PEACH: Pre-Training Sequence-to-Sequence Multilingual Models for Translation with Semi-Supervised Pseudo-Parallel Document Generation

Alireza Salemi[1][*] Amirhossein Abaskohi[1][*] Sara Tavakoli[1],
Yadollah Yaghoobzadeh[1], Azadeh Shakery[1,2]
[1]School of Electrical and Computer Engineering
College of Engineering, University of Tehran, Tehran, Iran
[2]School of Computer Science
Institute for Research in Fundamental Sciences (IPM), Iran
{alireza.salemi,amir.abaskohi,saratavakoli77,y.yaghoobzadeh,shakery}@ut.ac.ir

Abstract

Multilingual pre-training significantly improves many multilingual NLP tasks, including machine translation. Most existing methods are based on some variants of masked language modeling and text-denoising objectives on monolingual data. Multilingual pre-training on monolingual data ignores the availability of parallel data in many language pairs. Also, some other works integrate the available human-generated parallel translation data in their pre-training. This kind of parallel data is definitely helpful, but it is limited even in high-resource language pairs. This paper introduces a novel semi-supervised method, SPDG, that generates high-quality pseudo-parallel data for multilingual pre-training. First, a denoising model is pre-trained on monolingual data to reorder, add, remove, and substitute words, enhancing the pre-training documents' quality. Then, we generate different pseudo-translations for each pre-training document using dictionaries for word-by-word translation and applying the pre-trained denoising model. The resulting pseudo-parallel data is then used to pre-train our multilingual sequence-to-sequence model, PEACH. Our experiments show that PEACH outperforms existing approaches used in training mT5 (Xue et al., 2021) and mBART (Liu et al., 2020) on various translation tasks, including supervised, zero- and few-shot scenarios. Moreover, PEACH's ability to transfer knowledge between similar languages makes it particularly useful for low-resource languages. Our results demonstrate that with high-quality dictionaries for generating accurate pseudo-parallel, PEACH can be valuable for low-resource languages.

1 Introduction

Machine Translation (MT) involves transferring a text from one language to another. Recent investigations have revealed that multilingual pre-training on a large corpus is profitable for NLP

systems' performance on multilingual downstream tasks (Liu et al., 2020; Lample and Conneau, 2019; Conneau et al., 2020; Xue et al., 2021; Devlin et al., 2019) and knowledge transferability between languages (Wu and Dredze, 2019; K et al., 2020; Liu et al., 2020). Furthermore, using parallel data in pre-training encoder and encoder-decoder models effectively increases the models' performance in downstream tasks (Lample and Conneau, 2019; Chi et al., 2021). The existing pre-training approaches are mainly based on Masked Language Modeling (MLM) and its variations (Liu et al., 2020; Raffel et al., 2020; Xue et al., 2021; Lewis et al., 2020).

Although using parallel data in pre-training multilingual models improves their performance on downstream tasks, the amount of available parallel data is limited (Tran et al., 2020). Moreover, MLM-based objectives for sequence-to-sequence (seq2seq) models usually ask the model to generate an output in the same language as input, which is not in the interests of translation tasks. Additionally, MLM-based objectives use shared subwords or alphabets between different languages to learn shared embedding spaces across them (Lample and Conneau, 2019; Lample et al., 2017; Smith et al., 2017); this would not be possible for languages without shared alphabets.

Using dictionaries to define anchor points between different languages in cross-lingual pre-training of the encoder of seq2seq models has been investigated and shown to be effective for unsupervised translation (Duan et al., 2020). Still, it never has been used as a method for pre-training multilingual seq2seq models.

Our proposed method, Semi-Supervised Pseudo-Parallel Document Generation (SPDG), addresses the challenge of limited parallel data for low-resource languages by leveraging dictionaries to generate pseudo-parallel documents. SPDG adopts unsupervised translation techniques (Kim et al., 2018; Lample et al., 2017) to generate a high-

---

[*]equal contribution

quality translation for each pre-training document. We use a pre-trained denoising seq2seq model with word reordering, adding, removing, and substituting to enhance the quality of the word-by-word translated document. The improved unsupervised translated text is used as the target text for training our multilingual seq2seq model, PEACH, using SPDG as a new pre-training method. SPDG enables transfer of knowledge between similar languages, making it particularly useful for low-resource languages.

Our experiments show that PEACH outperforms the pre-trained models with mT5's MLM and mBART's MLM with Reordering objectives in English, French, and German. Additionally, PEACH demonstrates strong performance in zero- and few-shot scenarios. Moreover, we test our model for other multilingual tasks, such as natural language inference, to investigate the model's ability in this task. Our results show that our model achieves a higher score in this task than other objectives, which shows PEACH's ability to transfer knowledge between languages. The main contribution of this paper is twofold:

- We propose a novel semi-supervised pre-training method using bilingual dictionaries and pre-trained denoising models for seq2seq multilingual models.

- We show the benefits of SPDG objective in translation, supervised and zero- and few-shot cases, and knowledge transfer between languages.

2 Related Work

Among the first endeavor for MT, dictionary and rule-based methods were popular (Dolan et al., 1993; Kaji, 1988; Meyers et al., 1998), followed by Knowledge-Based Machine Translation (KBMT) and statistical methods (Mitamara et al., 1993; Carbonell et al., 1981; Koehn, 2009; Al-Onaizan et al., 1999). The popularity of neural machine translation has only grown in the recent decade with the introduction of the first deep neural model for translation (Kalchbrenner and Blunsom, 2013).

While the RNN-based seq2seq models seemed to be promising in neural machine translation (Wu et al., 2016; Bahdanau et al., 2015; Sutskever et al., 2014), the advent of the transformer architecture (Vaswani et al., 2017) plays an integral role in modern MT. With the introduction of

the transformer architecture, pre-training general-purpose language models seemed to be an effective way to improve different NLP tasks (Devlin et al., 2019; Liu et al., 2019). In most cases, transformer models were asked to denoise a noisy input to learn a language (Lewis et al., 2020; Devlin et al., 2019; Raffel et al., 2020). One of the most popular pre-training objectives for both encoder-only and encoder-decoder models is called Masked Language Modeling (MLM), in which the model should predict the masked part of a document and generate it in its output (Raffel et al., 2020). However, many other objectives were also developed for encoder-decoder and encoder-only models (Song et al., 2019; Clark et al., 2020).

Meanwhile, unsupervised methods for neural machine translation (NMT) using monolingual corpora based on adversarial learning (Lample et al., 2017) and transformer-based text denoising (Kim et al., 2018) was tested and demonstrated promising outcomes. Using bilingual dictionaries for defining anchors in pre-training unsupervised translation models was successful (Duan et al., 2020) but never has been used for generating data for supervised translation on a large scale. Our work differs from using dictionaries as anchor points for learning a better representation for tokens in encoder (Duan et al., 2020). We use dictionaries to generate a pseudo translation of the source language in the target language instead of just defining some anchor points. Thus, the model in pre-training steps learns to generate a text in the target language based on input in the source language using only monolingual data and dictionaries on a large scale.

Pre-training task-specific models by generating pseudo-summaries was successful in some cases for summarization (Salemi et al., 2021; Zhang et al., 2020), but it has not been performed for pre-training encoder-decoder seq2seq models for supervised translation according to the best of our knowledge. On the other hand, the endeavors for pre-training specific models for translation ended up in training multilingual language models (Xue et al., 2021; Liu et al., 2020). mT5 (Xue et al., 2021) is trained with the MLM objective of T5 (Raffel et al., 2020). In its pre-training objective, some spans of the input document are masked by specific tokens, and the model has to predict those spans by generating them in its output. mBART (Liu et al., 2020) is another multilingual model based on the BART (Lewis et al., 2020) model,

pre-trained with MLM with Reordering objective. In mBART's pre-training objective, the order of sentences in the input document is corrupted while a specific token masks some spans of the document. The model has to generate the original document in its output.

PEACH is different from both mentioned models because we use a semi-supervised method to generate several pseudo-translations (one for each selected language) of each pre-training document. These translations are then fed to pre-train PEACH. Furthermore, in the mentioned models, the inputs and outputs are from the same language while we ask the model to translate texts from one language to another in our pre-training phase.

3 PEACH

PEACH is a new sequence-to-sequence multilingual transformer model trained with SPDG, a semi-supervised pseudo-parallel document generation method. This section explains the pre-training objective and the model architecture.

3.1 Semi-Supervised Pseudo-Parallel Document Generation (SPDG)

Our proposed pre-training objective, SPDG, generates a pseudo-translation of the input document. For generating pseudo-translations, we use Kim et al. (2018)'s approach for unsupervised translation with some modifications. Our pipeline for pre-training a model based on SPDG is shown in Figure 2. We pre-train a seq2seq denoising model for the target language using the pre-training corpus of that language. Next, for each pre-training document in the source language, we translate it to the target language word-by-word using dictionaries. Then, we give this word-by-word translated document to the pre-trained model with denoising objectives to improve its quality and restore missing words.

Using this method, we can generate the pseudo-translation of each pre-training document from the source language to the target language. We use these pseudo-translations as gold translations for each pre-training document to pre-train a new language model for translation tasks. Since this pre-training objective is similar to translation, we hypothesize that the pre-trained model learns the translation task faster than the models trained using monolingual data.

Word-by-Word Translation Using Dictionaries The first step to generate pseudo-parallel documents is to map sentences from one language to another using dictionaries. We used bilingual dictionaries provided by Conneau et al. (2017) for our work. To map sentences word-by-word from one language to another, we first tokenize sentences using the NLTK[1] library. Then, for each token, we find a translation for the token in the target language using a dictionary from the source to the target language. Some tokens, such as punctuations and numbers, do not need to be translated to the target language because they are shared between them. Therefore, we just put them in the translated words set. Furthermore, we can not find any translation for named entities in dictionaries. To solve this issue, spaCy[2] small (<lang>_core_news_sm) models for named entity recognition for each language are used to extract named entities. We transliterate the named entities and put them in the translated words set. Tokens without translation in dictionaries that are not named entities, punctuations, or numbers are skipped. We hope denoising objectives could find an appropriate substitute for these tokens in the next step. The implementation details of word-by-word translation can be found in Appendix B.

Improving Word-by-Word Translations with Denoising Objectives A critical problem with word-by-word translation is that the word order in the target language is not usually the same as the source. Furthermore, some words in the source language might not have any translation in the target language or vice versa. Additionally, since many words have multiple meanings, word-by-word translation might select the wrong translation for a word.

We define four denoising objectives to overcome the mentioned challenges, and train a denoising model for each language. The pipeline is shown in Figure 1. First, we shuffle the words in each sentence in a document while keeping the relative order of shuffled words in different sentences in the document. Next, we remove, add, and replace some of the words in each sentence to encourage the model to resolve the aforementioned issues in word-by-word translation. We use the corrupted document as the model's input and ask the model to generate the original one as its output.

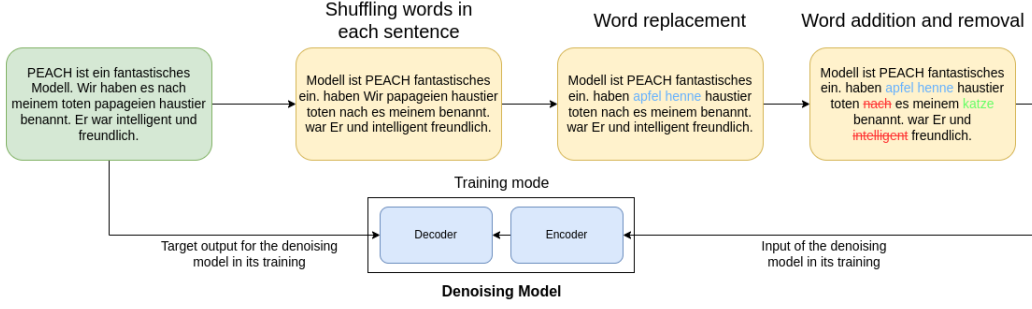The deshuffling objective aims to improve the

Figure 1: An overview of denoising objectives used for training denoising models. We use word shuffling, addition, substitution, and removing based on the values in Table 7 in Appendix C.
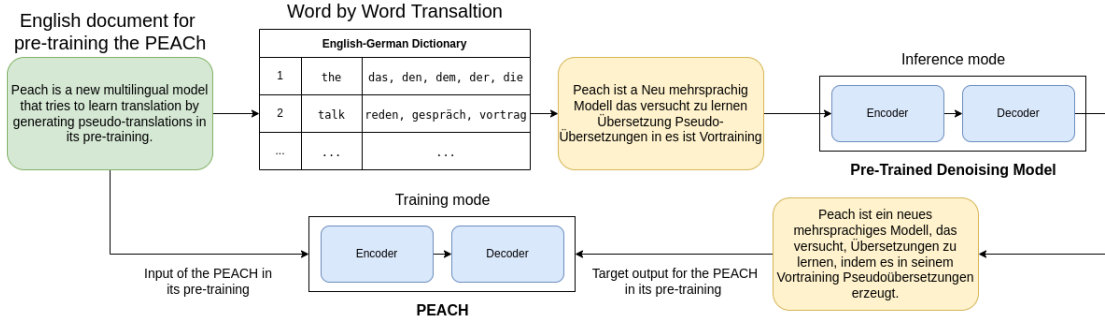

Figure 2: An overview of our pre-training pipeline for training a model based on SPDG. The method uses the output of the word-by-word translation of a pre-training document as the input of the trained denoising model based on Figure 1 to improve its quality.

ability of the model to reorder word-by-word translated documents. Removing and adding words help the model to correct some translations. Moreover, replacing is beneficial especially for correcting the word-by-word translation of ambiguous words.

Figure 2 depicts our pipeline for pre-training with SPDG on a single example. In the mentioned example, after word-by-word translation, some of the words in the pre-training document cannot be translated into German because they do not exist in the dictionary. Furthermore, the relative order of words in the word-by-word translated text is not grammatically correct, and some words can be substituted with more suitable ones. It can be seen that after applying the denoising model to the word-by-word translated text, the mentioned problems are resolved.

3.2 Pre-Training with Multilingual SPDG

Most common multilingual models, such as mT5 (Xue et al., 2021) and mBART (Liu et al., 2020), use MLM and MLM with Reordering as their pre-training objectives. Despite their success, these objectives are not perfectly aligned with the goal of MT. Specifically, these objectives are designed to work on monolingual inputs; they denoise the

input document in a specific language and produce the denoised version in the same language. Here, we design Algorithm 1, in which the pre-training task's input is in one language, and its output is in another language. The algorithms' inputs are the corpora of all languages that the model should be trained on as well as their names. The algorithm generates the input-output pairs for pre-training the multilingual model.

Algorithm 1: Multilingual SPDG

Input : Corpora, Langs
Output : MInputs, Moutputs
   $MInputs := \emptyset$
   $MOutputs := \emptyset$
   for $corpus$ **in** $Corpora$ **do**
      for doc **in** $corpus$ **do**
         for $lang$ **in** $Langs - Lang(doc)$ **do**
            $pst := SPDG(doc, Lang(doc), lang)$
            $MInputs := MInputs \cup \{doc\}$
            $MOutputs := MOutputs \cup \{pst\}$
         end for
      end for
   end for

In Algorithm 1, given a pre-training document, we generate a pseudo-translation of it to each of the other languages. So, the model can observe translations in different languages for a single document.

This helps the model in learning cross-lingual knowledge even about a language not present in a specific training instance. The mentioned claim is because the model learns about the language differences by translating the same input into multiple languages.

It should be noted that based on the goal of pre-training a language model for translation, it is possible to change Algorithm 1. For example, if the multilingual model is going to be used to just translate from or to English, there is no need to pre-train the model with the task of generating pseudo-translation from German to French. Since we are interested in evaluating our model on all pairs of the pre-training languages, we generate pseudo-translation for all pairs in Algorithm 1.

Architecture Our model, PEACH, and the other presented denoising models are all based on transformer (Vaswani et al., 2017) encoder-decoder architecture with a 12 layer encoder and a 12 layer decoder with 768 hidden size, 3072 feed-forward filter size, and 12 self-attention heads.

4 Experiments

In this section, we compare the results of PEACH, trained with SPDG, with other common objectives utilized for pre-training multilingual models. To investigate the effectiveness of SPDG in comparison with common objectives, we pre-trained two other models based on mT5's MLM objective (Xue et al., 2021) and mBART's MLM with Reordering objective (Liu et al., 2020) in the same setup.

The codes for pre-training and fine-tuning of all models are publicly available on GitHub[3].

4.1 Pre-Training Data and Configuration

We pre-train PEACH on English, French, and German with the CC100 corpora (Wenzek et al., 2020; Conneau et al., 2020). Due to the lack of computing power, we cannot use more than around $550M$ words of text from each language. So, we train our model on around $1.6B$ total words. Our pre-training batch size is 96, with a maximum of 512 input and output tokens, and we train it for 500K steps on Google Colab TPUs (v2-8). The AdaFactor (Shazeer and Stern, 2018) optimizer with a decay rate of 0.8 and a dropout rate of 0.1 is used in pre-training and fine-tuning. Furthermore, we use the SentencePiece BPE algorithm (Gage, 1994;

---

[3] https://github.com/AmirAbaskohi/PEACH

Kudo and Richardson, 2018) to generate a vocabulary of 32K words for denoising models and 96k for multilingual models. We pre-train PEACH with Multilingual SPDG for 75% of its pre-training steps and mT5's MLM (Xue et al., 2021) approach for the other 25% pre-training steps. The latter pre-training objective is used because it increases the scope of the fine-tuning tasks that our model can do well. Indeed, multilingual SPDG teaches the model to transform a text from one language to another, but it does not help the model in tasks where their inputs and outputs are in the same language. Therefore, pre-training the model with MLM for a few steps is helpful.

We train the denoising models with the same setup as PEACH. An important factor in training denoising models is the rate of corruption for training documents. We shuffle all words in sentences while removing, adding, and replacing a small proportion of them. We use the word-by-word translation script outputs to decide on these rates. First, we calculate the rate of missing words in word-by-word translation using dictionaries for all languages to a specific language on around 1GB of text of each language. Then, we use a normal distribution with mean and standard deviation of the same as the calculated numbers to define the rate of words that should be removed from a sentence. The values of corruption rates for each language are reported in Table 7 in Appendix C, in which we explain the method to find the best values for rates.

Due to the lack of computing power, we cannot train a large-scale PEACH and compare it with pre-trained models like mT5 or mBART. Instead, we train two models based on mT5 (Xue et al., 2021) objective, which we call MLM, and mBART (Liu et al., 2020) objective, which we call MLM with Reordering, with the same setup as PEACH. Also, we fine-tune a Transformer model with randomly initialized weights on downstream tasks.

4.2 Results

This section evaluates PEACH in various translation scenarios, including supervised, zero- and few-shot. We also evaluate PEACH's ability for cross-lingual knowledge transfer in translation and natural language inference tasks.

Supervised Translation In order to evaluate PEACH on translation tasks, we fine-tune it on the EN-DE and EN-FR parts of the WMT14 dataset (Bojar et al., 2014). Additionally, we fine-tune our

Model	WMT14		WMT19
	FR↔EN	DE↔EN	DE↔FR
MLM	21.38 ↔ 21.64	17.88 ↔ 19.54	16.59 ↔ 16.54
MLM with Reordering	29.02 ↔ 28.71	22.80 ↔ 25.53	21.39 ↔ 22.45
Transformer	9.15 ↔ 9.17	10.02 ↔ 9.79	9.16 ↔ 10.31
PEACH	**31.25 ↔ 29.98**	**23.61 ↔ 26.97**	**23.13 ↔ 25.25**

Table 1: The supervised translation results evaluated with BLEU score.

Model	WMT14		WMT19
	FR↔EN	DE↔EN	DE↔FR
$\text{SPDG}_{EN \leftrightarrow FR}$ (200k steps)	25.98 ↔ 25.42	–	–
$\text{SPDG}_{EN \leftrightarrow DE}$ (200k steps)	–	17.75 ↔ 22.97	–
$\text{SPDG}_{FR \leftrightarrow DE}$ (200k steps)	–	–	16.24 ↔ 18.77
$\text{SPDG}_{EN \leftrightarrow FR \leftrightarrow DE}$ (100k steps)	27.40 ↔ 26.60	21.21 ↔ 23.89	20.49 ↔ 22.32
$\text{SPDG}_{EN \leftrightarrow FR \leftrightarrow DE}$ (200k steps)	**29.04 ↔ 28.08**	**22.33 ↔ 25.29**	**21.67 ↔ 23.29**

Table 2: Results of different models trained with SPDG on either two or three indicated languages. The number of pre-training steps is shown in parenthesis.
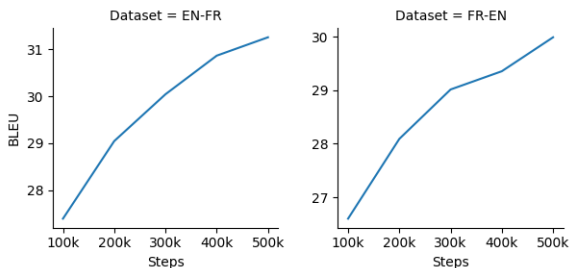

Figure 3: PEACH's performance in pre-training steps on WMT14's EN-FR section. Results for EN-DE and DE-FR are reported in Table 14 in Appendix E.

model on the FR-DE part of the WMT19 dataset (Barrault et al., 2019) in the same setup. Since the test set of WMT19 DE-FR datasets is not available publicly to the best of our knowledge, we evaluated the models on its validation set. The model is fine-tuned for 50K steps with a batch size of 96, a learning rate of 5×10^{-5}, and the same optimizer as pre-training. We use 10K warmup steps for fine-tuning. More information about the experiments' setup is reported in Appendix D. It should be noted that while translation downstream datasets usually have millions of samples, we at most use 50000×96 samples of them due to the lack of computing power. To support the selected number of samples for the downstream task, we report pre-training and fine-tuning time on the whole datasets for an epoch in Appendix D. This sample count is less than 15% of samples for the WMT14

English-French dataset. Additionally, since the primary purpose of this paper is to introduce a new method for pre-training multilingual models and the comparisons happen in the same setup for all objectives, the results are fair and valid.

The results of our model and other trained models on translation tasks are reported in Table 1. Additionally, the results of our model on EN-FR downstream dataset in some pre-training steps are shown in Figure 3. Also, the results for other downstream datasets are reported in Table 14 in Appendix E. The presented results show that PEACH outperforms other models, not only with 500K steps of pre-training but also even with its 200K steps pre-training checkpoint. Furthermore, the MLM method used in mT5 achieves worse results than MLM with Reordering objective that mBART used. We believe this is because the MLM objective of mT5 just asks the model to generate the masked spans in the output, while mBART's objective asks the model to reorder and predict the masked spans of the input document simultaneously. Indeed, the objective of mBART asks the model to generate complete sentences in its output, and that is why it can generate better translations. On the other hand, mT5 just predicts spans of text, which are not complete sentences in many cases.

We believe that the better results of our model stem from its pre-training objective which is similar to translation tasks. Indeed, we pre-trained our

37

model on a massive amount of pre-training data with a task similar to translation, which increases the model's ability in translation when it is fine-tuned with a smaller amount of translation samples.

To investigate the effect of pre-training on more than two languages on the performance of our model on translation tasks, we pre-train a model based on SPDG for 200K steps for each pair of languages, and fine-tune them for 50k steps, with the same setup as PEACH. The results are reported in Table 2. We show that our multilingual model with three languages outperforms other models not only with full pre-training for 200K steps but also with 100K steps of pre-training. We believe this is because we perform the SPDG objective between each pair of languages in its pre-training. Indeed, this approach for pre-training multilingual models helps the model simultaneously gain knowledge about other languages than the pair of languages in each pre-training example because it observes the same input with different outputs for each language. These results support our claim in section 3.2.

Zero- and Few-Shot Translation We evaluate the pre-trained models in a zero-shot setting to investigate our model's ability in low-resource scenarios. Each pre-trained model is evaluated on the test set of WMT14 EN-FR dataset without fine-tuning. The results of this experiment are reported in Figure 4. The results for EN-DE and DE-FR section of WMT14 and WMT19 are reported in Table 15 in Appendix E. The results in Figure 4 and Table 15 show that our model, PEACH, outperforms other models in zero-shot translation. We believe this stems from the similarity of its pre-training objective with actual translation tasks.
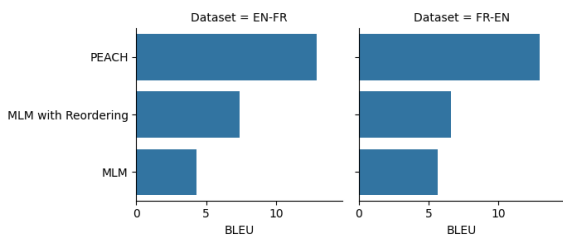

Figure 4: Comparing the pre-trained models in zero-shot setting on WMT14 EN-FR section. Results for EN-DE and DE-FR are reported in Table 15 in Appendix E.

For few-shot experiments, we fine-tuned PEACH on 50K samples from the English-French section of the WMT14 dataset at a maximum of 50K steps. The results are shown in Figure 5.

Accordingly, PEACH outperforms MLM with Reordering model trained in the same setup. Additionally, PEACH surpasses MLM and MLM with Reordeing models' checkpoints in 50K fine-tuning steps on around 5M samples, after only 10K and 25K steps of fine-tuning on 50K samples. We conclude that PEACH performs well in low-resource scenarios because it is trained on a massive amount of psuedo-translation data.
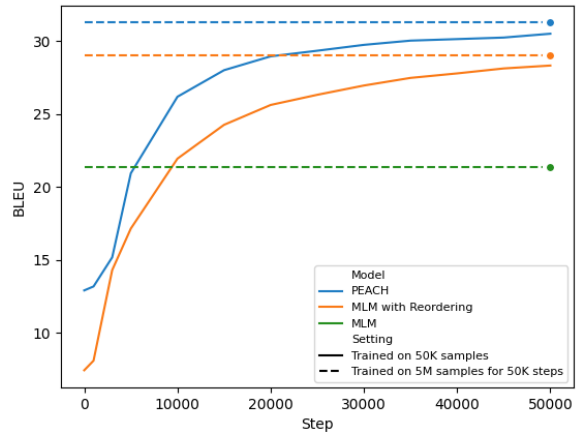

Figure 5: Results of fine-tuning PEACH with 50K samples of WMT14 EN-FR dataset for 0 to 50k steps, and its comparison with MLM and MLM with Reordering objectives on 50000×96 data points. PEACH outperforms the fully-trained MLM models after only 25K fine-tuning steps.

Cross-Lingual Transfer for Translation Here we evaluate each fine-tuned model on a language pair on how it performs for other pairs and directions. We use the fine-tuned models in Table 1 for these experiments.

The experimental results in Table 3 demonstrate that PEACH can transfer the knowledge learned from one language pair to another better than MLM with Reordering model. We believe this stems from our pre-training method in which we ask the model to generate pseudo-translations between each pair of languages. Furthermore, the results confirm Liu et al. (2020)'s experiments and show that whenever a model fine-tuned on a dataset from A to B is evaluated on A to C or C to B or B to A, the results on the evaluation dataset increase more than other combinations. Additionally, because the inputs of PEACH's encoder are human-generated texts while the decoder's expected outputs are the outputs of the denoising models, fine-tuning from A to B increases the performance of C to B more than A to C. Indeed, fine-tuning from A to B helps

Fine-Tuned / Evaluated	PEACH			MLM with Reordering		
	WMT14		WMT19	WMT14		WMT19
	FR↔EN	DE↔EN	DE↔FR	FR↔EN	DE↔EN	DE↔FR
$EN \to FR$	− ↔ 11.38	12.35 ↔ 16.57	12.38 ↔ 21.91	− ↔ 11.25	11.52 ↔ 12.51	11.65 ↔ 11.70
$FR \to EN$	11.30 ↔ −	14.62 ↔ 21.35	15.05 ↔ 17.28	11.27 ↔ −	12.88 ↔ 12.99	12.68 ↔ 11.28
$EN \to DE$	20.63 ↔ 11.99	− ↔ 12.70	19.88 ↔ 13.84	10.80 ↔ 11.29	− ↔ 12.64	12.89 ↔ 11.07
$DE \to EN$	18.97 ↔ 24.54	13.39 ↔ −	14.99 ↔ 18.85	10.99 ↔ 13.85	12.71 ↔ −	11.23 ↔ 11.09
$FR \to DE$	23.64 ↔ 24.69	18.59 ↔ 22.69	− ↔ 23.35	12.07 ↔ 11.43	12.81 ↔ 11.54	− ↔ 20.65
$DE \to FR$	24.88 ↔ 24.94	20.12 ↔ 20.74	23.03 ↔ −	14.72 ↔ 11.57	12.86 ↔ 11.92	21.56 ↔ −

Table 3: The results of experiments on cross-lingual knowledge transfer for translation. We fine-tune the model on one language and evaluate it on other languages. The results are reported using BLEU score.

Model	XNLI		
	EN	FR	DE
MLM	.676	.480	.463
MLM with Reordering	.710	.603	.527
PEACH	**.745**	**.637**	**.636**

Table 4: The accuracy results on the XNLI benchmark.

the decoder of our model learn to generate better outputs by observing human-generated texts in its decoder. This is because our model did not encounter human-generated texts as gold labels in its output during pre-training. On the other hand, observing more human-generated inputs is not as helpful as human-generated outputs since the inputs of the model's encoder were human-generated text in its pre-training.

In support of the previous point, the results in Table 3 show that PEACH fine-tuned on the DE-EN dataset achieves better results than MLM fine-tuned on the FR-EN dataset, when evaluated on the FR-EN dataset. Additionally, PEACH fine-tuned on the EN-FR dataset achieves a comparable result with MLM with Reordering fine-tuned on the DE-FR dataset, when evaluated on the DE-FR dataset (0.54 difference in BLEU). We believe this experiment shows PEACH's ability to transfer the knowledge learned from a language to another effectively.

Cross-Lingual Transfer for natural language inference   We focus on translation in this paper. However, we expect that PEACH's ability to transfer knowledge between languages is suitable for other cross-lingual scenarios as well. To test this hypothesis, we evaluate PEACH on the XNLI benchmark (Conneau et al., 2018). We fine-tune our model for 50K steps with a batch size of 256, a learning rate of 10^{-3}, and a maximum output length of 16 on the MultiNLI English dataset (Williams et al., 2018) and apply it to the XNLI benchmark. The results of this experiment are reported in Table 4.

According to Table 4, PEACH outperforms other models in transferring knowledge from English to German and French. Considering our pre-training objective, in which we ask the model to generate pseudo-translations for each pair of pre-training languages, we believe this objective helps PEACH to transfer the knowledge about the English dataset to other languages better than other pre-trained models.

5 Conclusion

We introduced SPDG, a semi-supervised method for pre-training multilingual seq2seq models, to address the lack of parallel data between different languages. In this new method, we use bilingual dictionaries and denoising models trained with reordering, adding, substituting, and removing words to generate a pseudo-translation for each pre-training document. We use this generated data to train our multilingual model, PEACH, for English, French, and German languages. Our results show that PEACH outperforms the common pre-training objectives for training multilingual models. Furthermore, PEACH shows a remarkable ability in zero- and few-shot translation and knowledge transfer between languages.

Limitations

The main limitations of our work can be classified into two types: 1) SPDG's limitations and 2) Computational limitations.

SPDG's Limitations   Although our method can address the issue of limited parallel data between different languages, it does not solve the problem completely. First, our method uses bilingual dictionaries to translate each pre-training document from one language to another, which is not always available for low-resource languages. Furthermore, the available dictionaries for low-resource languages do not have a high quality and are not comparable

with high-resource languages. Additionally, we use Named Entity Recognition (NER) models to transfer named entities of each pre-training document into its pseudo-translation, which is unavailable for some low-resource languages. Therefore, using unsupervised methods for NER can be a solution for the mentioned problem, which is not investigated in this work.

Computational limitations We did not have access to clusters of GPU or TPU to train our models on a large scale and compare them with the results reported in other papers about multilingual models. However, we tried to provide a realistic setting for our experiments. Further investigation into training models on a larger scale, same as standard multilingual models, can improve this work.

References

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz-Josef Och, David Purdy, Noah A Smith, and David Yarowsky. 1999. Statistical machine translation. In *Final Report, JHU Summer Workshop*, volume 30.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Jaime G. Carbonell, Richard E. Cullingford, and Anatole Gershman. 1981. Steps toward knowledge-based machine translation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3:376–392.

Zewen Chi, Li Dong, Shuming Ma, Shaohan Huang, Saksham Singhal, Xian-Ling Mao, Heyan Huang, Xia Song, and Furu Wei. 2021. mT6: Multilingual pretrained text-to-text transformer with translation pairs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1671–1683, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

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

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

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

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

Bill Dolan, Stephen D. Richardson, and Lucy Vanderwende. 1993. Combining dictionary-based and example-based methods for natural language analysis. In *TMI*.

Xiangyu Duan, Baijun Ji, Hao Jia, Min Tan, Min Zhang, Boxing Chen, Weihua Luo, and Yue Zhang. 2020. Bilingual dictionary based neural machine translation without using parallel sentences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1570–1579, Online. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *8th International*

Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net.

Hiroyuki Kaji. 1988. An efficient execution method for rule-based machine translation. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1700–1709.

Yunsu Kim, Jiahui Geng, and Hermann Ney. 2018. Improving unsupervised word-by-word translation with language model and denoising autoencoder. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 862–868, Brussels, Belgium. Association for Computational Linguistics.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

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

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *NeurIPS*.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

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

Adam Meyers, Michiko Kosaka, and Ralph Grishman. 1998. A multilingual procedure for dictionary-based sentence alignment. In *Proceedings of the Third Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 187–198, Langhorne, PA, USA. Springer.

Teruko Mitamara, Eric H Nyberg 3rd, and Jaime G Carbonell. 1993. Automated corpus analysis and the acquisition of large, multi-lingual knowledge bases for mt. In *Proceedings of the Fifth Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Alireza Salemi, Emad Kebriaei, Ghazal Neisi Minaei, and Azadeh Shakery. 2021. ARMAN: Pre-training with Semantically Selecting and Reordering of Sentences for Persian Abstractive Summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9391–9407, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *ICML*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. *Cross-Lingual Retrieval for Iterative Self-Supervised Training*. Curran Associates Inc., Red Hook, NY, USA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

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)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Y. Wu, M. Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, M. Krikun, Yuan Cao, Q. Gao, Klaus Macherey, J. Klingner, Apurva Shah, M. Johnson, X. Liu, Lukasz Kaiser, Stephan Gouws, Y. Kato, Taku Kudo, H. Kazawa, K. Stevens, George Kurian, Nishant Patil, W. Wang, C. Young, J. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, G. Corrado, Macduff Hughes, and J. Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

A Pre-Training and Downstream Datasets' Information

We use the CC100 corpus (Wenzek et al., 2020; Conneau et al., 2020) for pre-training all models in this work. More specifically, we used the English (EN), French (FR), and German (DE) parts of the mentioned corpus. Due to the lack of computing power and the massive amount of paragraphs in this corpus, we use around 3GB of the text of each language, approximately 550M words from each language and a total of 1.6B words, to pre-train our models. For more reproducibility, we select 500000×96, the total pre-training steps multiplied

by the used batch size, first paragraphs of each mentioned language in the corpus as pre-training samples.

In order to evaluate the models for translation tasks, we use English to French and English to German sections of the WMT14 (Bojar et al., 2014) and the French to German part of the WMT19 dataset (Barrault et al., 2019). The total number of samples in each set of each pre-training dataset is reported in Table 5. We do not use all the samples in all datasets due to the lack of computing power. We use at most 50000×96, the total fine-tuning steps multiplied by the batch size, unique samples of each dataset. We use all the samples for datasets with fewer samples than the mentioned number. To the best of our knowledge, the test set of the WMT19 FR-DE dataset is not publicly available. Therefore, we report the results on the validation set instead of the test set.

For the experiment on transferring knowledge from one language to another, we fine-tune PEACH on the MultiNLI dataset (Williams et al., 2018), consisting of natural language inference samples for the English language. Then, we evaluate the model on English, French, and German samples in test set in the XNLI dataset (Conneau et al., 2018), consisting of natural language inference samples for the mentioned languages. The number of samples for each dataset is reported in Table 6. Both mentioned datasets use three labels; neutral, entailment, and contradiction.

Dataset	Language	Train	Dev	Test
WMT14	EN↔FR	40836715	3000	3003
WMT14	EN↔DE	4508785	3000	3003
WMT19	EN↔FR	9824476	1512	-

Table 5: Number of Samples in supervised stranslation datasets.

Dataset	Language	Train	Test
MultiNLI	EN	392702	-
XNLI	EN	-	5010
XNLI	DE	-	5010
XNLI	FR	-	5010

Table 6: Number of Samples in natural language inference datasets.

B Word-by-Word Translation Implementation Details

The word-by-word translation was performed in batches of 1K documents. The batch size does not affect the algorithm's performance and should be chosen based on the available resources.

After lower casing the documents in a batch, named entities are extracted using the spaCy toolkit. The identified entities should be divided by white space characters since the named entities sometimes consist of multiple words. Since the spaCy toolkit for named entity recognition sometimes chooses definite articles as a part of named entities, we filter out definite articles such as "the," "le," "la," "les," "der," "die," and "das" and translate them using dictionaries in following steps.

In order to perform word-by-word translation, we first tokenize the document. We search for the translation of each token from the source language to the destination language using the appropriate dictionary. If we found more than one possible translation for a token, we uniformly select one of them. Suppose we can not find any translations for a token in the source to the destination language dictionary. In that case, we use source to English and English to destination dictionary to find a translation for the mentioned token. First, we search for a translation from the source language to English using the source to English dictionary. Next, we search for a translation from English to the destination language in the English to the destination dictionary. This technique is just helpful when there is a translation from the source token to English. If we can not find any translations for a token, we mark it as unknown to decide about it later.

For the terms that were marked as unknown, if the token contains numbers or punctuations, we transfer it without any change to the output as a translated word. Otherwise, we check if the word is in the extracted named entities. In this case, we transliterate the word into the destination language using polyglot library [4] and put it in the output as a translated word. For complex words such as "high-end," we break the word into its alphabetical components and search them in the dictionary. If we could find a translation for all components, we would translate each component and concatenate them using the proper separator. In the case that none of the aforementioned scenarios happens, we

---

[4] https://polyglot.readthedocs.io/en/latest/Transliteration.html

omit the word and hope the denoising pre-trained model can find a proper translation for it.

C Denoising Models Pre-Training and Corruption Rate Details

Language	Removing	Addition	Substitution
EN	.066/.061	.01-.03	.05-.07
FR	.152/.087	.01-.03	.05-.07
DE	.137/.085	.01-.03	.05-.07

Table 7: Rates used for pre-training objectives of Denoising models. For removing, we report mean/std.

The procedure for generating pre-training data for training the denoising model is shown in Figure 1. This procedure consists of sentence shuffling, word removing, addition, and substitution.

The first step for generating pre-training data is loading a batch of pre-training documents into the memory as the current batch. We used a batch size of 1K for generating pre-training data for training denoising models. The batch size plays an essential role in this procedure because our algorithm selects candidates for replacing some words in a sample from the words available in other sample in the current batch. We did not investigate the effect of batch size due to the lack of computing power.

After tokenizing the separated sentences using the NLTK toolkit, we shuffle the words in each sentence but keep the relative order of sentences. It helps the denoising model learn the relative order of sentences, which is crucial since the word-by-word translation algorithm might face documents with multiple sentences. Therefore, this will teach the denoising model how to figure out the boundaries of different sentences.

Next, for each sentence, we select $m \times c$ words to be replaced, in which m is the length of the sentence and c is a random number from a uniform distribution between the reported rates in Table 7. The algorithm selects $m \times c$ unique words from other samples in the current batch uniformly to be substituted with the selected words of the current document. The word addition objective works the same way as the substitution, but the algorithm does not replace any words. The word removing objective works the same, but it uses a normal distribution for generating the random number, and it just omits some words from each sentence without replacing them with other words.

The word substitution and addition rates in Table 7 were selected based on observation of the

outputs of the word-by-word translation algorithm. On the other hand, we computed the mean and standard deviation for the proportion of words that the word-by-word translation algorithm could not find any translation for them on the pre-training corpus. The main purpose of the word removing objective is to find a translation for the words that the word-by-word translation algorithm could not find any translation for them by considering the context of the sentence. Therefore, computing this number on the pre-training corpus that the final multilingual model will be trained on will improve the denoising model's ability to denoise the word-by-word translation algorithm's outputs. This decreases the number of words that the word-by-word translation algorithm or the denoising model could not find a translation for them.

D   Experiment Details and Setup

It takes six days to pre-train PEACH on 500000×96 pre-training documents for 500K steps and a batch size of 96. However, the downstream dataset for English-French translation consists of almost 40M samples, which is only 8M less than our pre-training documents and takes five days to fine-tune for just one epoch. Therefore, choosing 50000×96 samples for fine-tuning is plausible due to the number of total pre-training documents and steps of the model's pre-training. The setup for training denoising models is reported in Table 8. The experiment setups for pre-training multilingual models on English, French, and German are reported in Table 9. The setups for pre-training bilingual models used in different experiments are reported in Table 10. Table 11 reports the details of experiments on fine-tuning models on supervised translation tasks. The experiment setup for the few-shot scenario is reported in Table 12. The experiment setup for the fine-tuning on the XNLI (Conneau et al., 2018) task is reported in Table 13.

E   Figures' Details and Information

The reported numbers in Figures 3, 4, and 5 are reported in Tables 14, 15, and 16 for better readability.

Language	Learning rate	Steps	Batch Size	Max Input Length	Max Output Length
English	.01	500K	96	512	512
German	.01	500K	96	512	512
French	.01	500K	96	512	512

Table 8: Pre-training settings for denoising models.

Objective	Learning rate	Steps	Batch Size	Language	Max Input Length	Max Output Length
Multilingual SPDG	.01	500K	96	EN-FR-DE	512	512
MLM with Reordering	.01	500K	96	EN-FR-DE	512	512
MLM	.01	500K	96	EN-FR-DE	512	512

Table 9: Pre-training settings for multilingual models trained on English, German, and French.

Objective	Learning rate	Steps	Batch Size	Language	Max Input Length	Max Output Length
SPDG	.01	200K	96	EN-FR	512	512
SPDG	.01	200K	96	EN-DE	512	512
SPDG	.01	200K	96	DE-FR	512	512

Table 10: Pre-training settings for bilingual models.

Dataset	Learning rate	Steps	Batch Size	Beam Size	Beam alpha	Max Input	Max Output
WMT14 $_{\text{EN-FR}}$	5×10^{-5}	50K	96	1	.6	512	512
WMT14 $_{\text{EN-DE}}$	5×10^{-5}	50K	96	1	.6	512	512
WMT19 $_{\text{DE-FR}}$	5×10^{-5}	50K	96	1	.6	512	512

Table 11: Fine-tuning settings for models used in supervised translation experiments.

Dataset	Learning rate	Sample count	Steps	Batch Size	Beam Size	Beam alpha	Max Input	Max Output
WMT14 $_{\text{EN-FR}}$	5×10^{-5}	50K	1K-50K	96	1	.6	512	512

Table 12: Fine-tuning settings for the few-shot supervised translation experiment.

Dataset	Learning rate	Steps	Batch Size	Beam Size	Beam alpha	Max Input	Max Output	Language
XNLI	1×10^{-3}	50K	256	1	.6	512	16	EN-FR-DE

Table 13: Fine-tuning settings for knowledge transfer experiment on natural language inference.

Dataset // Pre-Training steps	100K steps	200K steps	300K steps	400K steps	500K steps
WMT14 $_{\text{FR} \leftrightarrow \text{EN}}$	27.40 ↔ 26.60	29.04 ↔ 28.08	30.04 ↔ 29.01	30.86 ↔ 29.35	31.25 ↔ 29.98
WMT14 $_{\text{DE} \leftrightarrow \text{EN}}$	21.21 ↔ 23.89	22.33 ↔ 25.29	22.87 ↔ 26.07	23.25 ↔ 26.52	23.61 ↔ 26.97
WMT19 $_{\text{DE} \leftrightarrow \text{FR}}$	20.49 ↔ 22.32	21.67 ↔ 23.29	22.12 ↔ 24.07	22.65 ↔ 24.70	23.13 ↔ 25.25

Table 14: PEACH's performance in different pre-training steps on downstream tasks evaluated with BLEU score. The fine-tuning setup is reported in Table 11. These numbers are reported in Figure 3.

Model	WMT14		WMT19
	FR↔EN	DE↔EN	DE↔FR
MLM	4.33 ↔ 5.64	6.40 ↔ 5.69	6.39 ↔ 4.56
MLM with Reordering	7.42 ↔ 6.63	7.73 ↔ 7.96	7.17 ↔ 7.71
PEACH	12.89 ↔ 12.98	11.75 ↔ 14.05	11.83 ↔ 13.11

Table 15: The zero-shot translation results of the models evaluated with BLEU score. These numbers are reported in Figure 4

Fine-tuning steps	PEACH	MLM	MLM with Reordering	Sample count
0	12.895563	-	7.420614	50K
1K	13.166867	-	8.080425	50K
3K	15.16206	-	14.288106	50K
5K	20.924359	-	17.13509	50K
10K	26.17157	-	21.928285	50K
15K	27.991698	-	24.24655	50K
20K	28.940293	-	25.608678	50K
25K	29.325823	-	26.304938	50K
30K	29.727194	-	26.939679	50K
35K	30.017766	-	27.462619	50K
40K	30.122644	-	27.770637	50K
45K	30.228142	-	28.110951	50K
50K	30.491127	-	28.309444	50K
50K	31.251482	21.384103	29.029701	5M

Table 16: The zero- and few-shot translation results of the models evaluated with BLEU score on EN-FR section of WMT 14. These numbers are reported in Figure 5