Combining Sequence Distillation and Transfer Learning for Efficient Low-Resource Neural Machine Translation Models

Raj Dabre **Atsushi Fujita**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
firstname.lastname@nict.go.jp

Abstract

In neural machine translation (NMT), sequence distillation (SD) through creation of distilled corpora leads to efficient (compact and fast) models. However, its effectiveness in extremely low-resource (ELR) settings has not been well-studied. On the other hand, transfer learning (TL) by leveraging larger helping corpora greatly improves translation quality in general. This paper investigates a combination of SD and TL for training efficient NMT models for ELR settings, where we utilize TL with helping corpora twice: once for distilling the ELR corpora and then during compact model training. We experimented with two ELR settings: Vietnamese–English and Hindi–English from the Asian Language Treebank dataset with 18k training sentence pairs. Using the compact models with 40% smaller parameters trained on the distilled ELR corpora, greedy search achieved 3.6 BLEU points improvement in average while reducing 40% of decoding time. We also confirmed that using both the distilled ELR and helping corpora in the second round of TL further improves translation quality. Our work highlights the importance of stage-wise application of SD and TL for efficient NMT modeling for ELR settings.

1 Introduction

Neural machine translation (NMT) (Bahdanau et al., 2015; Sutskever et al., 2014) enables end-to-end training of translation models and is known to give state-of-the-art results for a large variety of language pairs. NMT models with large hidden sizes or deep stacked layers tend to give better translations than those with small hidden sizes or fewer layers. Large models inevitably need more storage space and computation, and are difficult to deploy on low-computation and low-memory devices. Additionally, beam search decoding is known to improve translation quality but needs more computation and is unacceptable in a low-latency real-time application where faster decoding is as valuable as if not more valuable than translation quality. Consequently, neural models that are compact and fast are extremely important and a growing body of research known as neural model efficiency focuses on this issue.

One of the most popular techniques to train efficient models is knowledge distillation (Hinton et al., 2015) which relies on transferring the knowledge learned by a large model (called teacher) into a smaller model (called student). Sequence distillation (SD) (Kim and Rush, 2016) is a special case of knowledge distillation for sequence-to-sequence models, such as those used for NMT. Not only does it help in the training of compact and fast models with high translation quality, it sometimes helps in eliminating the need for beam search which further increases decoding speed. SD relies on the creation of distilled parallel corpora by translating the training source sentences into the target language by using a large model. The distilled corpora are simplified representations of how the large model sees the original corpora and their quality will have a direct impact on the translation quality of compact models trained with them.

While SD is known to perform extremely well for high-resource settings, its direct application to extremely low-resource (ELR) settings will not work due to over-fitting. Table 1 gives the BLEU scores (Papineni et al., 2002) for Vietnamese–English (Vi–En) and Hindi–English (Hi–En) translation tasks in the Asian Languages Treebank (ALT) (Riza et al., 2016),[1] where Transformer Base models (Vaswani et al., 2017) with 1, 2, 3, and 6 encoder and decoder layers were trained on the ALT training data of 18k sentence pairs. It is clear that there is a huge performance gap between the

[1] http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/ALT-Parallel-Corpus-20191206.zip

Proceedings of the 5th Conference on Machine Translation (WMT), pages 492–502
Online, November 19–20, 2020. ©2020 Association for Computational Linguistics

#Layer	Vi→En		En→Vi		Hi→En		En→Hi	
	G	B	G	B	G	B	G	B
1	14.6	15.4	19.0	20.5	9.7	10.1	10.8	11.7
2	16.4	17.6	21.1	22.7	10.9	12.1	12.3	13.5
3	16.4	18.1	22.1	23.7	11.6	12.7	12.6	13.8
6	19.4	20.5	24.0	25.2	14.2	15.2	15.0	16.3

Table 1: The impact of number of encoder-decoder layers on translation quality: BLEU scores of greedy (G) and beam search (B, beam size of 4).

6-layer models and shallower ones. Thus, distilled corpora generated using shallower models will certainly hurt the translation quality of compact models. However, when we used the 6-layer models to generate distilled corpora, the translations were almost identical to the reference translations (with almost 100 BLEU). The reason is over-fitting despite the use of classic regularization methods such as dropout. Consequently, we need to rely on helping corpora through transfer learning (TL) methods. TL can be used by itself to improve the performance in ELR settings regardless of model efficiency. However, very little is known about how SD and TL work together.

In this paper, we investigate how to train efficient (compact and fast) NMT models for ELR settings with helping corpora through domain adaptation or cross-lingual TL. We use TL twice: once for distilling ELR corpora and then for training efficient models. We expect that training multi-domain or cross-lingual models by simply concatenating, without oversampling, the ELR corpora with the helping corpora leads to NMT models that can help generate useful distilled corpora.

To evaluate the effect of our proposed method, we experimented with two ELR language pairs, Vietnamese–English and Hindi–English (4 translation directions), in the ALT dataset. When we trained compact NMT models with 40% fewer parameters only on the distilled ELR corpus, the resulting models showed improved translation quality with greedy search by 3.6 BLEU points in average over the models trained on the original ELR corpus, while reducing 40% of decoding time. Furthermore, when we jointly used the distilled ELR corpora with the helping corpora via TL, the quality of the resulting compact models was further improved by up to 3.7 BLEU points over the best score achieved by using no distilled data. This highlights the importance of stage-wise application of SD and TL for efficient NMT models in ELR settings with high translation quality. Although the individual techniques utilized in this work are not novel, their combination and our empirical observations pertaining to the development of efficient models for ELR settings are novel.

The contributions of our paper are as follows:

- An empirical study of the combination of TL methods and SD for efficient NMT modeling.
- A cost-benefit analysis of efficient models for ELR settings.

2 Related Work

Our work is at the intersection of knowledge distillation (Hinton et al., 2015) and transfer learning for training compact NMT models.

2.1 Sequence Distillation

Knowledge distillation for sequence-to-sequence models have been successful in training efficient (compact and fast) NMT models. Sequence distillation (SD) (Kim and Rush, 2016) for NMT is a simple approach which involves training a large NMT model on a parallel corpus, translating the source side of the corpus, and then using the pseudo-parallel corpus of the same source side and the generated pseudo-target, called distilled corpus, to train a compact NMT model. The pseudo-targets represent the large model's interpretation of the original targets and can be considered as smoothed label sequences. The sequences are simpler and hence easier for smaller models to learn. As our focus is on a simple and efficient solution for ELR settings, we decided to focus only on SD.

However, its impact on ELR settings is uncertain. Given that only few thousands of domain-specific sentences are available, training large NMT models tends to over-fit on the small corpora while compact NMT models will only lead to pseudo-targets of poor quality, both preventing the generation of useful distilled corpora. It is certainly possible to search for an optimal model size. However, it will involve a time-consuming hyper-parameter search, while the result may be specific to given corpora.

2.2 Transfer Learning

Transfer learning (TL) can be in the form of domain adaptation (Chu et al., 2017) or cross-lingual or multilingual transfer (Firat et al., 2016; Zoph et al., 2016; Dabre et al., 2019; Johnson et al., 2017; Dabre et al., 2020) using helping bilingual corpora.

Assume that L_1–L_2 is an ELR language pair and L_3–L_4 is a helping pair. The given parallel corpora for the two pairs may belong to different domains. Typically, pre-training a model on the larger L_3–L_4 corpus and then **fine-tuning** (**"ft"**) it on the smaller L_1–L_2 corpus is known to give the best translation quality for the L_1–L_2 pair (Zoph et al., 2016; Chu et al., 2017; Dabre et al., 2019), regardless of the number of model parameters. However, without careful regularization, this will definitely lead to the L_1–L_2 corpus being memorized. To address this, joint training of an NMT model using the following two methods on both corpora has been studied:

Mixed Training ("mxt"): Directly train on the concatenated corpus.

Mixed Fine-Tuning ("mxft"): First train on the L_3–L_4 corpus as in "ft," but perform fine-tuning on the concatenated corpus.

Prior to concatenating two corpora, the L_1–L_2 corpus is typically oversampled so that its size matches to the L_3–L_4 corpus. Also, we can prepend the source sentences with two artificial tokens, one indicating the domain of the corpus (Chu et al., 2017), and another indicating the target language into which we want to translate (Johnson et al., 2017). Note that when L_2 and L_4 are the same, the target language tokens are unnecessary. If L_1 and L_3 are also the same, then we are essentially performing domain adaptation.

2.3 Other Related Work

Some recent work tackled efficient NMT modeling in low-resource settings (Goyal et al., 2020; Gordon and Duh, 2020). Whereas they focus on applications of TL for compact models as this paper, there are some key differences between them and ours. Gordon and Duh (2020) focus on low-resource settings, but our low-resource data are significantly smaller than theirs. Second, whereas they use distillation twice and TL once, we recommend distillation once and TL twice. Finally, they do not examine cross-lingual TL for model compression. Goyal et al. (2020) focus on cross-lingual learning, but their approaches are centered more on leveraging orthographic or linguistic similarity, whereas we make no efforts towards orthographic unification. We thus consider parts of these studies to be orthogonal to ours.

Apart from domain adaptation and cross-lingual TL methods, low-resource settings can benefit from monolingual data, for instance, through back-translation (Sennrich et al., 2016), where target language monolingual data are translated into pseudo-source sentences. Recently, pre-training on monolingual data (Devlin et al., 2019; Song et al., 2019; Mao et al., 2020) has been proven to significantly improve the translation quality of ELR settings. Approaches involving helping monolingual data are usually more time-consuming than those that use helping bilingual corpora. Furthermore, given that our approach already needs a reasonable amount of time due to the application of TL and forward-translation of the source sentences of the parallel corpora for distilling them, we consider that such approaches should be used when no more gains can be obtained from helping bilingual corpora. We refer interested readers to work on distillation using unsupervised methods (Sun et al., 2020).

Independent of the application of TL, there exist methods for speeding up NMT, such as weight pruning (See et al., 2016) where model weights close to zero are pruned out, quantization (Lin et al., 2016) where weights are represented by faster to process integers instead of floating point numbers, aggressive model binarization (Courbariaux et al., 2017), and binary code prediction softmax (Oda et al., 2017) where the softmax is sped up by making it predict a binary code representing words instead of one-hot vectors. We expect these methods to further speed up the models obtained using our proposed method.

3 Our Approach: Transfer-Generate-Transfer

Refer to Figures 1 and 2 for a visual overview of our approaches. Figure 1 depicts the application of TL to generate distilled corpora for the ELR settings. Figure 2 depicts how the distilled ELR corpora can be used with the distilled or non-distilled helping corpora to train compact models. Our method for training compact NMT models for ELR settings can be summarized as follows:

1. Train a large joint NMT model using "**mxt**" or "**mxft**" on the concatenation of L_1–L_2 and L_3–L_4 corpora without oversampling L_1–L_2.
2. Use the joint NMT model to decode L_1 into pseudo-L_2 (L'_2) and to decode L_3 into pseudo-L_4 (L'_4).[2]

[2] Instead, a unidirectional $L_3 \rightarrow L_4$ model can be used to distill the L_3–L_4 corpus, because NMT models trained on the larger corpus will prevent from over-fitting and thereby generate reliable distilled data for this pair.

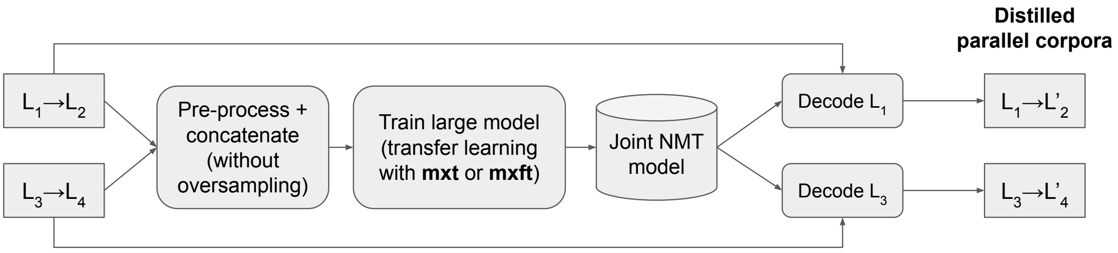
Distilled parallel corpora
$L_1 \rightarrow L_2$
$L_3 \rightarrow L_4$
Pre-process + concatenate (without oversampling)
Train large model (transfer learning with **mxt** or **mxft**)
Joint NMT model
Decode L_1
Decode L_3
$L_1 \rightarrow L'_2$
$L_3 \rightarrow L'_4$

Figure 1: First round of transfer learning: training a joint model to distill the parallel corpus for extremely low-resource language pair (L_1–L_2) by leveraging a helping parallel corpus (L_3–L_4).

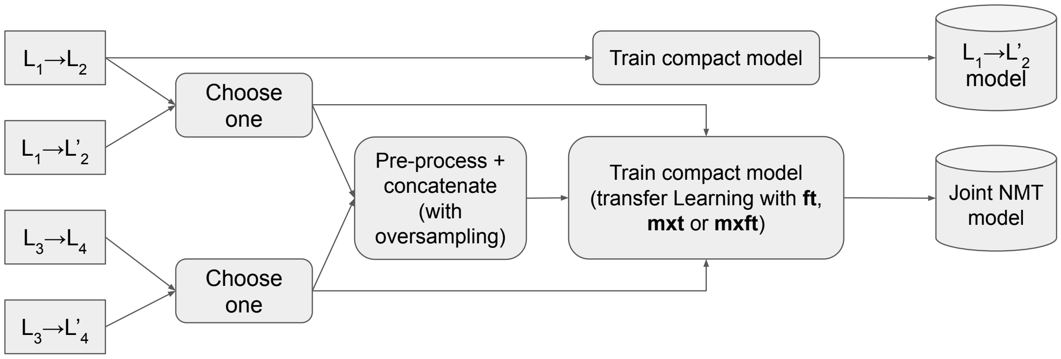
$L_1 \rightarrow L_2$
$L_1 \rightarrow L'_2$
$L_3 \rightarrow L_4$
$L_3 \rightarrow L'_4$
Choose one
Choose one
Train compact model
$L_1 \rightarrow L'_2$ model
Pre-process + concatenate (with oversampling)
Train compact model (transfer Learning with **ft**, **mxt** or **mxft**)
Joint NMT model

Figure 2: Second round of transfer learning: training an efficient NMT model for extremely low-resource language pairs (L_1–L_2) by leveraging a helping parallel corpus (L_3–L_4), using data distilled via the method in Figure 1. There are four possible ways of combining low-resource and helping corpora as each of them can be either distilled or non-distilled.

3. Train compact NMT models only on the L_1–L'_2 corpus, or together with the distilled or non-distilled helping corpora using "**ft**," "**mxt**," or "**mxft**."

Standard TL takes place when both the ELR and helping corpora are non-distilled. In this case, TL is not used to distill data, and the ELR corpus should be oversampled to match the size of the helping corpus to ensure the best translation quality. However, for the purposes of distillation, unlike previous work, we do not oversample the L_1–L_2 corpus before concatenating it with the L_3–L_4 corpus. We did so because our preliminary explorations revealed that oversampling causes the model to memorize the L_1–L_2 corpus, thereby preventing the generation of useful distilled corpora. Naturally, the lack of oversampling might negatively impact on the quality of distilled L_1–L'_2 corpus. One can empirically determine an optimal oversampling rate, but we decided to not search for it in order to make our method simple. We address this point in Section 5.1.3 with empirical evidence justifying our choice.

Note that one can pre-train compact NMT models on helping corpora and then fine-tune them on ELR corpora, avoiding SD altogether. However, the quality of TL is proportional to the quality of the pre-trained model, which tends to be high when using larger models. Furthermore, distilled data is prone to be simpler than the original data and thus has higher potential for leading to compact models. We hypothesize that distilling ELR corpora might help in better model compression. We test this hypothesis through experiments.

4 Experimental Settings

To determine the feasibility of the proposed method, we trained and evaluated NMT models in the following two groups of settings.

#1. With only distilled ELR corpora: To determine the impact of different TL settings on the quality of distilled ELR corpora and hence the compact models trained.

#2. With ELR and helping corpora: To determine the settings using both ELR and helping corpora that give compact models with highest possible translation quality.

4.1 Datasets

We experimented with the ELR Vietnamese–English (Vi–En) and Hindi–English (Hi–E) pairs from the Asian Languages Treebank (ALT) with 18,088 training, 1,000 development, and 1,018 test sentence pairs. As for the helping corpora, we used the training part of the IWSLT 2015 Vietnamese–English[3] and the IITB Hindi–English (Kunchukuttan et al., 2018),[4] consisting of 133k and 1.5M lines, respectively. We chose large as well as small helping corpora in order to determine the impact of helping corpora sizes on the model training.

4.2 Implementation Details

We used the Transformer model for our experiments (Vaswani et al., 2017) because it gives the state-of-the-art results for NMT. We made necessary changes to the code in the tensor2tensor v1.14 implementation of the Transformer in order to construct joint sub-word vocabularies as well as to handle oversampling. Tensor2tensor has its own default sub-word vocabulary learning method which we use as is by feeding it the surface word vocabulary list obtained from combining the ALT language pair and the helping language pair vocabularies. We used the default hyper-parameter setting[5] corresponding to "*transformer_base_single_gpu*" and separate source and target sub-word vocabularies of size 8,000. We chose small vocabularies as they are known to give better results for ELR settings by eliminating vocabulary sparsity. Small vocabularies also lead to models with smaller and faster softmax layers which is crucial for model compactness and speed.

We trained our models, evaluating them on the development set BLEU score every 1,000 iterations, and terminated training after 500,000 iterations or when the BLEU score did not change by more than 0.1 BLEU points for 10,000 iterations.

After training, we averaged the final 10 checkpoints to yield a single model for decoding. For decoding the test sets for evaluation, we compared greedy search and beam search with a beam size of 4, using a length penalty (alpha) of 0.6. On the other hand, for decoding the source sentences of the training sets for distillation, we only used beam search with the same beam size.

4.3 Models Evaluated

Our primary goal is to reduce the decoding time while achieving better translation quality than baselines. Following Kim and Rush (2016), who have shown that the number of encoder-decoder layers (L) have a significantly larger impact on decoding speed than hidden sizes (H), we mostly focus on compact models that use fewer encoder-decoder layers. Nevertheless, we also examine smaller hidden sizes in some experiments.

We trained simple baseline models from scratch with 1, 2, 3, and 6 layers only on the ALT training data (see Table 1).

4.3.1 Models for Distilling Corpora

To train joint models for each translation direction that is later used for distilling training data, we disjointly used the helping Vi→En, En→Vi, Hi→En, or En→Hi corpora. As we used separate source and target vocabularies and hence embedding layers, settings with a helping corpus for different translation direction can be a reasonable simulation of cross-lingual TL settings.

For joint training, we compared "**mxft**" and "**mxt**." We also considered the impact of using the domain indicator tokens (Chu et al., 2017). Thus, for each ELR and helping corpora combination, there were four types of joint models, and thus four different versions of distilled data.

4.3.2 Compact NMT Models for ELR Settings

We trained two types of models, ones that use only the distilled ELR corpora and ones that use the ELR as well as helping corpora.

#1. With only distilled ELR corpora: For each of the four helping corpora per translation direction that are used to distill data, we trained models with $L \in \{1, 2, 3\}$ and $H = 512$.[6] Additionally, we trained 3-layer models with $H \in \{128, 256\}$ to further study the tradeoff between model size and translation quality.

#2. With ELR and helping corpora: For each combination of translation direction and helping direction, we first determined the best distilled ELR corpus among four variants on

[3] https://github.com/stefan-it/nmt-en-vi

[4] http://www.cfilt.iitb.ac.in/iitb_parallel/

[5] The important hyper-parameters that remained constant throughout our experiments are: dropout of 0.1, ADAM optimizer with an initial learning rate of 0.1, 16,000 warm-up steps followed by decay for the learning rate, 8 attention heads, and a batch-size of 1,024.

[6] Feed-forward layer filter sizes were always 4 times the model's hidden size throughout this paper.

Model	Vi→En (VE)				En→Vi (EV)				Hi→En (HE)				En→Hi (EH)			
	VE	EV	HE	EH	VE	EV	HE	EH	VE	EV	HE	EH	VE	EV	HE	EH
$L=1$ $H=512$	19.7	18.9	17.0	12.9	24.9	25.4	20.9	21.1	12.9	12.1	13.8	8.5	14.5	14.2	12.2	14.6
$L=2$ $H=512$	21.9	20.8	18.6	13.6	26.7	28.0	22.2	22.7	14.9	14.2	16.3	9.3	16.7	16.2	13.4	16.9
$L=3$ $H=512$	**23.2**	22.0	18.9	14.1	28.5	**29.2**	22.7	23.1	15.7	15.0	**16.7**	9.6	17.7	16.4	13.9	**18.0**
$L=3$ $H=256$	21.1	20.1	17.7	13.5	26.6	27.3	22.1	22.4	14.1	13.5	15.7	8.7	15.4	15.6	13.5	16.2
$L=3$ $H=128$	19.4	18.2	16.7	12.5	24.5	25.4	21.3	21.1	12.2	11.4	13.7	8.4	12.8	13.6	12.2	14.8
TT	mxt	mxt	mxft	mxt	mxft	mxft	mxft	mxft	mxt	mxt	mxt	mxft	mxft	mxt	mxft	mxt
DT	yes	yes	yes	no	yes	yes	yes	no	yes	no	yes	no	yes	no	yes	yes
$L=6$ $H=512$ (see Table 1)	greedy: 19.4 beam: 20.5				greedy: 24.0 beam: 25.2				greedy: 14.2 beam: 15.2				greedy: 15.0 beam: 16.3			

Table 2: BLEU scores for each ELR translation task achieved by our proposed method with greedy search. The second row indicates the translation direction of helping data. The highest scores for each translation direction are highlighted in bold. "TT" and "DT" respectively represent the type of training ("mxt" or "mxft") and whether domain tags were used ("yes" or "no") for the joint training that led to the best distilled corpora. The last row shows the greedy and beam search BLEU scores of the baseline 6-layer models for comparison (see Table 1).

the basis of BLEU score of the $L = 3$ and $H = 512$ model trained only on it (#1), and then combined it with the distilled helping corpus to train models with $L \in \{1, 2, 3\}$ and $H = 512$ using "**ft**," "**mxt**," and "**mxft**." We also trained models with the same configurations for combinations of ELR and helping corpora where only one of the corpora are distilled. As (strong) baselines, we trained models with $L \in \{1, 2, 3, 6\}$ and $H = 512$ trained on non-distilled ELR and helping corpora.

5 Results

We show results using only distilled ELR corpora and then using it with helping corpora.

5.1 Using Only Distilled ELR Corpora

In Table 2, we show how domain adaptation and cross-lingual TL methods affect creation of distilled ELR corpora and hence the greedy search translation quality of efficient models. Greedy search is emphasized due to our focus on fast decoding speed as well as high translation quality.

5.1.1 Translation Quality of Efficient Models

For each translation direction, the best distilled corpora used to train models with 3 layers gives greedy search translation quality ranging from 1.5 to 4.0 BLEU points over the 6-layer non-distilled baseline model's beam search translation quality. Comparing the 1-, 2-, and 3-layer models trained with the best distilled corpora with their non-distilled counterparts in Table 1, we can see that there is an improvement of 2.9 to 5.5 BLEU points. Considering that we used the distilled equivalents of the original training data, this result shows the explicit effect of TL and SD which helps generate data that improves translation quality despite reducing the model size.

Training models on ELR corpora can finish quickly. Thus, our distilled corpora can be used in situations where quick deployment of compact and fast NMT models is important.

5.1.2 Domain Adaptation vs. Cross-Lingual Transfer

Our experiment revealed that cross-lingual training is definitely a viable alternative. For instance, in Vi→En translation, the best BLEU score was achieved when the helping direction was also Vi→En. When the helping direction was Hi→En, these improvements were much smaller. Nevertheless, it is clear that cross-lingual training is useful when domain adaptation is not possible. Work on script mapping to improve the quality of TL (Song et al., 2020; Goyal et al., 2020) indicates that our cross-lingual distillation procedure might give better results if we mapped Hi to Vi or vice-versa. We leave this for future work.

Consider two hypothetical settings for Vi→En translation, where we used the reversed, En→Vi and En→Hi, helping directions to generate distilled corpora for Vi→En translation. When using En→Vi as the helping direction, the BLEU scores of greedy search with 1-, 2-, and 3-layer models improved by 4.3, 4.4, and 5.6 BLEU points, respectively. These improvements are approximately 1.0 BLEU points lower than those obtained in the domain adaptation setting with Vi→En as the helping direction, but it shows that using helping corpora with different languages can be of some use. However, when using En→Hi as the helping direction, the BLEU scores dropped. Note that English

DT	TT	Vi→En	En→Vi	Hi→En	En→Hi
yes	mxft	21.5	**29.2**	15.3	15.4
yes	mxt	**23.2**	28.0	**16.7**	**18.0**
no	mxft	21.9	29.1	14.2	12.9
no	mxt	21.1	28.0	16.2	16.1

Table 3: Impact of domain tags (DT) and training type (TT) on the greedy search translation quality (BLEU) of models with $L = 3$ and $H = 512$. The best scores are in bold.

Size	Vi→En		En→Vi		Hi→En		En→Hi	
	HE	EH	HE	EH	HE	EH	HE	EH
133k	20.5	19.2	26.4	25.9	15.2	14.6	16.4	17.1
200k	21.3	**20.2**	27.1	28.0	16.3	15.2	16.4	17.1
500k	**21.5**	19.9	27.1	**28.2**	**17.2**	**15.8**	**17.5**	17.5
1500k	18.9	14.1	**22.7**	23.1	16.7	9.6	13.9	**18.0**

Table 4: Impact of helping corpus size on the greedy search translation quality (BLEU) for each translation task achieved with models with $L = 3$ and $H = 512$. The best scores are in bold.

and Vietnamese use the Roman alphabet which might enable cognate sharing even when the ELR and helping directions are opposite. However, this is not fully applicable when En→Hi is the helping direction. Furthermore, the Hindi–English corpus was much larger than the one for Vietnamese–English. Since we do not oversample the ELR corpora for distilling corpora, we expect that the model heavily focuses on the Hindi–English pair which could negatively impact on the quality of the resulting distilled corpora.

While similar observations are applicable to other translation directions, consider Hi→En and En→Hi translation. As before, using Hi→En and En→Hi helping directions respectively using domain adaptation resulted in the best distilled corpora. However, using the reverse En→Hi and Hi→En helping directions, respectively, led to a drop in translation quality. In contrast, using Vi→En and En→Vi helping directions led to distilled corpora that led to compact models giving translations within 1.0 BLEU points of those given by the best distilled corpora. This shows that in a cross-lingual TL setting for distilling ELR corpora, it may be better to have helping corpora that are not much larger than the ELR corpora. We validate this hypothesis in Section 5.1.3.

As for the use of domain indicator tags, 11 out 16 cases indicate that such tags are useful. In Table 3, we show the results of model with $L = 3$ and $H = 512$ trained on distilled data generated with and without domain indicator tags when training using "mxt" and "mxft" (4 combinations). For simplicity, we show results for when the ELR and helping directions are the same. Using domain tags gives better results when the helping corpora are substantially larger than the ELR corpora. But when the helping corpora are relatively smaller (Vietnamese–English), domain tags do not seem to have a large impact. Furthermore, "mxt" tends to be better than "mxft." Overall, simply concatenating the ELR and helping corpora without oversampling or domain indicators and then training joint model in one stage should be sufficient to yield useful distilled corpora. We will experiment with additional language pairs and domains in the future to conclusively determine a one-fits-all setting.

5.1.3 Impact of Helping Corpora Size

We observed that a large helping corpus degrades the translation quality in cross-lingual settings. Instead of determining an optimal oversampling ratio for the ELR corpus, we experimented with downsampling the helping corpus size. We did this to avoid running into the risk of over-fitting due to oversampling. We experimented with the downsampled versions of the Hindi–English corpus: we prepared sub-corpora with 500k, 200k, and 133k sentence pairs, assuring that a larger one subsumes all the smaller ones. For simplicity, we reused the best configurations reported in Table 2.

Table 4 shows the greedy search results. When using the entire Hindi–English helping corpus for Vi→En and En→Vi translation tasks, the BLEU score is substantially lower than the baseline models, indicating the poor quality of the distilled data. Note that we do not oversample the ELR corpora for distillation and thus coupling them with a larger helping corpus is detrimental to the final translation quality, as the NMT model sees more examples in the latter than the former. However, using significantly smaller corpora ensures that the NMT model sees much fewer examples in the helping corpus and thus is able to better learn from the ELR corpus leading to better distilled data. This is evidenced by the improved BLEU scores when using downsampled helping corpora. Naturally, using the Vi→En helping corpus gives the best results for Vi→En translation tasks, but the results using the downsampled Hindi–English helping corpora are within 2.0 BLEU points of the best. Note also that the BLEU score for Hi→En task using a helping corpus with 500k sentence pairs (17.2) surpasses the

Model		Size	Time	BLEU
$L = 1$	$H = 512$	19.0M	11.7s	18.4
$L = 2$	$H = 512$	27.0M	17.6s	20.8
$L = 3$	$H = 512$	34.0M	22.5s	21.8
$L = 3$	$H = 256$	11.0M	22.0s	18.8
$L = 3$	$H = 128$	4.0M	21.3s	17.3
$L = 6$	$H = 512$	56.6M	37.6s	18.2
(see Table 1)				

Table 5: Comparison of size, decoding time (with greedy search), and BLEU score for various models evaluated in Table 2. For each column, average value for four translation directions is reported.

score obtained using all the sentence pairs (16.7) by 0.5 BLEU points. For the reverse direction, the score (17.5) is within 0.5 BLEU points of the best score (18.0). It is clear that choosing an appropriate helping corpus size is important for generating useful distilled corpora. This result further reinforces our claim that cross-lingual training is a viable option for generating useful distilled data. Such cross-lingual training also has the potential to distill data that can help train compact models with BLEU score higher than larger models trained on non-distilled data. As for optimal size of helping corpus, the performance gap between using 200k and 500k helping sentence pairs is very small in most settings. This means that distilling data does not need too much helping corpus and thus in practice choosing a small sample of the helping corpus can help significantly save time for model training and subsequent corpus distillation. This also helps avoid the issue of oversampling and thereby maintaining the simplicity of the method.

A fair comparison with the same size (133k) of helping corpora confirmed that sharing at least one of source and target languages tends to improve the final translation quality in cross-lingual TL settings. For instance, Hi→En has a better impact than En→Hi on Vi→En. Similarly, En→Hi leads to higher BLEU score than Hi→En for En→Vi.

5.1.4 Size vs. Speed vs. Translation Quality

Table 5 compares size, decoding time, and BLEU score for various models. As the model size drops with fewer layers and smaller hidden sizes, BLEU score also drops. However, the decoding time decreases significantly. Note that reducing the number of layers mainly impacts on the decoding time, whereas reducing hidden sizes does not have such a huge impact, as reported in Kim and Rush (2016).

We observed that the model with $L = 3$ and $H = 512$ are approximately 1.7 times (or 40%) smaller and 1.7 times (40%) faster than the 6-layer models despite exhibiting improved translation quality of 3.6 BLEU points in average. If one wishes to save decoding time, we suggest to train a model with $L = 1$ and $H = 512$, which is approximately 3.0 times smaller and 3.2 times faster than a 6-layer model, while having comparable translation quality. If the priority is reducing model size, then using models with $L = 3$ and $H \in \{256, 128\}$ are 5.1 times to 14.2 times smaller, even though they do not benefit much from narrowing down H. The model with $L = 3$ and $H = 256$ is comparable to the one with $L = 1$ and $H = 512$ in terms of quality, but the latter is 1.7 times smaller than the former. We recommend experimenting with different model sizes before choosing the best one for the target application.

5.2 Using Both ELR and Helping Corpora

Table 6 gives the BLEU scores achieved by models trained on both ELR and helping corpora, where we compare the distilled ("Y") and non-distilled ("N") versions of corpora as well as the three types of training ("ft," "mxt," and "mxft").

5.2.1 Importance of Transfer Learning for Efficient Models

Comparing the results of using only ELR corpora against the results of TL without SD, TL already gives 1-layer models that are competitive, if not better than the 3-layer models trained on non-distilled ELR corpora and the 6-layer models trained on distilled ELR corpora. The 1-layer models are 1.9 and 3.2 times faster as well as approximately 1.8 and 3.0 times smaller than the 3- and 6-layer models, respectively (see Table 5). It is thus reasonable to avoid SD altogether when time is of the essence.

Among the training methods, "mxft" was in most cases slightly better than "ft" and both of them are substantially better than "mxt." This highlights the importance of stage-wise TL rather than innocently training on a combination of all corpora. Note that "mxt" achieved the highest BLEU score for some configurations, and it should be a reasonable option when there is not enough time for stage-wise training.

5.2.2 Importance of Distillation with Transfer Learning for NMT Efficiency

Using at least one distilled corpus, either ELR or helping corpora, is important in improving the translation quality of compact models. For instance,

			Vi→En								Hi→En							
ELR	HD	TT	$L = 1$		$L = 2$		$L = 3$		$L = 6$		$L = 1$		$L = 2$		$L = 3$		$L = 6$	
			G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B
N	-	-	14.6	15.4	16.4	17.6	16.4	18.1	19.4	20.5	9.7	10.1	10.9	12.1	11.6	12.7	14.2	15.2
Y	-	-	19.7	19.9	21.9	22.5	23.2	23.1	-	-	13.8	14.4	16.3	16.9	16.7	17.4	-	-
N	N	ft	21.5	22.8	24.6	25.9	25.6	26.8	26.6	27.5	19.5	20.7	25.0	26.1	26.4	27.2	28.1	29.0
		mxt	20.3	22.1	23.3	25.0	24.5	26.0	26.1	27.5	15.0	15.9	18.1	19.9	20.9	22.3	23.3	23.7
		mxft	21.4	22.9	26.2	27.3	26.7	28.0	**27.7**	**28.6**	19.9	20.9	25.4	26.3	27.8	28.7	**29.3**	**29.8**
N	Y	ft	21.9	23.4	25.0	26.3	26.1	27.1	-	-	20.9	21.9	25.9	**27.5**	27.8	28.8	-	-
		mxt	20.9	22.8	24.0	25.4	24.4	25.9	-	-	16.9	18.1	20.6	21.7	22.4	23.1	-	-
		mxft	22.3	24.3	26.4	27.7	26.4	27.6	-	-	21.5	**22.2**	**26.8**	**27.5**	**28.1**	**29.0**	-	-
Y	N	ft	24.1	24.3	26.7	26.9	27.3	27.7	-	-	20.5	20.2	24.7	24.6	25.9	25.9	-	-
		mxt	24.3	25.1	26.8	27.1	27.5	28.1	-	-	18.2	19.2	22.9	23.7	24.0	24.6	-	-
		mxft	24.3	25.1	27.5	**28.3**	27.6	28.0	-	-	20.3	20.9	25.0	24.5	25.9	24.5	-	-
Y	Y	ft	25.1	25.7	27.0	27.7	27.9	28.4	-	-	21.5	21.7	26.0	26.3	26.5	26.7	-	-
		mxt	24.8	25.5	**27.8**	28.1	**28.2**	28.8	-	-	19.8	20.1	24.2	24.7	25.4	25.6	-	-
		mxft	**25.2**	**25.8**	27.6	28.2	28.1	**28.9**	-	-	**21.6**	21.9	26.4	25.8	26.9	25.8	-	-

Table 6: BLEU scores for Vi→En and Hi→En translation tasks with greedy (G) and beam search (B). Models trained on either distilled ("Y") or non-distilled ("N") version of ELR and helping corpora ("ELR" and "HD" columns, respectively) using different domain adaptation techniques ("TT" column), are compared. The highest score(s) in each column are marked in bold.

the BLEU score of greedy search with the 1-layer models trained on some distilled data are up to 3.7 BLEU points higher than the best scores achieved by 1-layer models that do not use distilled data at all (21.5 and 19.9 for Vi→En and Hi→En by "N–N" models in Table 6, respectively). Although the gap between the performances tends to be narrower when the number of layers increases, this sacrifices compactness and decoding speed.

The behavior of models trained on distilled data differs depending on the combination of ELR and helping corpora. For Vi→En, distilling the ELR corpus ("Y–N") is more useful than distilling the helping corpus ("N–Y"). In contrast, for Hi→En, distilling the helping corpus ("N–Y") matters more. Recall that the Vi→En helping corpus is around 10 times smaller than the Hi→En helping corpus. This means that a compact model has to bear the burden of learning a much larger amount of knowledge from the larger helping corpus. Consequently, the compact model should be better at learning the Vi→En helping corpus, especially in its distilled form. Furthermore, given that the distilled ELR corpus for Vi→En already improves translation quality compared to its non-distilled counterpart, it should also help improve translation quality when used it in combination with the helping corpus. This is indicated by the best result for Vi→En achieved by distilling both the ELR and helping corpora. For this direction, the 2-layer models trained on distilled data are either competitive with if not better than the 6-layer models. For Hi→En, given that the size of helping corpus is significantly larger, distilling it into compact models is harder due to lack of parameters. This is the most likely reason behind the relatively small improvement by distilled data. Although the impact of SD on TL on Hi→En is not as impressive as for Vi→En, we advise experimenting with SD rather than not.

6 Conclusion

In this paper, we have explored the combination of transfer learning (TL) and sequence distillation for obtaining compact and fast models in extremely low-resource (ELR) settings. Our experiments on four translation directions revealed that leveraging helping corpora help in distilling ELR corpora that help train compact models with 3.6 average BLEU points improvement in translation quality. Compact models trained on distilled ELR corpora are not only fast but also give better translations than larger models trained on non-distilled ELR corpora. We showed the effects of choosing appropriate training methods, using domain indicator tags, and managing corpora sizes on translation quality. Our cost-benefit analysis of model size, decoding speed, and translation quality showed that we can achieve translation quality comparable to baselines trained on the original ELR corpora with models that are approximately 3.0 times smaller and 3.2 times faster than said baselines. We also showed that combining distilled ELR corpora with the distilled or non-distilled helping corpora, using simple TL methods, can further boost the performance of compact and hence fast NMT models. We strongly recommend to leverage distilled ELR corpora through stage-wise TL for compact and high-quality NMT for ELR settings.

In our future work, we will extend our approach for a single compact multilingual NMT model, for instance, focusing on multi-parallel ALT dataset.

Acknowledgments

A part of this work was conducted under the commissioned research program "Research and Development of Advanced Multilingual Translation Technology" in the "R&D Project for Information and Communications Technology (JPMI00316)" of the Ministry of Internal Affairs and Communications (MIC), Japan. Atsushi Fujita was partly supported by JSPS KAKENHI Grant Number 19H05660.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, USA. International Conference on Learning Representations.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 385–391, Vancouver, Canada. Association for Computational Linguistics.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2017. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. *CoRR*, abs/1602.02830.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A comprehensive survey of multilingual neural machine translation.

Raj Dabre, Atsushi Fujita, and Chenhui Chu. 2019. Exploiting multilingualism through multistage fine-tuning for low-resource neural machine translation. 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 1410–1416, Hong Kong, China. 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, USA. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, USA. Association for Computational Linguistics.

Mitchell Gordon and Kevin Duh. 2020. Distill, adapt, distill: Training small, in-domain models for neural machine translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 110–118, Online. Association for Computational Linguistics.

Vikrant Goyal, Sourav Kumar, and Dipti Misra Sharma. 2020. Efficient neural machine translation for low-resource languages via exploiting related languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 162–168, Online. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Melvin Johnson, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernand a Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, USA. Association for Computational Linguistics.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Darryl D. Lin, Sachin S. Talathi, and V. Sreekanth Annapureddy. 2016. Fixed point quantization of deep convolutional networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, pages 2849–2858, New York, USA.

Zhuoyuan Mao, Fabien Cromieres, Raj Dabre, Haiyue Song, and Sadao Kurohashi. 2020. JASS: Japanese-specific sequence to sequence pre-training for neural machine translation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages

3683–3691, Marseille, France. European Language Resources Association.

Yusuke Oda, Philip Arthur, Graham Neubig, Koichiro Yoshino, and Satoshi Nakamura. 2017. Neural machine translation via binary code prediction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 850–860, Vancouver, Canada. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, USA. Association for Computational Linguistics.

Hammam Riza, Michael Purwoadi, Gunarso, Teduh Uliniansyah, Aw Ai Ti, Sharifah Mahani Aljunied, Luong Chi Mai, Vu Tat Thang, Nguyen Phuong Thai, Vichet Chea, Rapid Sun, Sethserey Sam, Sopheap Seng, Khin Mar Soe, Khin Thandar Nwet, Masao Utiyama, and Chenche n Ding. 2016. Introduction of the Asian Language Treebank. In *Proceedings of the 2016 Conference of the Oriental Chapter of International Committee for Coordination and Standardization of Speech Database s and Assessment Technique (O-COCOSDA)*, pages 1–6, Bali, Indonesia.

Abigail See, Minh-Thang Luong, and Christopher D. Manning. 2016. Compression of neural machine translation models via pruning. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 291–301, Berlin, Germany. Association for Computational Linguistics.

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

Haiyue Song, Raj Dabre, Zhuoyuan Mao, Fei Cheng, Sadao Kurohashi, and Eiichiro Sumita. 2020. Pre-training via leveraging assisting languages for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 279–285, Online. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5926–5936, Long Beach, USA.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2020. Knowledge distillation for multilingual unsupervised neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3525–3535, Online. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th Neural Information Processing Systems Conference (NIPS)*, pages 3104–3112, Montréal, Canada. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 30th Neural Information Processing Systems Conference (NIPS)*, pages 5998–6008, Long Beach, USA. Curran Associates, Inc.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1575, Austin, USA. Association for Computational Linguistics.