Pivot-based Transfer Learning on Pre-trained Sequence-Sequence Models for Low-Resource Neural Machine Translation

Anonymous ACL submission

Abstract

We exploit the cross-lingual capabilities of Self-supervised Multilingual Sequence-to-sequence Pre-trained (SMSP) models for pivot-based transfer learning. Unlike previous research that used a high-resource pivot language, our source-pivot, pivot-target, as well as source-target corpora are extremely small in size (<100k sentences). We show that in this extremely low resource setting, our techniques yield better results compared to those that implement the same pivot-based transfer learning models on a simple Transformer architecture, particularly in zero-shot translation.

1 Introduction

Pivoting is recognized as a promising approach to tackle the problem of low-resource language translation, particularly zero-shot translation (Utiyama and Isahara, 2007). In pivoting, translation of a low-resource language pair (source-target) is decomposed into the problem of using two high-resource independent models: source-pivot and pivot-target. However, when two models are trained independently, time complexity doubles, and errors get propagated from source-pivot translation to the pivot-target translation. Some early Neural Machine Translation (NMT) solutions overcame the error propagation issue by sharing information between source-pivot and pivot-target models (Chen et al., 2017; Zheng et al., 2017; Cheng, 2019).

More recently, pivoting and transfer learning were combined to obtain better results compared to above methods. Kim et al. (2019) trained source-pivot and pivot-target models (parents) separately using a simple Transformer (Vaswani et al., 2017). Then the encoder from the source-pivot model, and the decoder from the pivot-target model were used to initialize source-target model (child). This is termed pivot-based plain transfer learning. They presented another model- step-wise pre-training, where a source→pivot model is first trained with a source-pivot parallel corpus. Then, training is continued with a pivot-target corpus, while freezing the encoder parameters of the previous model.

However, in both these models, the source encoder is trained to be used by the pivot decoder, and the target decoder is trained to use the pivot encoder output. As a solution to this problem, a cross-lingual encoder has been trained using source and pivot data. This models the source and pivot in the same embedding space. In order to learn this cross-lingual encoder, Kim et al. (2019) used a de-noising objective. They trained both the pivot-based plain transfer learning and step-wise pre-training models on top of this pre-trained cross-lingual encoder. Later, Ji et al. (2020) experimented with three different objectives to train the cross-lingual encoder: Masked language Model (MLM), Translation Language Model (TLM) and the BRidge Language Model (BRLM). They trained a pivot-target model on this encoder and tested with source-target data, simulating a zero-shot translation.

Although pivot-based transfer learning has shown very promising results, both Kim et al. (2019) and Ji et al. (2020) carry the assumption made by other pivot-based techniques - the existence of large source-pivot and pivot-target corpora. The cross-lingual encoder can be trained on monolingual data from the pivot and target languages, however Kim et al. (2019) showed that better results are obtained when parallel data is used. However, many low-resource languages do not have sufficient amounts of quality parallel data even with English (Ranathunga et al., 2021). Moreover, pre-training a cross-lingual encoder is an overhead.

In this paper, we show that when the available parallel corpora (source-pivot, pivot-target) are extremely small (<100k sentences), the gains of pivot-based plain transfer implemented on a simple Transformer architecture are negligible compared to a source-target model directly trained with source-target parallel data (direct translation),
while the step-wise pre-training models mostly lag behind direct translation. This observation stays the same even when a cross-lingual encoder is used. Particularly, zero-shot results of most of these models do not go above 1 BLEU.

Thus, we implement both pivot-based transfer learning and step-wise pre-training strategies on a Self-supervised Multilingual Sequence-to-sequence Pre-trained (SMSP) model. This way, source, pivot and target languages would have an implicit shared representation, meaning that this shared representation is not explicitly learnt for the translation task at hand as done in Kim et al. (2019); Ji et al. (2020). Rather, it has been learnt when the SMSP model was pre-trained with monolingual data from multiple languages. In multilingual models, representation of a language gets enriched by related languages in the model (Hu et al., 2020).

We show that, in this extremely low-resource setting, pivot-based plain transfer and step-wise pre-training models implemented on an SMSP model yield a higher result than directly fine-tuning the SMSP model with source-target data. In particular, step-wise pre-training shows very good results for zero-shot translation. This model even comes close to the basic pivoting (known to be a very strong zero-shot baseline (Arivazhagan et al., 2019)) when English is in the target side, thus can be considered as a viable alternative for pivoting that has double inference time.

2 Methodology

Unlike the simple Transformer models (Vaswani et al., 2017) that have to be trained from scratch, SMSP models such as BART (Lewis et al., 2019) and T5 (Raffel et al., 2019), as well as their multilingual counterparts (mBART (Liu et al., 2020; Tang et al., 2020) and mT5 (Xue et al., 2020)) have been already trained with massive amounts of monolingual data for denoising tasks. These multilingual SMSP models can be fine-tuned for NMT tasks (Liu et al., 2020).

In this work, we implement the pivot-based plain transfer learning and pivot-based step-wise pre-training strategies of Kim et al. (2019) on an SMSP model. Then we further pre-train the SMSP model and re-implement the two transfer learning models.

2.1 Pivot-based Plain Transfer Learning

This method (Figure 1) has two steps:

- **Bilingual fine-tuning:** We separately fine-tune the SMSP model with source→pivot and pivot→target parallel corpora. Once fine-tuned, we expect the encoder of the source→pivot model to have considerable knowledge about the source language. Similarly, the decoder of the pivot→target model is expected to have considerable knowledge about the target language.

- **Child model initialization with pre-trained encoder and decoder:** We transfer the encoder parameters of the fine-tuned source→pivot model, and the decoder parameters of the fine-tuned pivot→target model to initialize the child model (source→target). This initialized model is further fine-tuned on the source-target parallel corpus. If no source-target parallel data is available for fine-tuning, we have a zero-shot translation problem.

2.2 Pivot-based Step-wise Pre-training

Unlike in the previous method, in this method the target decoder is trained along with the source encoder before the final fine-tuning (Figure 2).

- **Step 1** - Fine-tune the SMSP model with a source-pivot parallel corpus.
- **Step 2** - Initialize a model by combining the encoder of this fine-tuned SMSP model with the decoder of the SMSP model and further fine-tune with a pivot-target parallel corpus, while freezing the encoder parameters.
- **Step 3** - Continue the fine-tuning with available source-target parallel corpus.

Freezing encoder parameters in the second step avoids catastrophic forgetting of the fine-tuned
source encoder when training with a pivot-target corpus. Although this prevents updating of the encoder parameters during pivot→target training, the encoder is capable of modeling the pivot language, as the SMSP model is pre-trained on the pivot language. Therefore freezing ensures the target decoder is trained with the outputs of the fine-tuned source encoder during pivot→target training. We can even expect strong zero-shot translation from this model after the second fine-tuning step. Although the training of the second fine-tuning step is on pivot→target direction, freezing the encoder helps it retain source language modeling by avoiding full adaptation to the pivot language inputs. Note that dropping the last step results in zero-shot translation of source→target.

2.3 Continuously Pre-training SMSP Model

SMSP models can be extended to support new languages by further pre-training them with monolingual data of those languages (Tang et al., 2020; Liu et al., 2021; Chen et al., 2020; Susanto et al., 2021). In contrast to those studies, we pre-train the SMSP model with monolingual data of the source and pivot languages that are already in the model. The objective of this pre-training is to provide a stronger cross-lingual signal to the encoder for the source and pivot languages, which are used at the decoder. Note that this process is similar to pre-training a cross-lingual encoder as done by Kim et al. (2019) and Ji et al. (2020). The only difference is, we start with an already pre-trained model, rather than from scratch on a Transformer model. For pre-training we used the monolingual data of the source-pivot parallel corpus. It is possible to use a larger monolingual corpus, but we are constrained by computational resources.

Once done, this further pre-trained SMSP model is used to initialize the pivot-based plain transfer learning and step-wise pre-training models.

3 Experimental Setup

3.1 Architecture

For our experiments, we select mBART50 as the SMSP model, because it has shown very promising results for the considered languages (Thillainathan et al., 2021). Implementation details of model fine-tuning are given in the Appendix.

For input representation, we use the SentencePiece model used in mBART50, which contains 250,000 sub-word tokens. Here, Vocabulary substitution or enhancement techniques as those used by Kim et al. (2019) were not needed as the out-of-vocabulary (OVV) rates for Sinhala and Tamil were very low for our dataset (7.88x10-5% for Sinhala and 0% for Tamil).

3.2 Dataset

The experiments are carried out using a multi-way parallel dataset of Sinhala, Tamil and English (Fernando et al., 2020). We tested for all the translation directions - each language was considered as a pivot for translation between the other two languages. Train, validation and test set sizes are 56,603, 1,950 and 1,962 sentences (respectively). Thus, unlike in Kim et al. (2019) and Ji et al. (2020), our parallel corpora are extremely small.

3.3 Baselines

1. Direct source→target: A standard transformer NMT model and a fine-tuned mBART50 model trained on source→target parallel data.

2. Basic pivot translation (Utiiyama and Isahara, 2007); using simple transformer models and fine-tuned mBART50 models.

3. Pivot-based plain transfer learning step-wise pre-training on a simple Transformer model (Kim et al., 2019).

4 Results and Analysis

Fine-tuning results are reported in Table 1 and zero-shot results are reported in Table 2.

Similar to Kim et al. (2019), we notice that the pivot-based plain transfer learning model implemented on the simple Transformer model outperforms the direct source-target translation model. However, the gains are marginal, compared to Kim et al. (2019). Step-wise pre-training always lags behind direct translation. We believe this is due to not having enough source-pivot and pivot-target parallel data to train the models. Adding a cross-lingual encoder marginally improves the results for pivot-based plain transfer model only.

Despite Sinhala and Tamil being under-represented in the mBART50 model, all the solutions implemented on mBART50 model outperform their simple Transformer counterparts, including the non-English-centric Sinhala-Tamil transla-
<table>
<thead>
<tr>
<th>Method</th>
<th>Si→Ta</th>
<th>Ta→Si</th>
<th>Sl→En</th>
<th>En→Sl</th>
<th>Ta→En</th>
<th>En→Ta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>19.0</td>
<td>28.6</td>
<td>31.3</td>
<td>26.7</td>
<td>29.8</td>
<td>17.2</td>
</tr>
<tr>
<td>Basic pivot</td>
<td>18.1</td>
<td>20.7</td>
<td>25.7</td>
<td>21.5</td>
<td>25.7</td>
<td>14.9</td>
</tr>
<tr>
<td>Pivot-based plain transfer</td>
<td>19.4</td>
<td>29.7</td>
<td>31.9</td>
<td>27.3</td>
<td>30.9</td>
<td>17.7</td>
</tr>
<tr>
<td>+ Continuous pre-training</td>
<td>19.6</td>
<td>29.6</td>
<td>32.1</td>
<td>27.7</td>
<td>30.7</td>
<td>18.0</td>
</tr>
<tr>
<td>Step-wise pre-training</td>
<td>18.7</td>
<td>28.3</td>
<td>29.9</td>
<td>26.8</td>
<td>29.3</td>
<td>17.0</td>
</tr>
<tr>
<td>+ Continuous pre-training</td>
<td>18.6</td>
<td>28.3</td>
<td>29.8</td>
<td>27.0</td>
<td>28.1</td>
<td>16.9</td>
</tr>
<tr>
<td>Transformers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mBART50 Fine-tuning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Direct</td>
<td>21.7</td>
<td>32.7</td>
<td>39.1</td>
<td>34.5</td>
<td>36.8</td>
<td>21.2</td>
</tr>
<tr>
<td>Basic pivot</td>
<td>20.1</td>
<td>30.0</td>
<td>33.3</td>
<td>30.3</td>
<td>33.3</td>
<td>19.4</td>
</tr>
<tr>
<td>Pivot-based plain transfer</td>
<td>23.6</td>
<td>34.4</td>
<td>39.3</td>
<td>36.3</td>
<td>37.5</td>
<td>22.5</td>
</tr>
<tr>
<td>+ Continuous pre-training</td>
<td>23.2</td>
<td>34.4</td>
<td>39.2</td>
<td>35.1</td>
<td>37.8</td>
<td>22.2</td>
</tr>
<tr>
<td>Step-wise pre-training</td>
<td>23.0</td>
<td>34.5</td>
<td>39.0</td>
<td>35.8</td>
<td>37.2</td>
<td>22.6</td>
</tr>
<tr>
<td>+ Continuous pre-training</td>
<td>22.8</td>
<td>25.6</td>
<td>39.2</td>
<td>32.9</td>
<td>37.2</td>
<td>22.2</td>
</tr>
</tbody>
</table>

Table 1: Simple Transformer Vs mBART Fine-tuning, reported in SacreBLEU (Post, 2018).

<table>
<thead>
<tr>
<th>Method</th>
<th>Si→Ta</th>
<th>Ta→Si</th>
<th>Sl→En</th>
<th>En→Sl</th>
<th>Ta→En</th>
<th>En→Ta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic pivot</td>
<td>18.1</td>
<td>20.7</td>
<td>25.7</td>
<td>21.5</td>
<td>25.7</td>
<td>14.9</td>
</tr>
<tr>
<td>Pivot-based plain transfer</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>+ Continuous pre-training</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Step-wise pre-training</td>
<td>1.7</td>
<td>1.3</td>
<td>2.4</td>
<td>1.1</td>
<td>2.3</td>
<td>1.4</td>
</tr>
<tr>
<td>+ Continuous pre-training</td>
<td>1.5</td>
<td>0.9</td>
<td>1.7</td>
<td>0.9</td>
<td>2.2</td>
<td>0.8</td>
</tr>
<tr>
<td>Transformers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mBART50 Fine-tuning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic pivot</td>
<td>20.1</td>
<td>30.0</td>
<td>33.3</td>
<td>30.3</td>
<td>33.3</td>
<td>19.4</td>
</tr>
<tr>
<td>Pivot-based plain transfer</td>
<td>0.1</td>
<td>0.1</td>
<td>0.3</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>+ Continuous pre-training</td>
<td>0.1</td>
<td>0.1</td>
<td>0.4</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Step-wise pre-training</td>
<td>12.7</td>
<td>19.6</td>
<td>26.9</td>
<td>13.7</td>
<td>27.7</td>
<td>10.5</td>
</tr>
<tr>
<td>+ Continuous pre-training</td>
<td>12.6</td>
<td>14.7</td>
<td>28.7</td>
<td>16.7</td>
<td>28.4</td>
<td>10.7</td>
</tr>
</tbody>
</table>

Table 2: Zero-shot results, reported in SacreBLEU (Post, 2018).

Pivot-based plain transfer model always outperforms the direct translation model. Step-wise pre-training model lags behind the direct translation model only for Sinhala→English. Interestingly, the results gained over the direct translation is the lowest when English is in the target side. We believe that this is due to the mBART model already being pre-trained with a much larger corpus of English, and having Sinhala or Tamil (languages relatively under-represented in mBART) as a pivot does not make much difference.

Continuously pre-training mBART with the monolingual version of the parallel data does not yield much gains for both models. In fact, in some translation directions, we observe a slight drop in the result. This could be due to the insufficient amount of data used in pre-training. (Kim et al., 2019) also stated that the cross-lingual encoder needs a larger dataset to be fine-tuned for another decoder. The benefit of using an SMSP model is even more pronounced for zero-shot translation; while the simple Transformer-based models fail miserably on all the zero-shot translation scenarios (with most of the BLEU scores near zero), zero-shot translation with step-wise pre-training on the mBART50 model yields very good results. In particular, when English is in the target side, this zero-shot result comes close to the basic pivoting result. This is a useful observation - because in a zero-shot setting, we can achieve a meaningful result without having to rely on basic pivoting that has double-inference time. Continuous pre-training positively impacts the zero-shot translation of the step-wise pre-training model, when English is in target side.

5 Conclusion

In this paper, we showed that pivot-based transfer learning strategies implemented on an SMSP model yield promising results over direct fine-tuning, as well as for zero-shot translation. Our solutions fit for a scenario where parallel corpora available among source-pivot-target languages are extremely small. Current experiments only considered source/pivot/target languages that are already included in the SMSP model. In future, we will investigate how our solutions can be extended to languages not included in the SMSP model.
References


A Appendix

A.1 Model Fine-tuning

For mBART50 (Tang et al., 2020), we used standard sequence-to-sequence Transformer based architecture (Vaswani et al., 2017), with 12 layers of encoder-decoder with the model dimension of 1024 on 16 heads.

For fine-tuning mBART50, batch size was set to maximum of 1024 tokens, checkpoint saving interval is set to 5000 updates and we use a maximum of up to 100k training updates similar to Thillainathan et al. (2021). For all directions, we fine-tuned with 0.3 dropout, 0.2 label smoothing, 2500 warm-up updates, 3e-5 maximum learning rate as described by Liu et al. (2020). Each model was optimized using Adam optimizer (Kingma and Ba, 2017) and Inverse Square Root Scheduler was used as the learning rate scheduler. Fairseq\(^3\) toolkit was used for model training and transfer implementations. It was taken approximately 24-28 hours for a single fine-tuning of mBART50 on NVIDIA Quadro RTX 6000 GPU.

For pre-training and fine-tuning of simple transformers, batch size was set to 32 tokens, update frequency is set to 4 and checkpoint saving interval is set to 10 epochs. We used 0.4 dropout, 0.2 label smoothing, 4000 warm-up updates, 1e-3 maximum learning rate. Adam optimizer (Kingma and Ba, 2017) and Inverse Square Root Scheduler were used as the optimizer and the learning rate scheduler (respectively) for the simple transformer experiments. All the pre-training and fine-tuning processes of simple transformers were trained up to 500 maximum epochs. It was taken approximately 4-5 hours for a single pre-training/fine-tuning of simple transformers on NVIDIA Quadro RTX 6000 GPU.

A.2 Continual Pre-training of the SMSP model

We utilize the parallel data as monolingual data for each selected language pair. We use the same nosing techniques as used by mBART model (Liu et al., 2020) which is Random Span masking and Order Permutation. We mask 0.3% of words in each instance (with random masking 0.1) by randomly sampling a span length according to a Poisson distribution ($\lambda = 3.5$). We also permute the order of sentences within each instance. The decoder input is the original text with one position offset. A language id symbol <LID> is used as the initial token to predict the sentence.

\(^3\)https://github.com/pytorch/fairseq