, and Min Zhang. Isotropic representation can improve zero-shot cross-lingual transfer on multilingual language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8104–8118, Singapore, December 2023. Association for Computational Linguistics.

[Kirkpatrick *et al.*, 2017] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017.

[Kumar *et al.*, 2010] M.Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[Lalor and Yu, 2020] John P. Lalor and Hong Yu. Dynamic data selection for curriculum learning via ability estimation. In *Proc. of EMNLP Findings*, 2020.

[Lee *et al.*, 2021] Dohyeon Lee, Jaeseong Lee, Gyewon Lee, Byung-gon Chun, and Seung-won Hwang. Scopa: Soft code-switching and pairwise alignment for zero-shot cross-lingual transfer. In *Proc. of CIKM*, 2021.

[Liu *et al.*, 2020] Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. In *Proc. of AAAI*, 2020.

[Lu *et al.*, 2023] Jinliang Lu, Yu Lu, and Jiajun Zhang. Take a closer look at multilinguality! improve multilingual pre-training using monolingual corpora only. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2891–2907, Singapore, December 2023. Association for Computational Linguistics.

[Ma *et al.*, 2022] Zhanyu Ma, Jian Ye, Xurui Yang, and Jianfeng Liu. HCLD: A hierarchical framework for zero-shot cross-lingual dialogue system. In *Proc. of COLING*, 2022.

[Nishikawa *et al.*, 2022] Sosuke Nishikawa, Ikuya Yamada, Yoshimasa Tsuruoka, and Isao Echizen. A multilingual bag-of-entities model for zero-shot cross-lingual text classification. In Antske Fokkens and Vivek Srikumar, editors, *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 1–12, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[Nooralahzadeh and Sennrich, 2023] Farhad Nooralahzadeh and Rico Sennrich. Improving the cross-lingual generalisation in visual question answering. In *Proc. of AAAI*, 2023.

[Pan *et al.*, 2021] Lin Pan, Chung-Wei Hang, Haode Qi, Abhishek Shah, Saloni Potdar, and Mo Yu. Multilingual BERT post-pretraining alignment. In *Proc. of NAACL*, 2021.

[Platanios *et al.*, 2019] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. Competence-based curriculum learning for neural machine translation. In *Proc. of NAACL*, 2019.

[Qin *et al.*, 2021] Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual nlp. In *Proc. of IJCAI*, 2021.

[Sachan and Xing, 2016] Mrinmaya Sachan and Eric Xing. Easy questions first? a case study on curriculum learning for question answering. In *Proc. of ACL*, 2016.

[Schuster *et al.*, 2019] Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proc. of NAACL*, 2019.

[Schwenk and Li, 2018] Holger Schwenk and Xian Li. A corpus for multilingual document classification in eight languages. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, may 2018. European Language Resources Association (ELRA).

[Shen and Feng, 2020] Lei Shen and Yang Feng. CDL: Curriculum dual learning for emotion-controllable response generation. In *Proc. of ACL*, 2020.

[Spitkovsky *et al.*, 2010] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Proc. of NAACL*, 2010.

[Tay *et al.*, 2019] Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C. Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proc. of ACL*, 2019.

[Wang *et al.*, 2023a] Fei Wang, Kuan-Hao Huang, Kai-Wei Chang, and Muhao Chen. Self-augmentation improves zero-shot cross-lingual transfer, 2023.

[Wang *et al.*, 2023b] Haoyu Wang, Yaqing Wang, Feijie Wu, Hongfei Xue, and Jing Gao. Macular: A multi-task adversarial framework for cross-lingual natural language understanding. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 5061–5070, New York, NY, USA, 2023. Association for Computing Machinery.

[Wei *et al.*, 2021a] Jason Wei, Chengyu Huang, Soroush Vosoughi, Yu Cheng, and Shiqi Xu. Few-shot text classification with triplet networks, data augmentation, and curriculum learning. In *Proc. of NAACL*, 2021.

[Wei *et al.*, 2021b] Xiangpeng Wei, Rongxiang Weng, Yue Hu, Luxi Xing, Heng Yu, and Weihua Luo. On learning universal representations across languages. In *Proc. of ICLR*, 2021.

[Wu and Ong, 2021] Zhengxuan Wu and Desmond C. Ong. On explaining your explanations of bert: An empirical study with sequence classification, 2021.

[Xu *et al.*, 2020] Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. Curriculum learning for natural language understanding. In *Proc. of ACL*, 2020.

[Yang *et al.*, 2019] Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proc. of EMNLP*, 2019.

[Yang *et al.*, 2021] Jian Yang, Yuwei Yin, Shuming Ma, Haoyang Huang, Dongdong Zhang, Zhoujun Li, and Furu Wei. Multilingual agreement for multilingual neural machine translation. In *Proc. of ACL*, 2021.

[Zheng *et al.*, 2021] Bo Zheng, Li Dong, Shaohan Huang, Wenhui Wang, Zewen Chi, Saksham Singhal, Wanxiang Che, Ting Liu, Xia Song, and Furu Wei. Consistency regularization for cross-lingual fine-tuning. In *Proc. of ACL*, 2021.

[Zhu *et al.*, 2021] Qingqing Zhu, Xiuying Chen, Pengfei Wu, JunFei Liu, and Dongyan Zhao. Combining curriculum learning and knowledge distillation for dialogue generation. In *Proc. of EMNLP Findings*, 2021.