# Adapting High-resource NMT Models to Translate Low-resource Related Languages without Parallel Data

**Wei-Jen Ko**[1]*, **Ahmed El-Kishky**[2]*, **Adithya Renduchintala**[3], **Vishrav Chaudhary**[3], **Naman Goyal**[3], **Francisco Guzmán**[3], **Pascale Fung**[4], **Philipp Koehn**[5], **Mona Diab**[3]
[1]University of Texas at Austin, [2]Twitter Cortex, [3]Facebook AI
[4]The Hong Kong University of Science and Technology, [5]Johns Hopkins University
wjko@utexas.edu, aelkishky@twitter.com
{adirendu,vishrav,naman,fguzman,mdiab}@fb.com
pascale@ece.ust.hk, phi@jhu.edu

## Abstract

The scarcity of parallel data is a major obstacle for training high-quality machine translation systems for low-resource languages. Fortunately, some low-resource languages are linguistically related or similar to high-resource languages; these related languages may share many lexical or syntactic structures. In this work, we exploit this linguistic overlap to facilitate translating to and from a low-resource language with only monolingual data, in addition to any parallel data in the related high-resource language. Our method, NMT-Adapt, combines denoising autoencoding, back-translation and adversarial objectives to utilize monolingual data for low-resource adaptation. We experiment on 7 languages from three different language families and show that our technique significantly improves translation into low-resource language compared to other translation baselines.

## 1 Introduction

While machine translation (MT) has made incredible strides due to the advent of deep neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2014) models, this improvement has been shown to be primarily in well-resourced languages with large available parallel training data.

However with the growth of internet communication and the rise of social media, individuals worldwide have begun communicating and producing content in their native low-resource languages. Many of these low-resource languages are closely related to a high-resource language. One such example are "dialects": variants of a language traditionally considered oral rather than written. Machine translating dialects using models trained on the formal variant of a language (typically the high-resource variant which is sometimes considered the "standardized form") can pose a challenge due to the prevalence of non standardized spelling as well significant slang vocabulary in the dialectal variant. Similar issues arise from translating a low-resource language using a related high-resource model (e.g., translating Catalan with a Spanish MT model).

While an intuitive approach to better translating low-resource related languages could be to obtain high-quality parallel data. This approach is often infeasible due to lack specialized expertise or bilingual translators. The problems are exacerbated by issues that arise in quality control for low-resource languages (Guzmán et al., 2019). This scarcity motivates our task of learning machine translation models for low-resource languages while leveraging readily available data such as parallel data from a closely related language or monolingual data in the low-resource language.[1]

The use of monolingual data when little to no parallel data is available has been investigated for machine translation. A few approaches involve synthesising more parallel data from monolingual data using backtranslation (Sennrich et al., 2015) or mining parallel data from large multilingual corpora (Tran et al., 2020; El-Kishky et al., 2020b,a; Schwenk et al., 2019). We introduce NMT-Adapt, a zero resource technique that does not need parallel data of any kind on the low resource language.

We investigate the performance of NMT-Adapt at translating two directions for each low-resource language: (1) low-resource to English and (2) English to low-resource. We claim that translating into English can be formulated as a typical unsupervised domain adaptation task, with the high-resource language as the source domain and the

*This work was conducted while author was working at Facebook AI

[1]We use low-resource language and dialect or variant interchangeably.



*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 802–812
August 1–6, 2021. 

related low-resource, the target domain. We then show that adversarial domain adaptation can be applied to this related language translation task. For the second scenario, translating into the low-resource language, the task is more challenging as it involves unsupervised adaptation of the generated output to a new domain. To approach this task, NMT-Adapt jointly optimizes four tasks to perform low-resource translation: (1) denoising autoencoder (2) adversarial training (3) high-resource translation and (4) low-resource backtranslation.

We test our proposed method and demonstrate its effectiveness in improving low-resource translation from three distinct families: (1) Iberian languages, (2) Indic languages, and (3) Semitic languages, specifically Arabic dialects. We make our code and resources publicly available.[2]

## 2 Related Work

**Zero-shot translation** Our work is closely related to that of zero-shot translation (Johnson et al., 2017; Chen et al., 2017; Al-Shedivat and Parikh, 2019). However, while zero-shot translation translates between a language pair with no parallel data, there is an assumption that both languages in the target pair have some parallel data with other languages. As such, the system can learn to process both languages. In one work, Currey and Heafield (2019) improved zero-shot translation using monolingual data on the pivot language. However, in our scenario, there is no parallel data between the low-resource language and any other language. In other work, Arivazhagan et al. (2019) showed that adding adversarial training to the encoder output could help zero shot training. We adopt a similar philosophy in our multi-task training to ensure our low-resource target is in the same latent space as the higher-resource language.

**Unsupervised translation** A related set of work is the family of unsupervised translation techniques; these approaches translate between language pairs with no parallel corpus of any kind. In work by Artetxe et al. (2018); Lample et al. (2018a), unsupervised translation is performed by training denoising autoencoding and backtranslation tasks concurrently. In these approaches, multiple pretraining methods were proposed to better initialize the model (Lample et al., 2018b; Lample and Conneau, 2019; Liu et al., 2020; Song et al., 2019).

[2] https://github.com/wjko2/NMT-Adapt

Different approaches were proposed that used parallel data between X-Y to improve unsupervised translation between X-Z (Garcia et al., 2020a; Li et al., 2020; Wang et al., 2020). This scenario differs from our setting as it does not assume that Y and Z are similar languages. These approaches leverage a cross-translation method on a multilingual NMT model where for a parallel data pair ($S_x$,$S_y$), they translate $S_x$ into language Z with the current model to get $S'_z$. Then use ($S_y$,$S'_z$) as an additional synthesized data pair to further improve the model. Garcia et al. (2020b) experiment using multilingual cross-translation on low-resource languages with some success. While these approaches view the parallel data as auxiliary, to supplement unsupervised NMT, our work looks at the problem from a domain adaptation perspective. We attempt to use monolingual data in Z to make the supervised model trained on X-Y generalize to Z.

**Leveraging High-resource Languages to Improve Low-resource Translation** Several works have leveraged data in high-resource languages to improve the translation of similar low-resource languages. Neubig and Hu (2018) showed that it is beneficial to mix the limited parallel data pairs of low-resource languages with high-resource language data. Lakew et al. (2019) proposed selecting high-resource language data with lower perplexity in the low-resource language model. Xia et al. (2019) created synthetic sentence pairs by unsupervised machine translation, using the high-resource language as a pivot. However these previous approaches emphasize translating from the low-resource language to English, while the opposite direction is either unconsidered or shows poor translation performance. Siddhant et al. (2020) trained multilingual translation and denoising simultaneously, and showed that the model could translate languages without parallel data into English near the performance of supervised multilingual NMT.

**Similar language translation** Similar to our work, there have been methods proposed that leverage similar languages to improve translation. Hassan et al. (2017) generated synthetic English-dialect parallel data from English-main language corpus. However, this method assumes that the vocabulary in the main language could be mapped word by word into the dialect vocabulary, and they calculate the corresponding word for substitution using

localized projection. This approach differs from our work in that it relies on the existence of a seed bilingual lexicon to the dialect/similar language. Additionally, the approach only considers translating from a dialect to English and not the reverse direction. Other work trains a massively multilingual many-to-many model and demonstrates that high-resource training data improves related low-resource language translation (Fan et al., 2020). In other work, Lakew et al. (2018) compared ways to model translations of different language varieties, in the setting that parallel data for both varieties is available, the variety for some pairs may not be labeled. Another line of work focus on translating between similar languages. In one such work, Pourdamghani and Knight (2017) learned a character-based cipher model. In other work, Wan et al. (2020) improved unsupervised translation between the main language and the dialect by separating the token embeddings into pivot and private parts while performing layer coordination.

## 3 Method

We describe the NMT-Adapt approach to translating a low-resource language into and out of English without utilizing any low-resource language parallel data. In Section 3.1, we describe how NMT-Adapt leverages a novel multi-task domain adaptation approach to translating English into a low-resource language. In Section 3.2, we then describe how we perform source-domain adaptation to translate a low-resource language into English. Finally, in Section 3.3, we demonstrate how we can leverage these two domain adaptations, to perform iterative backtranslation – further improving translation quality in both directions.

### 3.1 English to Low-resource

To translate from English into a low-resource language, NMT-Adapt is initialized with a pretrained mBART model whose pretraining is described in (Liu et al., 2020). Then, as shown in Figure 1, we continue to train the model simultaneously with *four* tasks inspired by (Lample et al., 2018a) and update the model with a weighted sum of the gradients from different tasks.

The language identifying tokens are placed at the same position as in mBART. For the encoder, *both* high and low-resource language source text, with and without noise, use the language token of the high-resource language [HRL] in the pretrained mBART. For the decoder, the related high and low-resource languages use their own, *different*, language tokens. We initialize the language token embedding of the low-resource language with the embedding from the high-resource language token.

**Task 1: Translation** The first task is translation from English into the high-resource language (HRL) which is trained using readily available high-resource parallel data. This task aims to transfer high-resource translation knowledge to aid in translating into the low-resource language. We use the cross entropy loss formulated as follows:

$$\mathcal{L}_t = \mathcal{L}_{CE}(\mathcal{D}(Z_{En}, [HRL]), X_{HRL}) \quad (1)$$

, where $Z_{En} = \mathcal{E}(X_{En}, [En])$. $(X_{En}, X_{HRL})$ is a parallel sentence pair. $\mathcal{E}, \mathcal{D}$ denotes the encoder and decoder functions, which take (input, language token) as parameters. $\mathcal{L}_{CE}$ denotes the cross entropy loss.

**Task 2: Denoising Autoencoding** For this task, we leverage monolingual text by introducing noise to each sentence, feeding the noised sentence into the encoder, and training the model to generate the original sentence. The noise we use is similar to (Lample et al., 2018a), which includes a random shuffling and masking of words. The shuffling is a random permutation of words, where the position of words is constrained to shift at most 3 words from the original position. Each word is masked with a uniform probability of 0.1. This task aims to learn a feature space for the languages, so that the encoder and decoder could transform between the features and the sentences. This is especially necessary for the low-resource language if it is not already pretrained in mBART. Adding noise was shown to be crucial to translation performance in (Lample et al., 2018a), as it forces the learned feature space to be more robust and contain high-level semantic knowledge.

We train the denoising autoencoding on both the low-resource and related high-resource languages and compute the loss as follows:

$$\mathcal{L}_{da} = \sum_{i=LRL,HRL} \mathcal{L}_{CE}(\mathcal{D}(Z_i, [i]), X_i) \quad (2)$$

, where $Z_i = \mathcal{E}(\mathcal{N}(X_i), [HRL])$. $X_i$ is from the monolingual corpus.

**Task 3: Backtranslation** For this task, we train on

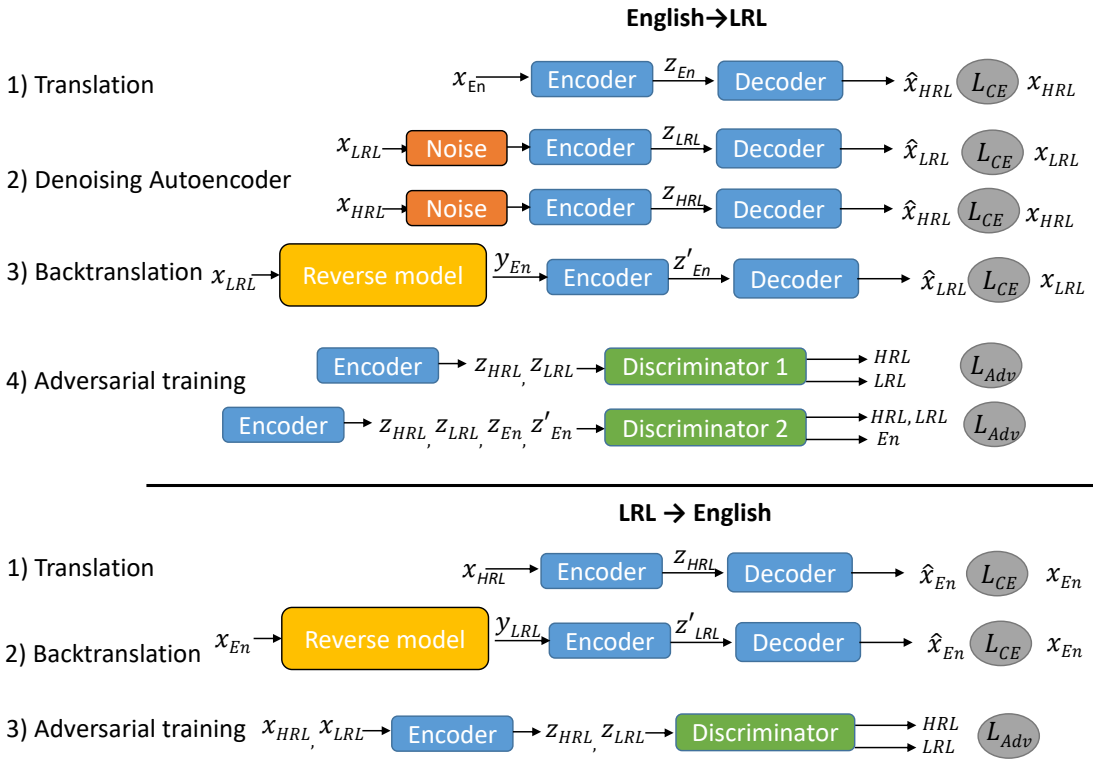


Figure 1: Illustration of the training tasks for translating from English into a low-resource language (LRL) and from an LRL to English.

English to low-resource backtranslation data. The aim of this task is to capture a language-modeling effect in the low-resource language. We describe how we obtain this data using the high-resource translation model to bootstrap backtranslation in Section 3.3.

The objective used is,

$$\mathcal{L}_{bt} = \mathcal{L}_{CE}(\mathcal{D}(Z'_{En}, [LRL]), X_{LRL}) \quad (3)$$

, where $Z'_{En} = \mathcal{E}(Y_{En}, [En])$. $(Y_{En}, X_{LRL})$ is an English to low-resource backtranslation pair.

**Task 4: Adversarial Training** The final task aims to make the encoder output language-agnostic features. The representation is language agnostic to the noised high and low-resource languages as well as English. Ideally, the encoder output should contain the semantic information of the sentence and little to no language-specific information. This way, any knowledge learned from the English to high-resource parallel data can be directly applied to generating the low-resource language by simply switching the language token during inference, without capturing spurious correlations (Gu et al., 2019a).

To adversarially mix the latent space of the encoder among the three languages, we use two critics (discriminators). The critics are recurrent networks to ensure that they can handle variable-length text input. Similar to Gu et al. (2019b), the adversarial component is trained using a Wasserstein loss, which is the difference of expectations between the two types of data. This loss minimizes the earth mover's distance between the distributions of different languages. We compute the loss function as follows:

$$\mathcal{L}_{adv1} = \mathbb{E}[Disc(Z_{HRL})] - \mathbb{E}[Disc(Z_{LRL})] \quad (4)$$

$$\mathcal{L}_{adv2} = \mathbb{E}[Disc(Z_{HRL} \cup Z_{LRL})] - \mathbb{E}[Disc(Z_{En} \cup Z'_{En})] \quad (5)$$

As shown in Equation 4, the first critic is trained to distinguish between the high and low-resource languages. Similarly, in Equation 5, the second critic is trained to distinguish between English and non-English (both high, and low-resource languages).

**Fine-tuning with Backtranslation**: Finally, we found that after training with the four tasks concurrently, it is beneficial to fine-tune solely using backtranslation for one pass before inference. We posit that this is because while spurious correlations are reduced by the adversarial training, they are not completely eliminated and using solely the

language tokens to control the output language is not sufficient. By fine-tuning on backtranslation, we are further adapting to the target side and encouraging the output probability distribution of the decoder to better match the desired output language.

### 3.2 Low-resource to English

We propose to model translating from the low-resource language to English as a domain adaptation task and design our model based on insights from domain-adversarial neural network (DANN) (Ganin et al., 2017), a domain adaptation technique widely used in many NLP tasks. This time, we train three tasks simultaneously:

**Task 1: Translation** We train high-resource to English translation on parallel data with the goal of adapting this knowledge to translate low-resource sentences. We compute this loss as follows:

$$\mathcal{L}_t = \mathcal{L}_{CE}(\mathcal{D}(Z_{HRL}, [En]), X_{En}) \quad (6)$$

, where $Z_{HRL} = \mathcal{E}(X_{HRL}, [HRL])$.

**Task 2: Backtranslation** Low-resource to English backtranslation translation, which we describe in Section 3.3. The objective is as follows:

$$\mathcal{L}_t = \mathcal{L}_{CE}(\mathcal{D}(Z'_{LRL}, [En]), X_{En}) \quad (7)$$

, where $Z'_{LRL} = \mathcal{E}(Y_{LRL}, [HRL])$.

**Task 3: Adversarial Training** We feed the sentences from the monolingual corpora of the high- and low-resource corpora into the encoder, and the encoder output is trained so that its input language cannot be distinguished by a critic. The goal is to encode the low-resource data into a shared space with the high-resource, so that the decoder trained on the translation task can be directly used. No noise was added to the input, since we did not observe an improvement. There is only one recurrent critic, which uses the Wasserstein loss and is computed as follows:

$$\mathcal{L}_{adv} = \mathbb{E}[Disc(Z_{HRL})] - \mathbb{E}[Disc(Z_{LRL})] \quad (8)$$

, where $Z_{LRL} = \mathcal{E}(X_{LRL}, [HRL])$.

Similar to the reverse direction, we initialize NMT-Adapt with a pretrained mBART, and use the same language token for high-resource and low-resource in the encoder.

### 3.3 Iterative Training

We describe how we can alternate training into/out-of English models to create better backtranslation data improving overall quality.

**Algorithm 1** Iterative training

$M_0^{LRL \to En} \leftarrow$ Train HRL to En model
$X_{mono} \leftarrow$ Monolingual LRL corpus
$X_{En} \leftarrow$ English sentences in the En-HRL parallel corpus
**for** k in 1,2... **do**
    // Generate backtranslation pairs
    Compute $M_{k-1}^{LRL \to En}(X_{mono})$

    // Train model as in Sec 3.1
    $M_k^{En \to LRL} \leftarrow$ trained En to LRL model

    // Generate backtranslation pairs
    Compute $M_k^{En \to LRL}(X_{En})$

    // Train model as in Sec 3.2
    $M_k^{LRL \to En} \leftarrow$ trained LRL to En model

    **if** Converged **then** break;

The iterative training process is described in Algorithm 1. We first create English to low-resource backtranslation data by fine-tuning mBART on the high-resource to English parallel data. Using this model, we translate monolingual low-resource text into English treating the low-resource sentences as if they were in the high-resource language. The resulting sentence pairs are used as backtranslation data to train the first iteration of our English to low-resource model.

After training English to low-resource, we use the model to translate the English sentences in the English-HRL parallel data into the low-resource language, and use those sentence pairs as backtranslation data to train the first iteration of our low-resource to English model.

We then use the first low-resource to English model to generate backtranslation pairs for the second English to low-resource model. We iteratively repeat this process of using our model of one direction to improve the other direction.

## 4 Experiments

### 4.1 Datasets

We experiment on three groups of languages. In each group, we have a large quantity of parallel training data for one language(high-resource) and no parallel for the related languages to simulate a low-resource scenario.

Our three groupings include (i) *Iberian languages*, where we treat Spanish as the high-

| Language | Group | Training Set | Train-Size | Test Set | Test-size | Monolingual | Mono-Size |
|---|---|---|---|---|---|---|---|
| Spanish | Iberian | QED (Guzman et al., 2013) | 694k | N/A | - | CC-100 | 1M |
| Catalan | Iberian | N/A | - | Global Voices (Tiedemann, 2012) | 15k | CC-100 | 1M |
| Portuguese | Iberian | N/A | - | TED (Qi et al., 2018) | 8k | CC-100 | 1M |
| Hindi | Indic | IIT Bombay (Kunchukuttan et al., 2018) | 769k | N/A | - | CC-100 | 1M |
| Marathi | Indic | N/A | - | TICO-19 (Anastasopoulos et al., 2020) | 2k | CC-100 | 1M |
| Nepali | Indic | N/A | - | FLoRes (Guzmán et al., 2019) | 3k | CC-100 | 1M |
| Urdu | Indic | N/A | - | TICO-19 (Anastasopoulos et al., 2020) | 2k | CC-100 | 1M |
| MSA | Arabic | QED (Guzman et al., 2013) | 465k | N/A | - | CC-100 | 1M |
| Egyptian Ar. | Arabic | N/A | - | Forum (Chen et al., 2018) | 11k | CC-100 | 1.2M |
| Levantine Ar. | Arabic | N/A | - | Web text (Raytheon, 2012) | 11k | CC-100 | 1M |

Table 1: The sources and size of the datasets we use for each language. The HRLs are used for training and the LRLs are used for testing.

resource and Portuguese and Catalan as related lower-resource languages. (ii) *Indic languages* where we treat Hindi as the high-resource language, and Marathi, Nepali, and Urdu as lower-resource related languages (iii) *Arabic*, where we treat Modern Standard Arabic (MSA) as the high-resource, and Egyptian and Levantine Arabic dialects as low-resource. Among the languages, the relationship between Urdu and Hindi is a special setting; while the two languages are mutually intelligible as spoken languages, they are written using different scripts. Additionally, in our experimental setting, all low-resource languages except for Nepali were not included in the original mBART pretraining.

The parallel corpus for each language is described in Table 1. Due to the scarcity of any parallel data for a few low-resource languages, we are not able to match the training and testing domains. For monolingual data, we randomly sample 1M sentences for each language from the CC-100 corpus[3] (Conneau et al., 2020; Wenzek et al., 2020). For quality control, we filter out sentences if more than $40\%$ of characters in the sentence do not belong to the alphabet set of the language. For quality and memory constraints, we only use sentences with length between 30 and 200 characters.

**Collecting Dialectical Arabic Data** While obtaining low-resource monolingual data is relatively straightforward, as language identifiers are often readily available for even low-resource text (Jauhiainen et al., 2019), identifying dialectical data is often less straightforward. This is because many dialects have been traditionally considered oral rather than written, and often lack standardized spelling, significant slang, or even lack of mutual intelligibility from the main language. In general, dialectical data has often been grouped in with the main language in language classifiers.

We describe the steps we took to obtain reliable dialectical Arabic monolingual data. As the CC-100 corpus does not distinguish between Modern Standard Arabic (MSA) and its dialectical variants, we train a finer-grained classifier that distinguishes between MSA and specific colloquial dialects. We base our language classifier on a BERT model pretrained for Arabic (Safaya et al., 2020) and finetune it for six-way classification: (i) Egyptian, (ii) Levantine, (iii) Gulf, (iv) Maghrebi, (v) Iraqi dialects as well as (vi) the literary Modern Standard Arabic (MSA). We use the data from (Bouamor et al., 2018) and (Zaidan and Callison-Burch, 2011) as training data, and the resulting classifier has an accuracy of $91\%$ on a held-out set. We take our trained Arabic dialect classifier and further classify Arabic monolingual data from CC-100 and select MSA, Levantine and Egyptian sentences as Arabic monolingual data for our experiments.

### 4.2 Training Details

We use the RMSprop optimizer with learning rate $0.01$ for the critics and the Adam optimizer for the rest of the model. We train our model using eight GPUs and a batch size of $1024$ tokens per GPU. We update the parameters once per eight batches. For the adversarial task, the generator is trained once per three updates, and the critic is trained every update.

Each of the tasks of (i) translation, (ii) backtranslation as well as (iii) LRL and HRL denoising (only for En→LRL direction), have the same number of samples and their cross entropy loss has equal weight. The adversarial loss, $\mathcal{L}_{adv}$, has the same weight on the critic, while it has a multiplier of $-60$ on the generator (encoder). This multiplier was tuned to ensure convergence and is negative as it's opposite to the discriminator loss.

For the first iteration, we train $128$ epochs from

[3] http://data.statmt.org/cc-100/

| *En→ LRL* | | Un-adapted Model | Adapted Models | | | |
|---|---|---|---|---|---|---|
| **LRL** | **HRL** | En→HRL | Adv | BT | BT+Adv | BT+Adv+fine-tune |
| Portuguese | Spanish | 3.8 | 10.1 | 14.8 | 18.0 | **21.2** |
| Catalan | Spanish | 6.8 | 9.1 | 21.2 | 22.5 | **23.6** |
| Marathi | Hindi | 7.3 | 8.4 | 9.5 | 15.6 | **16.1** |
| Nepali | Hindi | 11.2 | 17.6 | 16.7 | 25.3 | **26.3** |
| Urdu | Hindi | 0.3 | 3.4 | 0.2 | **7.2** | - |
| Egyptian Arabic | MSA | 3.5 | 3.8 | **8.0** | **8.0** | **8.0** |
| Levantine Arabic | MSA | 2.1 | 2.1 | 4.8 | **5.1** | 4.7 |

Table 2: BLEU score of the first iteration on the English to low-resource direction. Both the adversarial (Adv) and backtranslation (BT) components contribute to improving the results. The fine-tuning step is omitted for Urdu as decoding is already restricted to a different script-set from the related high-resource language.

| *LRL→En* | | Un-adapted Model | Adapted Models | | |
|---|---|---|---|---|---|
| **LRL** | **HRL** | HRL→En | Adv | BT | BT+Adv |
| Portuguese | Spanish | 12.3 | 21.7 | 32.7 | **36.0** |
| Catalan | Spanish | 12.2 | 13.9 | **25.3** | 24.6 |
| Marathi | Hindi | 3.9 | 7.0 | 8.1 | **12.7** |
| Nepali | Hindi | 14.8 | 16.9 | 14.1 | **18.2** |
| Urdu | Hindi | 0.3 | 1.0 | **10.5** | **10.5** |
| Egyptian Arabic | MSA | 14.9 | 14.0 | 15.2 | **15.8** |
| Levantine Arabic | MSA | **9.3** | 6.7 | **9.3** | 9.0 |

Table 3: BLEU score of the first iteration on the LRL to English direction. Both the adversarial(Adv) and backtranslation (BT) components contribute to improving the results.

English to the low-resource language and 64 iterations from low-resource language to English. For the second iteration we train 55 epochs for both directions. We follow the setting of (Liu et al., 2020) for all other settings and training parameters.

The critics consist of four layers: the third layer is a bidirectional GRU and the remaining three are fully connected layers. The hidden layer sizes are 512, 512 and 128 and we use an SELU activation function.

We ran experiments on 8-GPUs. Each iteration took less than 3 days and we used publicly available mBART-checkpoints for initialization. GPU memory usage of our method is only slightly larger than mBART. While we introduce additional parameters in discriminators, these additional parameters are insignificant compared to the size of the mBART model.

### 4.3 Results

We present results of applying NMT-Adapt to low-resource language translation.

#### 4.3.1 English to Low-Resource

We first evaluate performance of translating into the low-resource language. We compare the first iteration of NMT-Adapt to the following baseline systems: **(i)** En→HRL Model: directly using the model trained for En→HRL translation. **(ii)** Adversarial: Our full model without using the backtranslation objective and without the final fine-tuning. **(iii)** Backtranslation: mBART fine-tuned on backtranslation data created using the HRL→En model. **(iv)** BT+Adv: Our full model without the final fine-tuning. **(v)** BT+Adv+fine-tune: Our full model (NMT-Adapt) as described in Section 3.

As seen in Table 2, using solely the adversarial component only, we generally see improvement in the BLEU scores over using the high-resource translate model. This suggests that our proposed method of combining denoising autoencoding with adversarial loss is effective in adapting to a new target output domain.

Additionally, we observe a large improvement using only backtranslation data. This demonstrates that using the high-resource translation model to create LRL-En backtranslation data is highly effective for adapting to the low-resource target.

We further see that combining adversarial and backtranslation tasks further improve over each individually, showing that the two components are complementary. We also experimented on En-HRL translation with backtranslation but without adversarial loss. However, this yielded much worse results, showing that the improvement is not simply due to multitask learning.

For Arabic, backtranslation provides most of the gain, while for Portuguese and Nepali, the adversarial component is more important. For some languages like Marathi, the two components provides small gains individually, but shows a large

improvement while combined.

For Urdu, we found that backtranslation only using the Hindi model completely fails; this is intuitive as Hindi and Urdu are in completely different scripts and using a Hindi model to translate Urdu results in effectively random backtranslation data. When we attempt to apply models trained with the adversarial task, the model generates sentences with mixed Hindi, Urdu, and English. To ensure our model solely outputs Urdu, we restricted the output tokens by banning all tokens containing English or Devanagari (Hindi) characters. This allowed our model to output valid and semantically meaningful translations. This is an interesting result as it shows that our adversarial mixing allows translating similar languages even if they're written in different scripts. We report the BLEU score with the restriction. Since the tokens are already restricted, we skip the final fine-tuning step.

#### 4.3.2 Low-resource to English

Table 3 shows the results of the first iteration from translating from a low-resource language into English. We compare the following systems **(i)** HRL→En model: directly using the model trained for HRL→En translation. **(ii)** Adversarial: similar to our full model, but without using the backtranslation objective. **(iii)** Backtranslation: mBART fine-tuned on backtranslation data from our full model in the English-LRL direction. **(iv)** BT+Adv: Our full model.

For this direction, we can see that both the backtranslation and the adversarial domain adaptation components are generally effective. The exception is Arabic which may be due to noisiness of our dialect classification compared to low-resource language classification. Another reason could be due to the lack of written standardization for spoken dialects in comparison to low-resource, but standardized languages.

For these experiments, we did not apply any special precautions for Urdu on this direction despite it being in a different script from Hindi.

#### 4.3.3 Iterative Training

Table 4 shows the results of two iterations of training. For languages other than Arabic dialects, the second iteration generally shows improvement over the first iteration, showing that we can leverage an improved model in one direction to further improve the reverse direction. We found that the improvement after the third iteration is marginal.

We compare our results with a baseline using the HRL language as a pivot. The baseline uses a fine tuned mBART (Liu et al., 2020) to perform supervised translation between English and the HRL, and uses MASS (Song et al., 2019) to perform unsupervised translation between the HRL and the LRL. The mBART is tuned on the same parallel data used in our method, and the MASS uses the same monolingual data as in our method. For all languages and directions, our method significantly outperforms the pivot baseline.

#### 4.3.4 Comparison with Other Methods

In table 5, we compare a cross translation method using parallel corpora with multiple languages as auxiliary data (Garcia et al., 2020b) as well as results reported in (Guzmán et al., 2019) and (Liu et al., 2020). All methods use the same test set, English-Hindi parallel corpus, and tokenization for fair comparison. For English to Nepali, NMT-Adapt outperforms previous unsupervised methods using Hindi or multilingual parallel data, and is competitive with supervised methods. For Nepali to English direction, our method achieves similar performance to previous unsupervised methods. Note that we use a different tokenization than in table 3 and 4, to be consistent with previous work.

#### 4.3.5 Monolingual Data Ablation

Table 6 shows the first iteration English to Marathi results while varying the amount of monolingual data used. We see that the BLEU score increased from 11.3 to 16.1 as the number of sentences increased from 10k to 1M showing additional monolingual data significantly improves performance.

## 5 Conclusion

We presented NMT-Adapt, a novel approach for neural machine translation of low-resource languages which assumes zero parallel data or bilingual lexicon in the low-resource language. Utilizing parallel data in a similar high resource language as well as monolingual data in the low-resource language, we apply unsupervised adaptation to facilitate translation to and from the low-resource language. Our approach combines several tasks including adversarial training, denoising language modeling, and iterative back translation to facilitate the adaptation. Experiments demonstrate that this combination is more effective than any task on its own and generalizes across many different language groups.

| | English→LRL | | | LRL→English | | |
|---|---|---|---|---|---|---|
| **Language** | NMT-Adapt It.1 | NMT-Adapt It.2 | MBART+MASS | NMT-Adapt It.1 | NMT-Adapt It.2 | MBART+MASS |
| Portuguese | 21.2 | **30.7** | 26.6 | 36.0 | **39.8** | 38.1 |
| Catalan | 23.6 | **27.2** | 23.3 | 24.6 | **27.7** | 22.9 |
| Marathi | 16.1 | **19.2** | 13.1 | 12.7 | **15.0** | 5.8 |
| Nepali | **26.3** | **26.3** | 11.9 | 18.2 | **18.8** | 2.1 |
| Urdu | 7.2 | **14.6** | 5.1 | 10.5 | **13.6** | 4.9 |
| Egyptian Ar. | **8.0** | 6.6 | 3.3 | **15.8** | - | 11.7 |
| Levantine Ar. | **5.1** | 4.5 | 1.9 | **9.0** | - | 6.0 |

Table 4: BLEU results of iterative training. The second iteration generally improves among the first iteration, and NMT-Adapt outperforms the MBART+MASS baseline. For Arabic, as iteration 2 into Arabic was worse than iteration 1, we omit the corresponding iteration 2 into English.

| | | BLEU | |
|---|---|---|---|
| | | En→Ne | Ne→En |
| Unsupervised+ Hi parallel | NMT-Adapt | **9.2** | **18.8** |
| | (Guzmán et al., 2019) | 8.3 | **18.8** |
| | (Liu et al., 2020) | - | 17.9 |
| Unsupervised+ Multi. parallel | (Garcia et al., 2020b) | 8.9 | 21.7 |
| Sup. with Hi | (Guzmán et al., 2019) | 8.8 | **21.5** |
| | (Liu et al., 2020) | **9.6** | 21.3 |
| Sup. w/o Hi | (Guzmán et al., 2019) | 4.3 | 7.6 |

Table 5: Comparison with previous work on FLoRes dataset. NMT-Adapt outperforms previous unsupervised methods on En→Ne, and achieves similar performance to unsupervised baselines on Ne→En.

| **# sentences** | **BLEU** |
|---|---|
| 10k | 11.3 |
| 100k | 14.1 |
| 1M | 16.1 |

Table 6: First iteration English to Marathi results with variable amount of monolingual data.

## References


Maruan Al-Shedivat and Ankur P. Parikh. 2019. Consistency by agreement in zero-shot neural machine translation. In *NAACL*.

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federman, Dmitriy Genzel, Francisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Mengmeng Niu Graham Neubig, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. 2020. Tico-19: the translation initiative for covid-19. In *arXiv*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019. The missing ingredient in zero-shot neural machine translation. In *arXiv*.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *ICLR*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The madar arabic dialect corpus and lexicon. In *RECL*.

Song Chen, Jennifer Tracey, Christopher Walker, and Stephanie Strassel. 2018. Bolt arabic discussion forum parallel training data. In *LDC2019T01*.

Yun Chen, Yang Liu, Yong Cheng, and Victor O.K. Li. 2017. A teacher-student framework for zero-resource neural machine translation. In *ACL*.

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

Anna Currey and Kenneth Heafield. 2019. Zero-resource neural machine translation with monolingual pivot data. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020a. A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969.

Ahmed El-Kishky, Philipp Koehn, and Holger Schwenk. 2020b. Searching the web for cross-lingual parallel data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2417–2420.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2020. Beyond english-centric multilingual machine translation. *arXiv preprint arXiv:2010.11125*.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2017.

Domain-adversarial training of neural networks. In *EMNLP*.

Xavier Garcia, Pierre Foret, Thibault Sellam, and Ankur Parikh. 2020a. A multilingual view of unsupervised machine translation. In *Findings of EMNLP*.

Xavier Garcia, Aditya Siddhant, Orhan Firat, and Ankur P. Parikh. 2020b. Harnessing multilinguality in unsupervised machine translation for rare languages. In *arXiv*.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. 2019a. Improved zero-shot neural machine translation via ignoring spurious correlations. In *ACL*.

Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. 2019b. DialogWAE: Multimodal response generation with conditional wasserstein auto-encoder.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The flores evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english.

Francisco Guzman, Hassan Sajjad, A Abdelali, and S Vogel. 2013. The amara corpus: Building resources for translating the web's educational content. In *IWSLT*.

Hany Hassan, Mostafa Elaraby, and Ahmed Tawfik. 2017. Synthetic data for neural machine translation of spoken-dialects. In *Proceedings of the 14th International Workshop on Spoken Language Translation*.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019. Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65:675–782.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. In *TACL*.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The iit bombay english-hindi parallel corpus. In *LREC*.

Surafel M. Lakew, Alina Karakanta, Marcello Federico, Matteo Negri, and Marco Turchi. 2019. Adapting multilingual neural machine translation to unseen languages. In *IWSLT*.

Surafel Melaku Lakew, Aliia Erofeeva, and Marcello Federico. 2018. Neural machine translation into language varieties. In *Proceedings of the Third Conference on Machine Translation*.

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

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *ICLR*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *EMNLP*.

Zuchao Li, Hai Zhao, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2020. Reference language based unsupervised neural machine translation. In *Findings of EMNLP*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. In *TACL*.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *EMNLP*.

Nima Pourdamghani and Kevin Knight. 2017. Deciphering related languages. In *EMNLP*.

Ye Qi, Sachan Devendra, Felix Matthieu, Padmanabhan Sarguna, and Neubig Graham. 2018. When and why are pre-trained word embeddings useful for neural machine translation. In *NAACL*.

Raytheon. 2012. Bolt arabic discussion forum parallel training data. In *LDC2012T09*.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In *14th International Workshop on Semantic Evaluation (SemEval)*.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. In *ACL*.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Jorg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *LREC*.

Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. Cross-lingual retrieval for iterative self-supervised training. *arXiv preprint arXiv:2006.09526*.

Yu Wan, Baosong Yang, Derek F. Wong, Lidia S. Chao, Haihua Du, and Ben C.H. Ao. 2020. Unsupervised neural dialect translation with commonality and diversity modeling. In *AAAI*.

Mingxuan Wang, Hongxiao Bai, Hai Zhao, and Lei Li. 2020. Cross-lingual supervision improves unsupervised neural machine translation. In *arXiv*.

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

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. In *ACL*.

Omar F. Zaidan and Chris Callison-Burch. 2011. The arabic online commentary dataset: an annotated dataset of informal arabic with high dialectal content. In *ACL*.