Duplex Sequence-to-Sequence Learning for Reversible Machine Translation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Sequence-to-sequence learning naturally has two directions. How to effectively utilize supervision signals from both directions? Existing approaches either require two separate models, or a multitask-learned model but with inferior performance. In this paper, we propose REDER (**R**E**D**uplex Transform**ER**), a parameter-efficient model and apply it to machine translation. Either end of REDER can simultaneously input and output a distinct language. Thus REDER enables *reversible machine translation* by simply flipping the input and output ends. Experiments verify that REDER achieves the first success of reversible machine translation, which helps outperform multitask-trained baselines by about 1.3 BLEU while obtaining 5.5× speedup during inference.

1 Introduction

Neural sequence-to-sequence (*seq2seq*) learning [Sutskever et al., 2014] has been extensively used in various applications of natural language processing. Standard seq2seq neural networks usually employ the *encoder-decoder* framework, which includes an encoder to acquire the representations from the source side, and a decoder to yield the target side outputs from the encoded source representation [Bahdanau et al., 2015, Gehring et al., 2017, Vaswani et al., 2017].

Generally, given paired training data $\mathcal{D}_{\text{xy}} = \langle \mathcal{X}, \mathcal{Y} \rangle$, where $\mathcal{X}$ is the source side and $\mathcal{Y}$ is target side of the seq2seq task, supervision signals are always *bidirectional*. Thus we can learn the mapping not only from source to target ($f_{\theta_{\text{xy}}} : \mathcal{X} \mapsto \mathcal{Y}$) but also the mapping from target to source ($f_{\theta_{\text{yx}}} : \mathcal{Y} \mapsto \mathcal{X}$). This is very common in many applications. For example, given parallel corpus, we can obtain machine translation models from both Chinese to English and English to Chinese, or given image-sentence pairs, we can train image captioning models [Hossain et al., 2019] or learn text-to-image generation [Qiao et al., 2019].

Typical seq2seq learning utilizes the bidirectional supervisions by splitting the bidirectional supervisions into two unidirectional ones and train two individual seq2seq models on them, respectively. However, such splitting ignores the internal consistency between the bidirectional supervisions, and how to better use the bidirectional supervisions remains an open problem. One potential solution is multitask learning [Johnson et al., 2017], which jointly leverages the bidirectional supervisions within one model and expects the two unidirectional supervisions could boost each other (Figure 1(b)). But it does not work well in our case due to the parameter interference problem [Arivazhagan et al., 2019, Zhang et al., 2021]. For example, in the setting of multitask Chinese-to-English and English-to-Chinese bidirectional translations, the encoder/decoder of seq2seq networks is trained to simultaneously encode/decode both Chinese and English, which always leads to performance drop due to the divergence of the two languages. Another branch of solutions lies in cycle training [Sennrich et al., 2016, He et al., 2016, Zheng et al., 2020], which still include two individual models

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

for separately learning, but explicitly regularize their outputs by the cycle consistency of seq2seq problems. These approaches usually need to introduce extra monolingual data in machine translation.

In this paper, we propose to explore alternative approach for utilizing bidirectional supervisions called *duplex sequence-to-sequence* learning. We argue that current seq2seq networks does not benefit from multitask training because the seq2seq networks are *simplex* from the view of telecommunications[1]. Specifically, the data stream only flows from encoder to decoder in current seq2seq learning, and such simplex property makes the multitask learning suffer from the parameter interfere problem. If we have *duplex* neural networks, the data stream can flow from both ends, each of which only specializes in one languages when learning from bidirectional supervisions (Figure 1(c)), which avoids the parameter interference problem. Thus the bidirectional translation can be achieved as a reversible and unified process. Additionally, we could still use the cycle training idea in the duplex networks while using only one model.

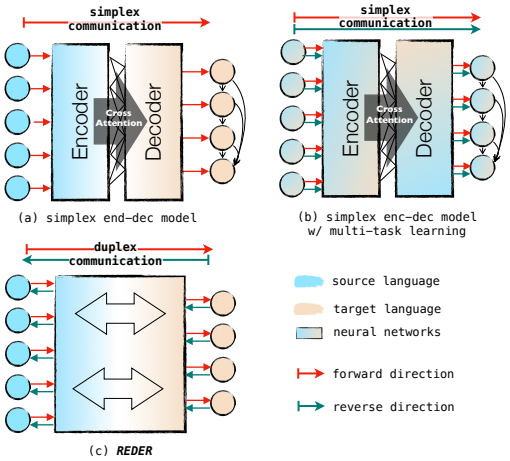

Figure 1: Illustration of different sequence-to-sequence neural models in regards to modeling direction and generation formulations.

Definition 1. *A sequence-to-sequence neural network with parameter θ is duplex if it satisfies the following: its network has two ends, i.e., a source end and a target end; both source and target ends can take input and output sequences; its network defines a forward mapping function $f_\theta^{\rightarrow} : \mathcal{X} \mapsto \mathcal{Y}$, and a reverse mapping function $f_\theta^{\leftarrow} : \mathcal{Y} \mapsto \mathcal{X}$, that satisfies the following reversibility: $f_\theta^{\leftarrow} = (f_\theta^{\rightarrow})^{-1}$ and $f_\theta^{\rightarrow} = (f_\theta^{\leftarrow})^{-1}$; besides, it satisfies the following cycle consistency: $\forall \boldsymbol{x} \in \mathcal{X} : f_\theta^{\leftarrow}(f_\theta^{\rightarrow}(\boldsymbol{x})) = \boldsymbol{x}$ and $\forall \boldsymbol{y} \in \mathcal{Y} : f_\theta^{\rightarrow}(f_\theta^{\leftarrow}(\boldsymbol{y})) = \boldsymbol{y}$.*

Based on the idea of duplex network, we propose REDER[2], the **R**Eversible **D**uplex Transform**ER**, and apply it to machine translation. Note that, building duplex seq2seq networks is non-trivial: a) vanilla encoder-decoder network is irreversible. The decoder's output end cannot take in input signals to exhibit the encoding functionality, and vice versa; b) the topologies of the encoder and decoder are heterogeneous, i.e. decoder works autoregressively, while the encoder is non-autoregressive. Therefore, we design REDER without encoder-decoder devision, where we include the reversible transformer units Gomez et al. [2017] and fully non-autoregressive modeling to solve the above two problems, respectively. Finally, REDER could work in the duplex fashion, thus which could better exploit the bidirectional supervisions for achieving better downstream task performance.

Experimental results show that the duplex idea really works: REDER achieves BLEU scores of 27.50 and 31.25 on standard WMT14 En-De and De-En benchmarks, respectively, which are top results among non-autoregressive machine translation models. multitask learning with does hurt the translation performance of simplex models both in the autoregressive (-0.5 BLEU) and non-autoregressive settings (-1.3 BLEU), but REDER achieve significant gains (+1.3 BLEU) compared to its simplex baseline. Although REDER adopt fully non-autoregressive modeling to realize the duplex networks, the gap of BLEU between REDER and autoregressive Transformer is negiable, while REDER enjoys 5.5× speed-up as bounus and can directly translate between two directions. To our best knowledge, REDER is the first duplex seq2seq network, which enables the first feasible reversible machine translation system.

2 REDER for Reversible Machine Translation

In this section, we introduce how to design a duplex neural seq2seq model, the REDER (**R**Eversible **D**uplex Transform**ER**), that satisfies Definition 1, and its application in machine translation that realizes the first neural reversible machine translation system.

[1]In telecommunications and computer networking, the simplex communication means the communication channel is unidirectional while the duplex communication is bidirectional.

[2]The model name is a *palindrome*, which implies the model works from both ends.

2.1 Challenges of Reversible Machine Translation

Reversible natural language processing [Franck, 1992, Strzalkowski, 1993] and its applications in machine translation [van Noord, 1990] were proposed for the purpose of building machine models that understand and generate natural languages as a reversible, unified process. Despite the success of neural machine translation with deep learning, designing neural architectures for reversible machine translation yet remains under-studied and has the following challenges:

Reversibility. Typical encoder-decoder networks and their neural components, such as Transformer layers, are irreversible, i.e. one cannot just obtain its inverse function by flipping the same encoder-decoder network. To meet our expectation, an inverse function of the network should be derived from the network itself.

Homogeneity. Intuitively, a pair of forward and reverse translation directions should resemble a homogeneous process of understanding and generation. However, typical encoder-decoder networks certainly do not meet such computational homogeneity due to extra cross attention layers in the decoder; and also because of the discrepancy that the decoder works autoregressively but the encoder does non-autoregressively. To meet our expectation, division of encoder and decoder should be no more exist in the desired network.

2.2 The Architecture of REDER

To solve the above challenges, we include two corresponding solutions in REDER to address the reversibility and homogeneity issues respectively, i.e., the reversible duplex Transformer layers, and the symmetric network architecture without encoder-decoder framework.

Figure 2 shows the overall architecture of REDER. As illustrated, REDER has two ends: the source end (left) and the target end (right). θ is the model parameter, shared by both directions. The architecture of REDER is composed of a series of identical Reversible Duplex Transformer layers. When performing the source-to-target mapping $f_\theta^{\rightarrow}$, a source sentence $\boldsymbol{x}$ (blue circles) 1) first transforms to its embedding $\boldsymbol{e}(\boldsymbol{x})$ and enters the source end; 2) then goes through the entire stacked layers and evolves to final representations $\mathbf{H}_L$ which predicts probabilities; 3) finally its target translation (orange circles) will be generated from the target ends. The generation process is fully non-autoregressive.

Likewise, the target-to-source mapping $f_\theta^{\leftarrow}$ is achieved by reversely executing the architecture of REDER from target end to source end. We will dive into the details of the key components of REDER in the following parts.

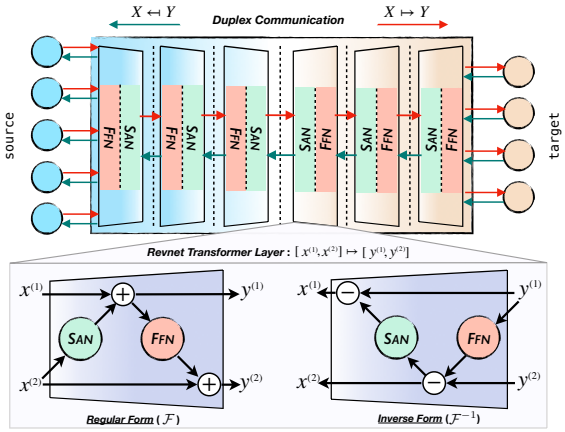

Figure 2: The proposed REDER for duplex sequence-to-sequence generation. The bottom two diagrams show the computation of the regular and reverse forms of a reversible layer. Notice that, to make the whole model symmetric, we reverse the 1-th to $L/2$-th layers, such that the overall computational operations of forward and reverse of REDER are homogeneous.

Reversibility: Reversible Duplex Transformer layers. We adopt the idea of reversible residual network (revnet, Gomez et al., 2017, Kitaev et al., 2020) in the design of the reversible duplex Transformer layer. The bottom of Figure 2 illustrates the forward and reverse computations of a layer. Each layer is composed of a multi-head self-attention (SAN) and a feed-forward network (SAN) with a special reversible design to ensure duplex behavior. Formally, the *regular form* of the l-th layer $\mathcal{F}_l$ performs as follow:

$$\mathbf{H}_l = \mathcal{F}_l(\mathbf{H}_{l-1}) \Leftrightarrow [\mathbf{H}_l^{(1)}; \mathbf{H}_l^{(2)}] = \mathcal{F}_l([\mathbf{H}_{l-1}^{(1)}; \mathbf{H}_{l-1}^{(2)}]),$$
$$\text{where} \quad \mathbf{H}_l^{(1)} = \mathbf{H}_{l-1}^{(1)} + \textsc{San}(\mathbf{H}_{l-1}^{(2)}), \quad \mathbf{H}_l^{(2)} = \mathbf{H}_{l-1}^{(2)} + \textsc{Ffn}(\mathbf{H}_l^{(1)}).$$

The *reverse form* of $\mathcal{F}_l^{-1}$ can be computed by subtracting (instead of adding) the residuals:

$$\mathbf{H}_{l-1} = \mathcal{F}_l^{-1}(\mathbf{H}_l) \Leftrightarrow [\mathbf{H}_{l-1}^{(1)}; \mathbf{H}_{l-1}^{(2)}] = \mathcal{F}_l^{-1}([\mathbf{H}_l^{(1)}; \mathbf{H}_l^{(2)}]),$$
$$\text{where} \quad \mathbf{H}_{l-1}^{(2)} = \mathbf{H}_l^{(2)} - \textsc{Ffn}(\mathbf{H}_{l-1}^{(1)}), \quad \mathbf{H}_{l-1}^{(1)} = \mathbf{H}_l^{(1)} - \textsc{San}(\mathbf{H}_{l-1}^{(2)}).$$

For better modeling the reordering between source and target languages, we employ relative self-attention [Shaw et al., 2018] instead of the original one [Vaswani et al., 2017].

Homogeneity: Symmetric network architecture without encoder-decoder framework. To meet our need to ensure homogeneous network computations for forward and reverse directional tasks, we therefore choose to discard the encoder-decoder paradigm.

Symmetric network. To achieve homogeneous computations, one solution is to make our network symmetric, as depicted at the top of Figure 2. Specifically, we let the 1-th to $L/2$-th layers be the reverse form, whereas the latter $(L/2+1)$-th to L-th be the regular form:

$$f_\theta^{\rightarrow}(\boldsymbol{x}) \triangleq \mathcal{F}_1^{-1} \circ \cdots \circ \mathcal{F}_{L/2}^{-1} \circ \mathcal{F}_{L/2+1} \circ \cdots \circ \mathcal{F}_L(\boldsymbol{x}),$$
$$f_\theta^{\leftarrow}(\boldsymbol{y}) \triangleq \mathcal{F}_L \circ \cdots \circ \mathcal{F}_{L/2+1} \circ \mathcal{F}_{L/2}^{-1} \circ \cdots \circ \mathcal{F}_1^{-1}(\boldsymbol{y}),$$

where $\circ$ means a layer is connected to the next layer. Thereby the forward and reverse computations of REDER become homogeneous: the forward computational operation series reads as a palindrome string $\langle \texttt{fs} \cdots \texttt{fssf} \cdots \texttt{sf} \rangle$ and so does the reverse series, where `s` and `f` denote SAN and FFN.

Fully non-autoregressive modeling. Note that without encoder-decoder division, REDER works in a fully non-autoregressive fashion in both reading and generating sequences. Specifically, given an input sequence $\boldsymbol{x}$, $\mathbf{H}_{0,i} = [\mathbf{H}_{0,i}^{(1)}; \mathbf{H}_{0,i}^{(2)}] = [\boldsymbol{e}(x_i); \boldsymbol{e}(x_i)]$, is the i-th element of REDER's input, which is the concatenation of two copies of the embedding of x_i. Once a forward computation is done, the concatenation of the output of the model $[\mathbf{H}_{L,i}^{(1)}; \mathbf{H}_{L,i}^{(2)}]$ serves as the representations of target translation. And then, a softmax operation is performed to measure the similarity between the model output $[\mathbf{H}_{L,i}^{(1)}; \mathbf{H}_{L,i}^{(2)}]$ and the concatenated embedding of ground-truth reference $[\boldsymbol{e}(y_i), \boldsymbol{e}(y_i)]$, to obtain the prediction probability:

$$p(y_i|\boldsymbol{x};\theta) = \mathrm{softmax}([\boldsymbol{e}(y_i); \boldsymbol{e}(y_i)]^\top [\mathbf{H}_{L,i}^{(1)}; \mathbf{H}_{L,i}^{(2)}]/2).$$

We can likewise derive the procedure of $f_\theta^{\leftarrow}$ for the target-to-source direction. Due to the conditional independence assumption among target tokens introduced by non-autoregressive generation, the log-likelihood of a translation becomes:

$$\log p(\boldsymbol{y}|\boldsymbol{x};\theta) = \sum_i \log p_\theta(y_i|\boldsymbol{x})$$

Modeling variable-length input and output. Encoder-decoder models can easily model variable-length input and output of most seq2seq problems. However, discarding encoder-decoder separation imposes a new challenge: the width of all the layers of the network is depending on the length of the input, thus it is very difficult to allow variable-length input and output, especially when the input is shorter than the output. We resort to the Connectionist Temporal Classification (CTC) [Graves et al., 2006] to solve this problem, a latent alignment approach with superior performance and the flexibility of variable length prediction. Given the conditional independence assumption, CTC is capable of efficiently finding all valid alignments $\boldsymbol{a}$ which derives from the target $\boldsymbol{y}$ by allowing consecutive repetitions and inserting *blank* tokens, and marginalizes log-likelihood:

$$\log p_{\mathrm{ctc}}(\boldsymbol{y}|\boldsymbol{x};\theta) = \log \sum_{\boldsymbol{a} \in \Gamma(\boldsymbol{y})} p_\theta(\boldsymbol{a}|\boldsymbol{x}),$$

where $\Gamma^{-1}(\boldsymbol{a})$ is the collapse function that recovers the target sequence by collapsing consecutive repeated tokens, and then removing all blank tokens. Note that CTC requires that the length of source input should not be smaller than the target output, which is not the case in machine translation. To deal with this, we follow previous useful practice by upsampling the source tokens by 2 times [Saharia et al., 2020, Libovický and Helcl, 2018], and filter those examples when the target lengths are still larger than the one of upsampled source sentences.

Remark. *Reversibility in REDER can be assured in the continuous representation level, where REDER can recover from output representations (last layer) to input embeddings (first layer), which is also the motivation and basis of the auxiliary learning signal, $\mathcal{L}_{\text{fba}}$, in the next section. Reversibility might not hold in the discrete token level, because the irreversible argmax operation discretizes probabilities to tokens. But REDER still shows decent reconstruction capability in practice, as visually depicted in Appendix.*

2.3 Training

Given a parallel corpus and a single model θ, REDER can be jointly supervised by source-to-target and target-to-source translation for $f_\theta^{\rightarrow}$ and $f_\theta^{\leftarrow}$, respectively. Thus both translation directions can be achieved in one REDER model. We refer this to *bidirectional training*, which is opposite to *unidirectional training*, where each translation direction needs a separate model. Moreover, the reversibility of REDER enables appealing potentials to exploit consistency/agreement between forward and reverse directions. We introduce two auxiliary learning signals as follows.

Cycle Consistency Symmetry of a pair of seq2seq tasks enables the use of cycle consistency [He et al., 2016, Cheng et al., 2016a]. Given a source sentence $\boldsymbol{x}$, we first obtain the forward prediction, and then we use the REDER to reconstruct this prediction to the source language:

$$\bar{\boldsymbol{y}} = f_\theta^{\rightarrow}(\boldsymbol{x}), \quad \bar{\boldsymbol{x}} = f_\theta^{\leftarrow}(\bar{\boldsymbol{y}}).$$

Finally, we maximize the consistency or agreement between the original one $\boldsymbol{x}$ and reconstructed one $\bar{\boldsymbol{x}}$. Thus, the loss function reads

$$\mathcal{L}_{\text{cc}}(\boldsymbol{x};\theta) = \texttt{distance}_{\text{cc}}(\boldsymbol{x}, f_\theta^{\leftarrow}(f_\theta^{\rightarrow}(\boldsymbol{x}))) = \texttt{distance}_{\text{cc}}(\boldsymbol{x}, \bar{\boldsymbol{x}}).$$

We expect it can provide an auxiliary signal that a valid prediction should be loyal to reconstruct its source input. Here we use cross-entropy between the probabilistic prediction of the reverse model as distance to measure the consistency.

Layer-wise Forward-Backward Agreement Since REDER is fully reversible, which consists of a series of computationally inverse of intermediate layers, an interesting question arises: given the desired output (i.e., the target sentence), is it possible to derive the desired intermediate hidden representation by the backward target-to-source computation, and encourage the forward source-to-target intermediate hidden representations as close as possible to these "optimal" representations?

Given a source sentence $\boldsymbol{x}$, the inner representations of each layer in forward direction are:

$$\overrightarrow{\mathbf{H}}^1 = \mathcal{F}_1(\boldsymbol{e}(\boldsymbol{x})),\ \overrightarrow{\mathbf{H}}^2 = \mathcal{F}_2(\overrightarrow{\mathbf{H}}^1)\ \ldots\ \overrightarrow{\mathbf{H}}^L = \mathcal{F}_L(\overrightarrow{\mathbf{H}}^{L-1}),$$

and given its corresponding target sequence $\boldsymbol{y}$ as the optimal desired output[3], the inner representations of each layer in reverse direction are:

$$\overleftarrow{\mathbf{H}}^L = \mathcal{F}_L^{-1}(\boldsymbol{e}(y)),\ \overleftarrow{\mathbf{H}}^{L-1} = \mathcal{F}_{L-1}^{-1}(\overleftarrow{\mathbf{H}}^L)\ \ldots\ \overleftarrow{\mathbf{H}}^1 = \mathcal{F}_L^{-1}(\overleftarrow{\mathbf{H}}^2),$$

where $\overrightarrow{\mathbf{H}}^l$ and $\overleftarrow{\mathbf{H}}^l$ represent the representations of l-th layer in forward and reverse models, respectively. As we consider these reverse inner layer representations as "optimal", we try to minimize the cosine distance between the forward and backward corresponding inner layer representations:

$$\mathcal{L}_{\text{fba}}(\boldsymbol{x}|\boldsymbol{y};\theta) = \frac{1}{L}\sum_{l=1}^{L} 1 - cos(\overrightarrow{\mathbf{H}}^l, \texttt{sg}(\overleftarrow{\mathbf{H}}^l)),$$

where $\texttt{sg}(\cdot)$ is stop-gradient operation.

A potential danger of both of the above auxiliary objectives is learning a degenerate solution, where it would probably cheat this task by simply learning an identity mapping. We solve this problem by setting a two-stage training scheme for them, where we first train REDER without using any auxiliary losses until a predefined updates, and then activate the additional losses and continue training the model until convergence.

[3] for CTC-based model where the model prediction are the alignments, we instead extract the token sequence of the best alignment $\boldsymbol{a}^*$, predicted by the model, associated with the ground-truth $\boldsymbol{y}$ as the optimal desired output.

Final Objective Given a parallel dataset $\mathcal{D}_{\text{xy}} = \{(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)} | n = 1...N\}$ of i.i.d observations, the final objective of REDER is to minimize

$$\mathcal{L}(\theta; \mathcal{D}_{\text{xy}}) = \sum_{n=1}^{N} \Big(-\log p_{\text{ctc}}(\boldsymbol{y}^{(n)} | \boldsymbol{x}^{(n)}; \theta) - \log p_{\text{ctc}}(\boldsymbol{x}^{(n)} | \boldsymbol{y}^{(n)}; \theta) + \underbrace{\lambda_{\text{cc}} \mathcal{L}_{\text{cc}}(\boldsymbol{x}^{(n)}; \theta) + \lambda_{\text{cc}} \mathcal{L}_{\text{cc}}(\boldsymbol{y}^{(n)}; \theta)}_{\text{cycle consistencies}} + \underbrace{\lambda_{\text{fba}} \mathcal{L}_{\text{fba}}(\boldsymbol{x}^{(n)} | \boldsymbol{y}^{(n)}; \theta) + \lambda_{\text{fba}} \mathcal{L}_{\text{fba}}(\boldsymbol{y}^{(n)} | \boldsymbol{x}^{(n)}; \theta)}_{\text{layer-wise forward-backward agreements}} \Big)$$

where λ_{cc} and λ_{fba} are coefficients of the auxiliary losses.

3 Related Work

Sequence-to-Sequence Models Exploiting Bidirectional Signals. Several studies try to utilize bidirectionality as constraint to improve sequence-to-sequence tasks such as machine translation [Cheng et al., 2016a,b]. Dual learning [He et al., 2016, Xia et al., 2017] leverage reinforcement learning to interact between two simplex translation models. Later, Xia et al. [2018] propose a partially model-level dual learning that shares some components of both models for forward and reverse task. Zheng et al. [2020] propose to model the two directional translation model with language models in a variational probabilistic framework. These approaches model two directional tasks by setting up two separate simplex models to consider the task bidirectionality. Different from them, REDER can unify direction pair within one duplex model and directly model the bidirectionality at a completely model level. Besides, other studies try to unify two directional tasks by multitask learning [Johnson et al., 2017, Chan et al., 2019] by sharing the same computational process of a single simplex model, which would inevitably result in parameter interference issue where two tasks compete for the limited model capacity. In contrast, REDER formulates both tasks in one model by simply exchanging input and output ends, each of which specialize in a language, thus bidirectional translation becomes a reversible process in which both directions do not need to compete for limited model capacity.

Non-autoregressive Sequence Generation. Non-autoregressive translation (NAT) models [Gu et al., 2018] aims to alliviate decoding inefficiency of traditional autoregressive seq2seq models. Fully NAT models could generate sequence in parallel within only one shot but sacrifice performance [Ma et al., 2019, Shu et al., 2020, Bao et al., 2019, Wei et al., 2019, Qian et al., 2020, Gu and Kong, 2020]. Besides, semi-autoregressive models greatly improve the performance of NAT models, which perform iterative refinement of translations based on previous predictions [Lee et al., 2018, Ghazvininejad et al., 2019, Gu et al., 2019, Kasai et al., 2020, Ghazvininejad et al., 2020]. In this work, REDER takes the advantages of the probabilistic modeling of fully NAT models for resolving the designing challenge of computational homogeneity for both translation directions.

Reversible Neural Architectures. Various reversible neural networks have been proposed for different purposes. On one hand, reversible neural networks help model flexible probability distributions with tractable likelihoods [Dinh et al., 2014, 2017, Kingma et al., 2016], which define a mapping between a simple, known density and a complicated desired density. On the other hand, reversibility can also assist to develop memory-efficient algorithms. The most popular approach is the reversible residual network (revnet, Gomez et al., 2017), which modifies the residual network and allows the activations at any given layer to be recovered from the activations at the following layer. Therefore layers can be reversed one-by-one as back-propagation proceeds from the output of the network to its input. Some follow-up work extends the idea of revnet to RNNs [MacKay et al., 2018] and Transformer [Kitaev et al., 2020] in NLP. We borrow the idea of revnet as the basic unit of our proposed REDER, however, for different purposes that we want to build a duplex seq2seq model to govern two directional tasks reversibly. In this line, van der Ouderaa and Worrall [2019] propose a reversible GAN approach for image-to-image translation in computer vision, which to a certain extent shares the intuition with ours.

4 Experiments

We conduct extensive experiments on standard machine translation benchmarks to inspect REDER's performance on seq2seq tasks. We demonstrate that REDER achieves competitive results, if not better, compared to strong autoregressive (AT) and non-autoregressive (NAT) baselines. REDER is also the

first approach that enables reversible machine translation in one unified model, where bidirectional training with paired translation directions surprisingly helps boost each them with substantial gains.

4.1 Experimental Setup

Datasets. We evaluate our proposal on two standard translation benchmarks, i.e., WMT14 English (EN) ↔ German (DE) (4.5M training pairs), and WMT16 English (EN) ↔ Romanian (RO) (610K training pairs). We apply the same prepossessing steps as mentioned in prior work (EN↔DE: Zhou et al., 2020, EN↔RO: Lee et al., 2018). BLEU [Papineni et al., 2002] is used to evaluate the translation performance for all models.

Knowledge Distillation (KD). Following previous NAT studies [Gu et al., 2018, Zhou et al., 2020], REDERs are trained on distilled data generated from pre-trained auto-regressive Transformer models. The beam size is set to 4 during generation.

Beam Search Decoding. We implement two kinds of inference policies. The first one is parallel decoding that adopts tokens with the highest probability at each position. For CTC-based models, we also implement beam search to REDER with an efficient library of C++ implementation[4].

Implementation Details. We design REDER based on the hyper-parameters of Transformer-*base* [Vaswani et al., 2017]. The number of head is 8, the dimension is 512, and the dimension of FFN is 2048. REDER consists of 12 stacked layers. For both AT and NAT models, we set the dropout rate 0.1 for WMT14 EN↔DE and WMT16 EN↔RO. We adopt weight decay with a decay rate 0.01 and label smoothing with $\epsilon = 0.1$. By default, we upsample the source input by $2\times$ for CTC. We set λ_{cc} and λ_{fba} to 0.1 for all experiments. All models are trained for 300K updates using Nvidia V100 GPUs with a batch size of approximately 64K tokens. Following prior studies [Vaswani et al., 2017], we compute tokenized case-sensitive BLEU. We measure the validation BLEU scores every 2,000 updates, and average the best 5 checkpoints to obtain the final model. As in previous NAT studies, we measure the GPU latency by running the model with a single sentence per batch on a single Nvidia V100 GPU. All models are implemented on `fairseq` [Ott et al., 2019].

4.2 Main Results

As shown in Table 1, we compare REDER with AT and NAT approaches with and without multitask learning, as well as existing approaches that also leverages bidirectional learning signals.

REDER achieves competitive results compared with strong NAT baselines. We show that a unified REDER trained on the same parallel data can simultaneously work in two directions, which has a comparable capability as strong NAT models such as GLAT [Qian et al., 2020] (line 14 vs line 8). With the help of beam search and re-ranking, the performance of REDER can be further boosted to be comparable with the state-of-the-art NAT method GLAT+CTC [Gu and Kong, 2020] (line 14 vs line 9/10). Gu and Kong [2020] explore the best technique combination for NAT, whose tricks can also supplement to enhance REDER. We leave this for exploration.

Duplex learning has more potential than multitask learning and back-translation. Given the same parallel corpus as training data, multitask learning (MTL) yields considerable performance degradation of either AT (line 1 vs line 2). By re-implementing GLAT+CTC [Gu and Kong, 2020] (line 10/11) as the strong NAT competitor for more convincing comparison. We observe that MTL would hurt more severely for NAT models, such as GLAT+CTC models (line 11 vs line 10), and REDER as well (line 13 vs line 12). These results verify our concern of multitask-learned models regarding parameter interference. Meanwhile, when no external monolingual resources available, back-translation (BT) only adds mild points from training data. In contrast, duplex learning allows REDER gain more benefits, becoming a better alternative and a parameter-efficient choice to exploit more potentials from the provided parallel data. Plus, duplex learning is orthogonal to BT, which could further improve REDER with monolingual data.

REDER performs on par with autoregrassive Transformer. Despite the challenge in regards to non-autoregressive modeling and non-encoder-decoder design, REDER closely approaches the simplex AT models (line 14 vs line 1), while REDER can translate both direction in one model. Besides, REDER even surpasses multitask-learned bidirectional autoregressive Transformer model

[4]https://github.com/parlance/ctcdecode

Table 1: Comparisons between our models and existing models. The speed-up is measured on the WMT14 EN-DE test set. All NAT models are trained with KD. "↔": whether to allow bidirectional translation. "MTL": multitask learning. "BT": back-translation. All our implemented CTC-based models employ beam search decoding with beam size of 20.

	Systems	↔	$\|\theta\|$	Speed	WMT14 EN-DE	WMT14 DE-EN	WMT16 EN-RO	WMT16 RO-EN
AT	① Transformer-*base* (KD teacher)	✗	62M× 2	1.0×	27.60	31.50	33.85	33.70
	② w/ *MTL*	✓	62M	1.0×	27.06	30.96	-	-
	③ w/ *BT*	✗	62M×2	1.0×	27.82	31.91	-	-
NAT	④ vanilla NAT [Gu et al., 2018]	✗	62M×2	15.6×	17.69	21.47	27.29	29.06
	⑤ CTC w/o KD [Libovický and Helcl, 2018]	✗	58M×2	-	16.56	18.64	19.54	24.67
	⑥ *CTC* [Saharia et al., 2020]	✗	58M×2	18.6×	25.70	28.10	32.20	31.60
	⑦ *Imputer* [Saharia et al., 2020]	✗	58M×2	18.6×	25.80	28.40	32.30	31.70
	⑧ *GLAT* [Qian et al., 2020]	✗	62M×2	15.3×	26.55	31.02	32.87	33.51
	⑨ *GLAT+CTC* [Gu and Kong, 2020]	✗	62M×2	16.8×	27.20	**31.39**	**33.71**	**34.16**
	our re-implementations of Gu and Kong [2020]:							
	⑩ *GLAT+CTC*	✗	62M×2	16.2×	26.79	30.45	-	-
	⑪ w/ *MTL*	✓	62M	16.2×	25.50	29.49	-	-
REDER	⑫ *simplex* REDER	✗	58M×2	5.5×	26.20	30.02	32.67	32.98
	⑬ w/ *MTL*	✓	58M	5.5×	25.58	29.12	-	-
	⑭ *duplex* REDER (final model)	✓	58M	5.5×	**27.50**	31.25	33.60	34.03
previous studies	Reformer [Kitaev et al., 2020]	✗	62M×2	-×	27.60	-	-	-
	Model-level DL *big* [Xia et al., 2018]	✗	210M×2	-	28.90	31.90	-	-
	KERMIT [Chan et al., 2019]	✗	124M	-	25.60	27.40	-	-
	KERMIT + mono [Chan et al., 2019]	✗	124M	-	28.10	28.60	-	-
	MGNMT [Zheng et al., 2020]	✓	195M	-	27.70	31.40	32.70	33.90

(line 14 vs line 2). Additionally, REDER enjoys a faster decoding spend than AT models. This evidence shows the advantage and practical value of reversible machine translation as a more decent solution for parameter-effiecient bidirectional translation systems.

Comparison with existing approaches. Reformer [Kitaev et al., 2020] also employs revnet to make parts of the Transformer model to reduce the memory consumption for training, while REDER is fully reversible with a different motivation of maximizing the use of bidirectional signals. Existing simplex approaches exploiting bidirectional signals require two separate simplex models for both directions [Xia et al., 2018, Zheng et al., 2020]. REDER, in contrast, only needs one unified duplex model and coherently models two directions. Alternatively, Chan et al. [2019] use a same simplex network to achieve bidirectional translation via multitask learning, needing to split limited capacity for both directions, which underperform REDER on parallel settings.

4.3 Ablation Study of Model Design

REDER is developed on the top of various components in terms of data (knowledge distillation), learning objective (CTC), architecture (revnet, relative attention), and auxiliary losses endowed by reversibility of REDER. We analyze their effects through various combinations in Table 2. We first consider training REDER for a single direction to seek the best practice to run REDER for sequence-to-sequence tasks. KD and CTC are essential to training REDER, as suggested by previous NAT studies [Saharia et al., 2020, Gu and Kong, 2020]. Meanwhile, we notice the benefit of relative self-attention. We therefore use these three techniques by default for all of the proposed models. As for the duplex variants of models that learn both directions simultaneously, they can further improve the translation accuracy by substantial margins. These results verify our motivation that the paired translation directions could be better learned in a unified reversible model. Reversibility enables us to utilize layer-wise forward-backward agreement and cycle consistency, which are also shown to boost improvement considerably.

Table 2: Ablation on WMT14 EN→DE test set with different combinations of techniques. R-SAN denotes relative self attention. For fair comparison, all CTC and non-CTC variants do not use beam search decoding.

KD	CTC	revnet	R-SAN	$\mathcal{L}_{\text{cc}}$	$\mathcal{L}_{\text{fba}}$	BLEU
						11.40
✓						19.50
	✓					16.90
✓	✓					25.01
✓	✓		✓			25.55
✓	✓		✓		✓	25.90
✓	✓	✓	✓			26.20
✓	✓	✓	✓	✓		26.70
✓	✓	✓	✓		✓	26.65
✓	✓	✓	✓	✓	✓	**26.89**

4.4 Decoding: Effect of Beam search and Re-ranking

The performance of REDERs can be further boosted with additional (beam-search or re-ranking) techniques. For CTC beam search, we use the teacher model (AT *base*) to re-rank the translation candidates obtained by the beam search to determine the one with the best quality. As shown in Table 3, a larger beam size results in a smaller BP for AT models, meaning it produces shorter translations [Stahlberg and Byrne, 2019, Eikema and Aziz, 2020]. CTC beam search helps REDER produce longer outputs (larger BPs) but only endows a little improvement. With beam search and AT reranking, REDER can generate more decent translations. These results imply that we need to find a better way to train REDER (and probably the NAT family) if we do not want to involve an extra AT for such a somewhat inconvenient re-ranking.

Table 3: Comparisons regarding decoding methods for REDER on WMT14 EN↔DE. The brevity penalty (BP) given by BLEU indicates the adequacy of translation: the lower the BP, the more inadequate the translation.

Systems	EN-DE	DE-EN	BP	Speed
Transformer (AT, teacher)	27.20	31.00	0.980	1.0 ×
+ *beam=5*	27.60	31.50	0.998	-
+ *beam=20*	27.65	31.12	0.954	-
REDER w/ parallel decoding	26.89	30.90	0.935	19.8 ×
+ *beam=20*	26.90	30.75	0.995	6.8 ×
+ *beam=20* + AT reranking	27.50	31.25	1.000	5.5 ×
+ *beam=100*	26.92	30.95	0.982	2.1 ×
+ *beam=100* + AT reranking	27.59	31.45	1.000	1.2 ×

4.5 Training: Impact of Knowledge Distillation

Like other NAT approaches, we find that REDER heavily relies on knowledge distillation. We report the performance of models trained on raw data and distilled data generated from AT models in Table 4. As we can see, without KD, the accuracy of REDER significantly drops. Then, we aim to explore the most proper way to integrate KD data. We observe that if we only use KD data of one direction (only target-side data are KD'ed, e.g., German sentences in EN-DE), it only mostly benefits a single direction. These results imply that we need to provide KD data of both directions to train REDER, and we then try to figure how to do so. We notice that if we mix the KD data of both directions by concatenating them directly, it somehow improves results compared to the policy only using single-direction KD data. Finally, we find the best way is to separately feed KD data in accordance to directions, i.e., feeding EN-DE KD data when training the EN-DE direction and providing DE-EN KD data when training the reverse direction DE-EN.

Table 4: Performance regarding KD on WMT14 EN↔DE. #data means the amount of data points for each direction.

Systems	EN-DE	DE-EN	#data
all raw	17.85	19.68	N/N
EN→DE KD (only DE distillated)	25.49	26.57	N/N
DE→EN KD (only EN distillated)	23.04	28.82	N/N
mixture KD	26.50	29.65	$2N/2N$
separate KD (final model)	27.50	31.25	N/N

5 Conclusion and Future Work

In this paper, we propose REDER, the **RE**versible **D**uplex Transform**ER** for sequence-to-sequence problem and apply it to machine translation that for the first time shows the feasibility of a reversible machine translation system. REDER is a fully reversible model that can transform one sequence to the other one forth and back, by reading and generating through its two ends. We verify our our motivation and the effectiveness of REDER on several widely-used NMT benchmarks, where REDER shows appealing performance over strong baselines.

As for promising future directions, REDER can be applied to monolingual, multilingual and zero-shot settings, thanks to the fact that each "end" of REDER specializes a language. For instance, given trained REDERs $\mathcal{M}_{\mathrm{En}\leftrightarrow\mathrm{De}}$ and $\mathcal{M}_{\mathrm{En}\leftrightarrow\mathrm{Ja}}$, we combine last half layers (the DE end) of $\mathcal{M}_{\mathrm{En}\leftrightarrow\mathrm{De}}$ and the JA end of $\mathcal{M}_{\mathrm{En}\leftrightarrow\mathrm{Ja}}$ to obtain a zero-shot $\mathcal{M}_{\mathrm{De}\leftrightarrow\mathrm{Ja}}$, translating between German and Japanese. Likewise, composition of an English end and its reverse results in $\mathcal{M}_{\mathrm{En}\leftrightarrow\mathrm{En}}$, which can learn from monolingual data like an autoencoder. This compositional fashion resembles the LEGO, which manipulates only a linear number of language ends. Therefore, while adding a new language to a multilingual REDER system (in a form of composition of ends of involved languages), we would probably not need to re-train the whole system like we do for a current multilingual NMT system, which reduces the difficulty and cost to train, deploy and maintain a large scale multilingual translation system.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*, 2019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Yu Bao, Hao Zhou, Jiangtao Feng, Mingxuan Wang, Shujian Huang, Jiajun Chen, and Lei Li. Non-autoregressive transformer by position learning. *arXiv preprint arXiv:1911.10677*, 2019.

William Chan, Nikita Kitaev, Kelvin Guu, Mitchell Stern, and Jakob Uszkoreit. Kermit: Generative insertion-based modeling for sequences. *arXiv preprint arXiv:1906.01604*, 2019.

Yong Cheng, Shiqi Shen, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Agreement-based joint training for bidirectional attention-based neural machine translation. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2761–2767. IJCAI/AAAI Press, 2016a. URL http://www.ijcai.org/Abstract/16/392.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1965–1974, 2016b.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Bryan Eikema and Wilker Aziz. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.398. URL https://www.aclweb.org/anthology/2020.coling-main.398.

Gordon Franck. Reversible grammars and natural language processing. In *Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing: Technological Challenges of the 1990's*, page 102–109, New York, NY, USA, 1992. Association for Computing Machinery. ISBN 089791502X. doi: 10.1145/143559.143597. URL https://doi.org/10.1145/143559.143597.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR, 2017. URL http://proceedings.mlr.press/v70/gehring17a.html.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. 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 6112–6121, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1633. URL https://www.aclweb.org/anthology/D19-1633.

Marjan Ghazvininejad, Omer Levy, and Luke Zettlemoyer. Semi-autoregressive training improves mask-predict decoding. *arXiv preprint arXiv:2001.08785*, 2020.

Aidan N. Gomez, Mengye Ren, Raquel Urtasun, and Roger B. Grosse. The reversible residual network: Backpropagation without storing activations. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2214–2224, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f9be311e65d81a9ad8150a60844bb94c-Abstract.html.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In William W. Cohen and Andrew W. Moore, editors, *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 369–376. ACM, 2006. doi: 10.1145/1143844.1143891. URL https://doi.org/10.1145/1143844.1143891.

Jiatao Gu and Xiang Kong. Fully non-autoregressive neural machine translation: Tricks of the trade. *arXiv preprint arXiv:2012.15833*, 2020.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=B1l8BtlCb.

Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein transformer. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11179–11189, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/675f9820626f5bc0afb47b57890b466e-Abstract.html.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 820–828, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/5b69b9cb83065d403869739ae7f0995e-Abstract.html.

MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, 51(6):1–36, 2019.

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. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. doi: 10.1162/tacl_a_00065. URL https://www.aclweb.org/anthology/Q17-1024.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. Non-autoregressive machine translation with disentangled context transformer. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5144–5155. PMLR, 2020. URL http://proceedings.mlr.press/v119/kasai20a.html.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems*, 29:4743–4751, 2016.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1149. URL https://www.aclweb.org/anthology/D18-1149.

Jindřich Libovický and Jindřich Helcl. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1336. URL https://www.aclweb.org/anthology/D18-1336.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. 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 4282–4292, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1437. URL https://www.aclweb.org/anthology/D19-1437.

Matthew MacKay, Paul Vicol, Jimmy Ba, and Roger B. Grosse. Reversible recurrent neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9043–9054, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/4ff6fa96179cdc2838e8d8ce64cd10a7-Abstract.html.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL https://www.aclweb.org/anthology/N19-4009.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. Glancing transformer for non-autoregressive neural machine translation. *arXiv preprint arXiv:2008.07905*, 2020.

Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1505–1514, 2019.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. Non-autoregressive machine translation with latent alignments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1098–1108, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.83. URL https://www.aclweb.org/anthology/2020.emnlp-main.83.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://www.aclweb.org/anthology/P16-1009.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074. URL https://www.aclweb.org/anthology/N18-2074.

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *EMNLP*, 2020.

Felix Stahlberg and Bill Byrne. On NMT search errors and model errors: Cat got your tongue? 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 3356–3362, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1331. URL https://www.aclweb.org/anthology/D19-1331.

Tomek Strzalkowski. *Reversible Grammar in Natural Language Processing*, volume 255. Springer Science & Business Media, 1993.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html.

Tycho FA van der Ouderaa and Daniel E Worrall. Reversible gans for memory-efficient image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4720–4728, 2019.

Gertjan van Noord. Reversible unification based machine translation. In *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*, 1990. URL https://www.aclweb.org/anthology/C90-2052.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, and Xu Sun. Imitation learning for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1304–1312, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1125. URL https://www.aclweb.org/anthology/P19-1125.

Yingce Xia, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. Dual supervised learning. In *International Conference on Machine Learning*, pages 3789–3798. PMLR, 2017.

Yingce Xia, Xu Tan, Fei Tian, Tao Qin, Nenghai Yu, and Tie-Yan Liu. Model-level dual learning. In *International Conference on Machine Learning*, pages 5383–5392. PMLR, 2018.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Wj4ODo0uyCF.

Zaixiang Zheng, Hao Zhou, Shujian Huang, Lei Li, Xin-Yu Dai, and Jiajun Chen. Mirror-generative neural machine translation. In *International Conference on Learning Representations*, 2020.

Chunting Zhou, Jiatao Gu, and Graham Neubig. Understanding knowledge distillation in non-autoregressive machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=BygFVAEKDH.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]