This paper demonstrates that multilingual denoising pre-training produces significant performance gains across a wide variety of machine translation (MT) tasks. We present mBART—a sequence-to-sequence denoising auto-encoder pre-trained on large-scale monolingual corpora in many languages using the BART objective (Lewis et al., 2019). mBART is the first method for pre-training a complete sequence-to-sequence model by denoising full texts in multiple languages, whereas previous approaches have focused only on the encoder, decoder, or reconstructing parts of the text. Pre-training a complete model allows it to be directly fine-tuned for supervised (both sentence-level and document-level) and unsupervised machine translation, with no task-specific modifications. We demonstrate that adding mBART initialization produces performance gains in all but the highest-resource settings, including up to 12 BLEU points for low-resource MT and over 5 BLEU points for many document-level and unsupervised models. We also show that it enables transfer to language pairs with no bi-text or that were not in the pre-training corpus, and present extensive analysis of which factors contribute the most to effective pre-training.\footnote{Equal contribution. Most of the work was done when the first author worked at Facebook.}

1 Introduction

Despite its wide adoption for other NLP tasks (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019b; Lewis et al., 2019; Raffel et al., 2019),

self-supervised pre-training is not yet common practice in machine translation (MT). Existing approaches (Lample and Conneau, 2019; Edunov et al., 2019; Lewis et al., 2019; Raffel et al., 2019) have been proposed either to partially pre-train the model or to only focus on English corpora. In this paper, we show that significant performance gains are possible by pre-training a complete autoregressive model with an objective that noises and reconstructs full texts across many languages.

In this work, we present mBART—a multilingual sequence-to-sequence (Seq2Seq) denoising auto-encoder. mBART is trained by applying the BART (Lewis et al., 2019) to large-scale monolingual corpora across many languages. The input texts are noised by masking phrases and permuting sentences, and a single Transformer (Vaswani et al., 2017) model is learned to recover the texts. Different from other pre-training approaches for MT (Lample and Conneau, 2019; Song et al., 2019), mBART pre-trains a complete autoregressive Seq2Seq model. mBART is trained once for all languages, providing a set of parameters that can be fine-tuned for any of the language pairs in both supervised and unsupervised settings, without any task-specific or language-specific modifications or initialization schemes.

Extensive experiments demonstrate that this simple approach works remarkably well. We first focus on existing MT benchmarks. For supervised sentence-level MT, mBART initialization leads to significant gains (up to 12 BLEU points) across low/medium-resource pairs (<10M bi-text pairs), without sacrificing performance in high-resource settings. These results further improve with back-translation (BT), setting a new state-of-the-art on WMT16 English-Romanian and the FloRes test sets. For document-level MT, our document-level pre-training improves results by up to 5.5
Bleu points. For the unsupervised case, we see consistent gains and produce the first non-degenerate results for less related language pairs (e.g., 9.5 BLEU gain on Nepali-English). Previous pre-training schemes have only considered subsets of these applications, but we compare performance where possible and demonstrate that mBART consistently performs the best.

We also show that mBART enables new types of transfer across language pairs. For example, fine-tuning on bi-text in one language pair (e.g., Korean-English) creates a model that can translate from all other languages in the monolingual pre-training set (e.g., Italian-English), with no further training. We also show that languages not in the pre-training corpora can benefit from mBART, strongly suggesting that the initialization is at least partially language universal. Finally, we present a detailed analysis of which factors contribute the most to effective pre-training, including the number of languages and their overall similarity.

2 Multilingual Denoising Pre-training

2.1 Data: CC25 Corpus

Datasets We pre-train on 25 languages (CC25) extracted from the CC corpora (Wenzek et al., 2019; Conneau et al., 2019).\(^2\) CC25 includes languages from different families and with varied amounts of text (Figure 1). Following Lample and Conneau (2019), we re-balanced the corpus by up/down-sampling text from each language \(i\) with a ratio \(\lambda_i\):

\[
\lambda_i = \frac{1}{p_i} \cdot \frac{p_i^\alpha}{\sum_i p_i^\alpha},
\]

where \(p_i\) is the percentage of each language in CC-25. We use the smoothing parameter \(\alpha = 0.7\).

Pre-processing We tokenize with a sentence-piece model (SMP: Kudo and Richardson, 2018) learned on the full CC data that includes 250,000 subword tokens. Although not all of these languages are used for pre-training, this tokenization supports fine-tuning on additional languages. We do not apply additional preprocessing, such as true-casing or normalizing punctuation/characters.

2.2 Model: mBART

Our models follow the BART (Lewis et al., 2019) Seq2Seq pre-training scheme, as reviewed in this section. Whereas BART was only pretrained for English, we systematically study the effects of pre-training on different sets of languages.

Architecture We use a standard Seq2Seq Transformer architecture (Vaswani et al., 2017), with 12 layers of encoder and 12 layers of decoder with model dimension of 1024 on 16 heads (\(\sim 680\)M parameters). We include an additional layer-normalization layer on top of both the encoder and decoder, which we found stabilized training at FP16 precision.

Learning Our training data covers \(K\) languages: \(D = \{D_1, \ldots, D_K\}\) where each \(D_i\) is a collection of monolingual documents in language \(i\). We (1) assume access to a noising function \(g\), defined below, that corrupts text, and (2) train the model to predict the original text \(X\) given \(g(X)\). More formally, we aim to maximize \(L_\theta\):

\[
L_\theta = \sum_{D_i \in D} \sum_{X \in D_i} \log P(X|g(X); \theta),
\]

where \(X\) is an instance in language \(i\) and the distribution \(P\) is defined by the Seq2Seq model.

Noise Function Following Lewis et al. (2019), we use two types of noise in \(g\). We first remove spans of text and replace them with a mask token. We mask 35% of the words in each instance by randomly sampling a span length according to a Poisson distribution (\(\lambda = 3.5\)). We also permute

---

\(^2\)https://github.com/facebookresearch/ccnet.
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. It is also possible to use other noise types, such as those in Lample et al. (2018c), but we leave the exploration of the optimal noising strategy to future work.

**Instance Format**  For each instance of a batch, we sample a language id symbol <LID>, and we pack as many consecutive sentences as possible sampled from the corresponding corpus of <LID>, until either it hits the document boundary or reaches the 512 max token length. Sentences in the instance are separated by the end of sentence (<S>) token. Then, we append the selected <LID> token to represent the end of this instance. Pre-training at “multi sentence” level enables us to work on both sentence and document translation.

**Optimization**  Our full model (including 25 languages) is trained on 256 Nvidia V100 GPUs (32GB) for 500K steps. The total batch size is around 128K tokens per GPU, matching BART (Lewis et al., 2019) configuration. We use the Adam optimizer ($\epsilon = 1e^{-6}, \beta_2 = 0.98$) and linear learning rate decay scheduling. The total training time was approximately 2.5 weeks. We started the training with dropout 0.1 and reduced it to 0.05 at 250K steps and 0 at 400K steps. All experiments are done with Fairseq (Ott et al., 2019).

**Reproducibility**  One potential issue of the proposed approach is the replicability problem due to the requirement of massive monolingual corpora and computational resources, with fine-grained selection on hyper-parameters during pre-training. It is likely to get slightly different fine-tuning performance if we re-train the system again. Tackling on this, we will release the pre-trained checkpoints as well as the code with full instructions for pre-training a new model.

**Related Work: XLM(-R) and MASS**  There are several closely related approaches of multilingual pre-training for machine translation. XLM (Lample and Conneau, 2019) and XLM-R (Conneau et al., 2019) pretrain BERT (Devlin et al., 2019; Liu et al., 2019) in a multilingual fashion, and the resulted parameters can be used to initialize the translation model encoder. Different from XLM(-R), mBART simultaneously pre-trains the encoder and the decoder due to the Seq2Seq setup, which is more natural to adapt to machine translation applications.

Similar to mBART, MASS (Song et al., 2019) is also a Seq2Seq-based pre-training technique with “word-masking”. However, the decoder of MASS only predicted tokens that was masked in the encoder, whereas mBART reconstructs the full target sequence which allows to apply not only “masking” but any possible noise functions.

Furthermore, both XLM and MASS did not show evidence of the pre-trained models improving translation performance over two languages.

### 2.3 Pre-trained Models

To better measure the effects of different levels of multilinguality during pre-training, we built a range of models as follows:

- **mBART25**  We pre-train a model on all 25 languages, using the setting described in §2.2.

- **mBART06**  To explore the effect of pre-training on related languages, we pretrain a model on a subset of six European languages: Ro, It, Cs, Fr, Es, and En. For a fair comparison, we use $\sim 1/4$ of the mBART25 batch size, which allows our model to have the same number of updates per language during pre-training.

- **mBART02**  We pre-train bilingual models, using English and one other language for four language pairs: En-De, En-Ro, En-It. We use a batch size of $\sim 1/12$ of that in the mBART25.

- **BART-En/Ro**  To help establish a better understanding towards multilingual pre-training, we also train monolingual BART models on the En and Ro corpus only, respectively.

- **Random**  As additional baselines, we will also include a comparison with a model randomly initialized without pre-training for each translation task. Because the sizes of different downstream datasets vary, we always grid-search the hyper-parameters (architecture, dropout, etc.) to find the best non-pretrained configuration.
Figure 2: Framework for our multilingual denoising pre-training (left) and fine-tuning on downstream MT tasks (right), where we use (1) sentence permutation and (2) word-span masking as the injected noise. A special language id token is added at both the encoder and decoder. One multilingual pre-trained model is used for all tasks.

All models use the same vocabulary (§2.1). Not all tokens will frequently occur in all pre-training corpora, but later experiments show that this large vocabulary can improve generalization in multilingual settings even for unseen languages.

2.4 Scaling-up Matters

Scaling-up the training data and model parameters has been a key factor in pre-training (Devlin et al., 2019; Conneau et al., 2019; Raffel et al., 2019). Compared to conventional semi-supervised methods (e.g., back-translation) and other pre-training for MT (Lample and Conneau, 2019; Song et al., 2019), we pre-train mBART on much more monolingual data with relatively deeper architecture. This scale, in combination with the new multi-lingual training, is central to our results (sections 3 to 5), although future work could more carefully study the relative contributions of each.

3 Sentence-level Machine Translation

This section shows that mBART pre-training provides consistent performance gains in low to medium resource sentence-level MT settings, including bi-text only and with back translation, and outperforms other existing pre-training schemes (§3.2). We also present a detailed analysis to understand better which factors contribute the most to these gains (§3.3), and show that pre-training can even improve performance for languages not present in the pre-training data (§3.4).

3.1 Experimental Settings

Datasets We gather 24 pairs of publicly available parallel corpora that cover all the languages in CC25 (Figure 1). Most pairs are from previous WMT (Gu, Kk, Tr, Ro, Et, Lt, Fi, Lv, Cs, Es, Zh, De, Ru, Fr ↔ En) and IWSLT (Vi, Ja, Ko, Ni, Ar, It ↔ En) competitions. We also use FLoRes pairs (Guzmán et al., 2019, En-Ne and En-Si), En-Hi from IITB (Kunchukuttan et al., 2017), and En-My from WAT19 (Ding et al., 2018, 2019). We divide the datasets into three categories—low resource (<1M sentence pairs), medium resource (>1M and <10M), and high resource (>10M).

Fine-tuning & Decoding We fine-tune mBART on a single pair of bi-text data, feeding the source language into the encoder and decoding the target language. As shown in Figure 2, we load the pre-trained weights and train the MT model on bi-texts with teacher forcing. For all directions, we train with 0.3 dropout, 0.2 label smoothing, 2500 warm-up steps, 3e−5 maximum learning rate. We use a maximum of 40K training updates for all low and medium resource pairs and 100K for high resource pairs. The final models are selected based on validation likelihood. We use beam-search with beam size 5 for decoding. Our initial experiments indicate that the fine-tuning process is generally stable with different seeds. Therefore, to reduce the total computation, all our results are reported with single execution. We validate the statistical significance with scripts from the mosesdecoder.\(^3\)

Table 1: Low/medium resource machine translation Pre-training consistently improves over a randomly initialized baseline, with particularly large gains on low resource language pairs (e.g., Vi-En).

<table>
<thead>
<tr>
<th>Languages</th>
<th>En-Gu</th>
<th>En-Kk</th>
<th>En-Vi</th>
<th>En-Tr</th>
<th>En-Ja</th>
<th>En-Ko</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
<td>WMT19</td>
<td>WMT19</td>
<td>IWSLT15</td>
<td>WMT17</td>
<td>IWSLT17</td>
<td>IWSLT17</td>
</tr>
<tr>
<td>Size</td>
<td>10K</td>
<td>91K</td>
<td>133K</td>
<td>207K</td>
<td>223K</td>
<td>230K</td>
</tr>
<tr>
<td>Direction</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
</tr>
<tr>
<td>Random</td>
<td>0.0</td>
<td>0.0</td>
<td>0.8</td>
<td>0.2</td>
<td>23.6</td>
<td>24.8</td>
</tr>
<tr>
<td>mBART25</td>
<td>0.3</td>
<td>0.1</td>
<td>7.4</td>
<td>2.5</td>
<td>36.1</td>
<td>35.4</td>
</tr>
</tbody>
</table>

Table 2: High resource machine translation where all the datasets are from their latest WMT competitions. We only evaluate our models on En-X translation.

<table>
<thead>
<tr>
<th>Languages</th>
<th>Cs</th>
<th>Es</th>
<th>Zh</th>
<th>De</th>
<th>Ru</th>
<th>Fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
<td>FLoRes</td>
<td>ITTB</td>
<td>WMT18</td>
<td>WMT19</td>
<td>WMT17</td>
<td>WMT17</td>
</tr>
<tr>
<td>Size</td>
<td>11M</td>
<td>15M</td>
<td>25M</td>
<td>28M</td>
<td>29M</td>
<td>41M</td>
</tr>
<tr>
<td>RANDOM</td>
<td>16.5</td>
<td>33.2</td>
<td>35.0</td>
<td>30.9</td>
<td>31.5</td>
<td>41.4</td>
</tr>
<tr>
<td>mBART25</td>
<td>18.0</td>
<td>34.0</td>
<td>33.3</td>
<td>30.5</td>
<td>31.3</td>
<td>41.0</td>
</tr>
</tbody>
</table>

3.2 Main Results

As shown in Table 1, initializing with the pre-trained mBART25 weights shows gains on all the low and medium resource pairs when compared with randomly initialized baselines. We observe gains of 12 or more BLEU points on low resource pairs such as En-Vi, En-Tr, and noisily aligned pairs like En-Hi. Fine-tuning still fails in extremely low-resource cases such as En-Gu, which have ~10k examples. In these settings, unsupervised translation is more appropriate, see §5.2. For high resource cases (Table 2), we do not observe consistent gains, and pre-training slightly hurts performance when more than 25M parallel sentences are available. When a significant amount of bi-text data is given, we suspect that supervised training washes out the pre-trained weights.

Note that some reported runs of our baseline systems using the vanilla Transformers with randomly initialized weights have considerably noticeable gaps between the SoTA systems reported in the original competitions. The difference is mainly because we train and search

4http://matrix.statmt.org/.
multilingual translation with pretraining in future work.

**Plus Back-Translation** Back-translation (BT; Sennrich et al., 2016) is a standard approach to augment bi-text with target-side monolingual data. We combine our pre-training with BT and test it on low resource language pairs—En-Si and En-Ne—using the FLoRes dataset (Guzmán et al., 2019). We use the same monolingual data as Guzmán et al. (2019) to generate BT data. Figure 3 shows that initializing the model with our mBART25 pre-trained parameters improves BLEU scores at each iteration of back translation, resulting in new state-of-the-art results in all four translation directions. It indicates that the pre-trained mBART weights can be directly plugged into existing pipeline using BT.

**Compared with Other Pre-training Approaches** We also compare our pre-trained models with recent self-supervised pre-training methods, as shown in Table 3. We consider En-Ro translation, the only pair with established results. Our mBART model outperforms all the other pre-trained models, both with and without BT augmentation. We also show comparisons with the conventional BART model trained on the same En and Ro data only. Both have improvements over baselines, although worse than mBART results, indicating that pre-training in a multilingual setting is essential. Moreover, combining BT leads to additional gains, resulting in a new state-of-the-art for Ro-En translation.

### 3.3 Analysis

We also present additional analyses, to better quantify when our pre-training helps.

### How many languages should you pre-train on?

We investigate when it is helpful for pre-training to include languages other than the targeted language pair that will be used during fine tuning. Table 4 shows performance on four X-En pairs. Pre-training on more languages helps most when the target language monolingual data is limited (e.g., En-My, where the size of My is around 0.5% of En).

In contrast, when monolingual data is plentiful (De, Ro), pre-training on multiple languages slightly hurts the final results (<1 BLEU). In these cases, additional languages may reduce the capacity available for each test language. Additionally, the fact that mBART06 performs similar to mBART02 on Ro-En suggests that pre-training with similar languages is particularly helpful.
How many pre-training steps are needed? We plot Ro-En BLEU score vs. Pre-training steps in Figure 4, where we take the saved checkpoints (every 25K steps) and apply the same fine-tuning process described in §3.1. Without any pre-training, our model overfits and performs much worse than the baseline. However, after just 25K steps (5% of training), both models outperform the best baseline. The models keep improving by over 3 BLEU for the rest of pre-training and have not fully converged after 500K steps. In addition, mBART25 is consistently slightly worse than mBART02, which confirms the observation in Table 4.

How much bi-text is needed? Tables 1 and 2 show that pre-training consistently improves for low and medium resource language pairs. To verify this trend, we plot performance for differing sized subsets of the En-De dataset. More precisely, we take the full En-De corpus (28M pairs) and randomly sample 10K, 50K, 100K, 500K, 1M, 5M, 10M datasets. We compare performance without pre-training to the mBART02 results, as shown in Figure 5. The pre-trained model is able to achieve over 20 BLEU with only 10K training examples, whereas the baseline system scores 0. Unsurprisingly, increasing the size of bi-text corpus improves both models. Our pre-trained model consistently outperforms the baseline models, but the gap reduces with increasing amounts of bi-text, especially after 10M sentence pairs. This result confirms our observation in §3.2 that our pre-training does not help translation in high-resource pairs.

3.4 Generalization to Languages NOT in Pre-training

In this section, we show that mBART can improve performance even for languages that did not appear in the pre-training corpora, suggesting that the pre-training has language universal aspects. Similar phenomena have also been reported in other multilingual pre-training approaches in other NLP applications (Pires et al., 2019; Wang et al., 2019; Artetxe et al., 2019).

Experimental Settings We report results fine-tuning for three pairs, Ni-En, Ar-En, and De-Ni, using the pre-trained mBART25, mBART06, and mBART02 (EnRo) models. The mBART06 and mBART02 models are not pre-trained on Ar, De or Ni text, but all languages are in mBART25. Both De and Ni are European languages and are related to En, Ro, and the other languages in the mBART06 pre-training data.

Results As shown in Table 5, we find large gains from pre-training on English-Romanian, even when translating a distantly related unseen language (Arabic) and two unseen languages (German and Dutch). The best results are achieved when pre-training includes both test languages, although pre-training on other languages is surprisingly competitive.

Unseen Vocabularies Arabic is distantly related to the languages in mBART02 and mBART06, and has a disjoint character set. This means that its word embeddings are largely not estimated during pre-training. However, we obtain similar improvements on Ar-En pairs to those on Ni-En. This result suggests that the pre-trained Transformer layers learn universal properties of language that generalize well even with minimal lexical overlap.
Table 5: Generalization to unseen languages Language transfer results, fine-tuning on language-pairs without pre-training on them. mBART25 uses all languages during pre-training, while other settings contain at least one unseen language pair. For each model, we also show the gap to mBART25 results.

<table>
<thead>
<tr>
<th>Model</th>
<th>Source Languages</th>
<th>Target Languages</th>
<th>En-De</th>
<th>De-Nl</th>
<th>Ni-De</th>
<th>Ni-Nl</th>
<th>Ar-En</th>
<th>Ar-Ni</th>
<th>En-De</th>
<th>De-Nl</th>
<th>Ni-De</th>
<th>Ni-Nl</th>
</tr>
</thead>
<tbody>
<tr>
<td>RANDOM</td>
<td>None</td>
<td></td>
<td>34.6</td>
<td>(−8.7)</td>
<td>29.3</td>
<td>(−5.5)</td>
<td>27.5</td>
<td>(−10.1)</td>
<td>16.9</td>
<td>(−4.7)</td>
<td>21.3</td>
<td>(−6.4)</td>
</tr>
<tr>
<td>mBART02</td>
<td>En Ro</td>
<td></td>
<td>41.4</td>
<td>(−2.9)</td>
<td>34.5</td>
<td>(−0.3)</td>
<td>34.9</td>
<td>(−2.7)</td>
<td>21.2</td>
<td>(−0.4)</td>
<td>26.1</td>
<td>(−1.6)</td>
</tr>
<tr>
<td>mBART06</td>
<td>En Ro Cs It Fr Es</td>
<td></td>
<td>43.1</td>
<td>(−0.2)</td>
<td>34.6</td>
<td>(−0.2)</td>
<td>37.3</td>
<td>(−0.3)</td>
<td>21.1</td>
<td>(−0.5)</td>
<td>26.4</td>
<td>(−1.3)</td>
</tr>
<tr>
<td>mBART25</td>
<td>All</td>
<td></td>
<td>43.3</td>
<td>34.8</td>
<td>37.6</td>
<td>21.6</td>
<td>27.7</td>
<td>26.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Unseen Source or Target Languages Table 5 shows different performance when the unseen languages are on the source side, target side, or both sides. If both sides are unseen, the performance (in terms of difference from mBART25) is worse than where at least one language is seen during pre-training. Furthermore, although the En-X pairs perform similarly, mBART06 outperforms mBART02 on X-En pairs. Fine-tuning unseen languages on the source side is more difficult, and is worthy of extensive future study.

4 Document-level Machine Translation

We evaluate mBART on document-level machine translation tasks, where the goal is to translate segments of text that contain more than one sentence (up to an entire document). During pre-training, we use document fragments of up to 512 tokens, allowing the models to learn dependencies between sentences. We show that this pre-training significantly improves document-level translation.

4.1 Experimental Settings

Datasets We evaluate performance on two common document-level MT datasets: WMT19 En-De and TED15 Zh-En. For En-De, we use the document data from WMT19 to train our model, without any additional sentence-level data. The Zh-En dataset is from IWSLT 2014 and 2015 (Cettolo et al., 2012, 2015). Following Miculicich et al. (2018), we use 2010-2013 TED as the test set.

Pre-processing We pre-process with the approach used in pre-training. For each block, sentences are separated by end of sentence symbols (</S>) and the entire instance is ended with the specific language id (<LID>). On average, documents are split into 2–4 instances.

Fine-tuning & Decoding We use the same fine-tuning scheme as for sentence-level translation (§3.1), without using any task-specific techniques developed by previous work (Miculicich et al., 2018; Li et al., 2019), such as constrained contexts or restricted attention. For decoding, we simply pack the source sentences into blocks, and translate each instance block autoregressively. The model does not know how many sentences to generate in advance and decoding stops when <LID> is predicted. We use beam size 5 by default.

Baselines & Evaluation We train 4 models: a document-level (Doc-) MT model (§4.1) and a corresponding sentence-level (Sent-) MT model (§3.1) as the baseline, both with and without pre-training. We use mBART25 as the common pre-trained model for En-De and Zh-En. For En-De, even though our mBART25 Doc-MT model decodes multiple sentences together, the translated sentences can be aligned to the source sentences, which allows us to evaluate BLEU scores both on sentence-level (s-BLEU) and document-level (d-BLEU). For Zh-En, however, we cannot produce the same number of translated sentences as the reference due to alignment errors in the test data. We only provide the d-BLEU scores on this direction.

We also compare our models with Hierarchical Attention Networks (HAN, Miculicich et al., 2018) on Zh-En, which is the state-of-the-art non-pretraining approach for document-level translation for this pair. They combine two layers of attention—first within and then across sentences.

5Standard BLEU scores match n-grams at sentence-level. We also consider document-level where we match n-grams over the whole document resulting in a slightly higher score.
Table 6: Document-level machine translation on En-De and Zh-En. (∗) The randomly initialized Doc-MT model cannot produce translations aligned to the original sentences, so only document evaluation is possible.

<table>
<thead>
<tr>
<th>Model</th>
<th>Random</th>
<th>mBART25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s-BLEU</td>
<td>d-BLEU</td>
</tr>
<tr>
<td>Sent-MT</td>
<td>34.5</td>
<td>35.9</td>
</tr>
<tr>
<td>Doc-MT</td>
<td>∗</td>
<td>7.7</td>
</tr>
<tr>
<td>HAN (2018)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sent-MT</td>
<td>22.0</td>
<td>28.4</td>
</tr>
<tr>
<td>Doc-MT</td>
<td>3.2</td>
<td>29.6</td>
</tr>
</tbody>
</table>

4.2 Main Results

Table 6 shows the main results for both En-De and Zh-En at both sentence-level and document-level.

Random vs. Pre-trained The MT models initialized with pre-trained weights outperform randomly initialized models by large margins, for both sentence-level and document-level training. Our mBART25 models (both Sent-MT and Doc-MT) also outperform HAN (Miculicich et al., 2018), despite the fact that they are not customized for document-level MT.

Sent-MT vs. Doc-MT For En-De and En-Zh, the mBART25 Doc-MT models outperform mBART25 fine-tuned at sentence-level by large margins, reversing the trend seen for models without pre-training. For both datasets, randomly initialized Doc-MT fails to work, resulting in much worse results than the sentence-level models. Such large performance gaps indicate that pre-training is critical for document level performance. It is in general difficult to collect high-quality document-level data in large quantities, suggesting that pre-training may be a strong strategy for future work. We also include a sampled example in Figure 6.

5 Unsupervised Machine Translation

In addition to supervised machine translation, we also evaluate our model on tasks where no bi-text is available for the target language pair. We define three types of unsupervised translation:

1. No bi-text of any kind. A common solution is to learn from back-translation (Artetxe et al., 2017; Lample et al., 2018c). We show that mBART provides a simple and effective initialization scheme for these methods (§5.1).

2. No bi-text for the target pair, but both languages appear in bi-text corpora with other pairs. This setup is common for multilingual MT systems (Johnson et al., 2017; Gu et al., 2019). In this paper, we limit our focus to building models for single language pairs, and leave discussions for multilingual MT to future work.

3. No bi-text for the target pair is available, but there is bi-text for translating from some other language into the target language. mBART supports effective transfer, even if the source language has no bi-text of any form (§5.2).

5.1 Unsupervised Machine Translation via Back-Translation

Datasets We evaluate our pre-trained models on En-De, En-Ne, and En-Si. En and De are both European languages sharing many sub-words, whereas Ne and Si are quite distinct from En. We use the same test sets as supervised benchmarks §3.1, and use the same pre-training data (CC25) for back-translation to avoid introducing new information.

Learning Following Lample and Conneau (XLM, 2019), we initialize the translation model with the mBART weights, and then learn to predict the monolingual sentences conditioned on source sentences generated by on-the-fly BT. Furthermore, we constrain mBART to only generating tokens in target language\(^7\) for the first 1000 steps of on-the-fly BT, to avoid it copying the source text.

Results Table 7 shows the unsupervised translation results compared with non-pretrained models, as well as models with existing pre-training methods. Our models achieve large gains over non-pretrained models for all directions, and

\(^7\)We mask out the output probability of predicted tokens which appear less than 1% in the target monolingual corpus.
Figure 6: An example of document-level translation from mBART25 Sent-MT and Doc-MT, held out from the test set of TED15 Zh-En. The Doc-MT system produces much more fluent and coherent translation, which is closer to the reference translation. For instance, Doc-MT model produces several "And" to connect sentences to make it reads better, while the Sent-MT model does not contain global knowledge and produce sentences independently. Additionally, both systems produce much better translations than models without pre-training where the non-pretrained Doc-MT model completely fails to produce readable translation output.
outperform XLM significantly for dissimilar pairs (En-Ne, En-Si) where the existing approaches completely fail. For En-De, our model also performs comparably against XLM and MASS.

### 5.2 Unsupervised Machine Translation via Language Transfer

We also report results when the target language appears in a bi-text with some other source language.

**Datasets** We only consider X→En translation, and choose the bitexts of 12 language pairs from §3.1, covering Indic languages (Ne, Hi, Si, Gu), European languages (Ro, It, Cs, Ni), East Asian languages (Zh, Ja, Ko), and Arabic (Ar).

**Results** The pre-trained mBART25 model is fine-tuned on each language pair, and then evaluated on the rest of pairs, as seen in Table 8. We also present the direct fine-tuning performance (§3) on the diagonal, for reference. We see transfer for all pairs with all fine-tuned models except from Gu-En where the supervised model completely fails (0.3 BLEU). In some cases we can achieve similar (Cs-En) or even much better (Ne-En, Gu-En) results compared with the supervised results. We also show an example of language transfer in Figure 7.

As a comparison, we also apply the same procedure on randomly initialized models without pre-training, which always ends up with ≈ 0 BLEU. This indicates that multilingual pre-training is essential and produces universal representations across languages, so that once the model learns to translate one language to En, it learns to translate all languages with similar representations.

**When is language transfer useful?** Table 8 also shows that the size of transfer effects varies with the similarity of different languages. First, for most pairs, language transfer works better when fine-tuning is also conducted in the same language family, especially between Indic languages (Hi, Ne, Gu). However, significant vocabulary sharing is not required for effective transfer. For instance, Zh-En and It-En achieve the best transfer learning results on Ko-En and Ar-En, respectively. This is despite the low vocabulary overlap (even character overlap) between (Zh, Ko) and (It, Ar).

**With BT** We present a comparison of unsupervised MT with BT vs. language transfer in Table 9 where language transfer works better when there exists a close language translation to transfer from.

Moreover, we show promising results for combining these two techniques. We start from the best transferred model and apply (iterative) BT on the same monolingual corpus used in pre-training. Table 9 presents the results with 1 iteration of BT. We see improvements for all pairs. The complete analysis of both methods is left as future work.

### 6 Related Work

**Self-supervised Learning for Text Generation** This work inherits from the recent success brought by pre-training for NLP applications (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019; Yang et al., 2019b; Liu et al., 2019), especially for text generation (Radford et al., 2019; Song et al., 2019; Dong et al., 2019; Raffel et al., 2019; Lewis et al., 2019). The pre-trained models are usually used as the initialization for fine-tuning downstream tasks such as controllable language modeling (Shirish Keskar et al., 2019), summarization (Song et al., 2019; Liu and Lapata, 2019) and dialogue generation (Zhang et al., 2019).

Specifically for machine translation, unsupervised pre-training methods were also explored to improve the performance. Qi et al. (2018) investigated the application of pre-trained word embeddings for MT; Ramachandran et al. (2017) proposed to pre-train the encoder-decoder modules as two separate language models. Yang et al. (2019a); Zhu et al. (2020) explored fusion approaches to incorporate the pre-trained BERT weights to improve NMT training. In contrast to most prior work, we focus on pre-training one denoising autoencoder, and adapt the weights of the entire model for various MT applications.
Table 8: Unsupervised MT via language transfer on X-En translations. The model fine-tuned on one language pair is directly tested on another. We use gray color to show the direct fine-tuning results, and light gray color to show language transfer within similar language groups. We bold the highest transferring score for each pair.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Zh</th>
<th>Ja</th>
<th>Ko</th>
<th>Cs</th>
<th>Ro</th>
<th>Nl</th>
<th>It</th>
<th>Ar</th>
<th>Hi</th>
<th>Ne</th>
<th>Si</th>
<th>Gu</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>23.7</td>
<td>8.8</td>
<td>9.2</td>
<td>2.8</td>
<td>7.8</td>
<td>7.0</td>
<td>6.8</td>
<td>6.2</td>
<td>7.2</td>
<td>4.2</td>
<td>5.9</td>
<td>0.0</td>
</tr>
<tr>
<td>TED</td>
<td>12.2</td>
<td>16.9</td>
<td>24.6</td>
<td>5.7</td>
<td>8.5</td>
<td>9.5</td>
<td>9.1</td>
<td>8.7</td>
<td>9.6</td>
<td>8.8</td>
<td>11.1</td>
<td>0.0</td>
</tr>
<tr>
<td>Testing Languages</td>
<td>Z</td>
<td>H</td>
<td>23.7</td>
<td>8.8</td>
<td>9.2</td>
<td>2.8</td>
<td>7.8</td>
<td>7.0</td>
<td>6.8</td>
<td>6.2</td>
<td>7.2</td>
<td>4.2</td>
</tr>
<tr>
<td>Ja</td>
<td>9.9</td>
<td>19.1</td>
<td>12.2</td>
<td>0.9</td>
<td>4.8</td>
<td>6.4</td>
<td>5.1</td>
<td>5.6</td>
<td>4.7</td>
<td>4.2</td>
<td>6.5</td>
<td>0.0</td>
</tr>
<tr>
<td>Ko</td>
<td>5.8</td>
<td>16.9</td>
<td>24.6</td>
<td>5.7</td>
<td>8.5</td>
<td>9.5</td>
<td>9.1</td>
<td>8.7</td>
<td>9.6</td>
<td>8.8</td>
<td>11.1</td>
<td>0.0</td>
</tr>
<tr>
<td>Cs</td>
<td>9.3</td>
<td>15.1</td>
<td>17.2</td>
<td>21.6</td>
<td>19.5</td>
<td>17.0</td>
<td>16.7</td>
<td>16.9</td>
<td>13.2</td>
<td>15.1</td>
<td>16.4</td>
<td>0.0</td>
</tr>
<tr>
<td>Ro</td>
<td>16.2</td>
<td>18.7</td>
<td>17.9</td>
<td>23.0</td>
<td>37.8</td>
<td>22.3</td>
<td>21.6</td>
<td>22.6</td>
<td>16.4</td>
<td>18.5</td>
<td>22.1</td>
<td>0.0</td>
</tr>
<tr>
<td>Nl</td>
<td>14.4</td>
<td>30.4</td>
<td>32.3</td>
<td>21.2</td>
<td>27.0</td>
<td>43.3</td>
<td>34.1</td>
<td>31.0</td>
<td>24.6</td>
<td>23.3</td>
<td>27.3</td>
<td>0.0</td>
</tr>
<tr>
<td>Ir</td>
<td>16.9</td>
<td>25.8</td>
<td>27.8</td>
<td>17.1</td>
<td>23.4</td>
<td>30.2</td>
<td>39.8</td>
<td>30.6</td>
<td>20.1</td>
<td>18.5</td>
<td>23.2</td>
<td>0.0</td>
</tr>
<tr>
<td>Ar</td>
<td>5.8</td>
<td>15.5</td>
<td>12.8</td>
<td>12.7</td>
<td>12.0</td>
<td>14.7</td>
<td>14.7</td>
<td>37.6</td>
<td>11.6</td>
<td>13.0</td>
<td>16.7</td>
<td>0.0</td>
</tr>
<tr>
<td>Hi</td>
<td>3.2</td>
<td>10.1</td>
<td>9.9</td>
<td>5.8</td>
<td>6.7</td>
<td>6.1</td>
<td>5.0</td>
<td>7.6</td>
<td>23.5</td>
<td>14.5</td>
<td>13.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Ne</td>
<td>2.1</td>
<td>6.7</td>
<td>6.5</td>
<td>5.0</td>
<td>4.3</td>
<td>3.0</td>
<td>2.2</td>
<td>5.2</td>
<td>17.9</td>
<td>14.5</td>
<td>10.8</td>
<td>0.0</td>
</tr>
<tr>
<td>Si</td>
<td>5.0</td>
<td>5.7</td>
<td>3.8</td>
<td>3.8</td>
<td>1.3</td>
<td>0.9</td>
<td>0.5</td>
<td>3.5</td>
<td>8.1</td>
<td>8.9</td>
<td>13.7</td>
<td>0.0</td>
</tr>
<tr>
<td>Gu</td>
<td>8.2</td>
<td>8.5</td>
<td>4.7</td>
<td>5.4</td>
<td>3.5</td>
<td>2.1</td>
<td>0.0</td>
<td>6.2</td>
<td>13.8</td>
<td>13.5</td>
<td>12.8</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Table 9: BT vs. language transfer for unsupervised MT for X-En translations. For language transfer, we present the best transferring scores together with the language transferred from.

Table 9: BT vs. language transfer for unsupervised MT for X-En translations. For language transfer, we present the best transferring scores together with the language transferred from.

<table>
<thead>
<tr>
<th>Source</th>
<th>online BT</th>
<th>Transfer</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ro</td>
<td>30.5</td>
<td>23.0</td>
<td>33.9</td>
</tr>
<tr>
<td>Ne</td>
<td>10.0</td>
<td>17.9</td>
<td>22.1</td>
</tr>
<tr>
<td>Zh</td>
<td>11.3</td>
<td>9.2</td>
<td>15.0</td>
</tr>
<tr>
<td>Nl</td>
<td>28.5</td>
<td>34.1</td>
<td>35.4</td>
</tr>
</tbody>
</table>

Figure 7: An example of unsupervised MT via language transfer. mBART models finetuned with Ko or Zh are able to translate Ja sentence to En almost as correctly as in the supervised case.

Source online BT Transfer Combined
Ro 30.5 23.0 (Cs) 33.9
Ne 10.0 17.9 (Hi) 22.1
Zh 11.3 9.2 (Ko) 15.0
Nl 28.5 34.1 (It) 35.4

Table 8: Unsupervised MT via language transfer on X-En translations. The model fine-tuned on one language pair is directly tested on another. We use gray color to show the direct fine-tuning results, and light gray color to show language transfer within similar language groups. We bold the highest transferring score for each pair.

Multilinguality in NLP tasks This work is also related to the continual trend of multilingual language learning, including aligning multilingual word embeddings (Mikolov et al., 2013; Chen and Cardie, 2018; Lample et al., 2018b) into universal space, and learning crosslingual models (Wada and Iwata, 2018; Lample and Conneau, 2019; Conneau et al., 2019) to exploit shared representations across languages.

For MT, the most relevant field is multilingual translation (Firat et al., 2016; Johnson et al., 2017; Aharoni et al., 2019; Arivazhagan et al., 2019) where the ultimate goal is to jointly train one translation model that translates multiple language directions at the same time, and shares representations to improve the translation performance on low-resource languages (Gu et al., 2018). In this paper, we focus on multilingualism in the pre-training stage and fine-tune the learned model in the standard bilingual scenario.
Compared with multilingual translation, we do not require parallel data across multiple languages but the targeted direction, which improves the scalability to low-resource languages and specific domains.

**Document Translation** As one of the key applications, our work is also related to previous efforts for incorporating document-level context into neural machine translation (Wang et al., 2017; Jean et al., 2017; Tiedemann and Scherrer, 2017; Miculicich et al., 2018; Tu et al., 2018). Li et al. (2019) is the most relevant work that also utilized pre-trained encoder (BERT) for handling longer context. However, the focus has been on designing new task-specific techniques, and doing sentence-level translation with a wider input context. To the best of our knowledge, our multilingual pre-trained model is the first that shows improved results on document-level translation with standard Seq2Seq models.

**Unsupervised Translation** This work also summarizes the previous efforts of learning to translate between languages without a direct parallel corpus. When no parallel data of any kind is available, Artetxe et al. (2017) and Lample et al. (2018a) proposed to jointly learn denoising auto-encoder and back-translation from both directions, which, however, required good initialization and only worked well on similar language pairs. Wu et al. (2019) solve the problem by mining sentences from Wikipedia and using them as weakly supervised translation pairs. Similar to Lample and Conneau (2019) and Song et al. (2019), we follow the first approach and treat our pre-trained model as the initialization step. We also investigate unsupervised translation using language transfer, which is similar to Pourdamghani et al. (2019), where the authors generate translationese of the source language and train a system on high-resource languages to correct these intermediate utterances. It is also closely related to Conneau et al. (2018) and Artetxe et al. (2019) for cross-lingual representation learning where we also show representation learned by mBART can be easily transferred between language without supervised data.

7 Conclusion

We demonstrate that multilingual de-noising pre-training is able to significantly improve both supervised and unsupervised machine translation at both the sentence level and document level. We analyze when and how pre-training is most effective and can be combined with other approaches such as back-translation. Our results also show the transfer learning ability of the learned representations from multilingual pre-training.

In future work, we will scale-up the current pre-training to more languages, for example, an mBART100 model. The size of our model makes it expensive to deploy in production—future work will explore pre-training more efficient models.

Acknowledgments

We thank Marc’Aurelio Ranzato, Guillaume Lample, Alexis Conneau, and Michael Auli for sharing their expertise on low-resource and unsupervised machine translation and Peng-Jen Chen and Jiajun Shen for details about FloRes and WAT datasets. We also thank our colleagues at FAIR and FAIAR for valuable feedback.

References


Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc’Aurelio


Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A


Takashi Wada and Tomoharu Iwata. 2018. Unsupervised cross-lingual word embedding by multilingual neural language models. CoRR, abs/1809.02306. DOI: https://doi.org/10.18653/v1/P19-1300


