Self-supervised and Supervised Joint Training for Resource-rich Machine Translation

Yong Cheng[1]   Wei Wang[†]   Lu Jiang[1 2]   Wolfgang Macherey[1]

Abstract

Self-supervised pre-training of text representations has been successfully applied to low-resource Neural Machine Translation (NMT). However, it usually fails to achieve notable gains on resource-rich NMT. In this paper, we propose a joint training approach, F_2-*XEnDec*, to combine self-supervised and supervised learning to optimize NMT models. To exploit complementary self-supervised signals for supervised learning, NMT models are trained on examples that are interbred from monolingual and parallel sentences through a new process called crossover encoder-decoder. Experiments on two resource-rich translation benchmarks, WMT'14 English-German and WMT'14 English-French, demonstrate that our approach achieves substantial improvements over several strong baseline methods and obtains a new state of the art of 46.19 BLEU on English-French when incorporating back translation. Results also show that our approach is capable of improving model robustness to input perturbations such as code-switching noise which frequently appears on social media.

1. Introduction

Self-supervised pre-training of text representations (Peters et al., 2018; Radford et al., 2018) has achieved tremendous success in natural language processing applications. Inspired by BERT (Devlin et al., 2019), recent works attempt to leverage sequence-to-sequence model pre-training for Neural Machine Translation (NMT) (Lewis et al., 2019; Song et al., 2019; Liu et al., 2020b). Generally, these methods comprise two stages: pre-training and finetuning. During the pre-training stage, the model is learned with a self-

supervised task on abundant unlabeled data (*i.e.* monolingual sentences). In the second stage, the full or partial model is finetuned on a downstream translation task of labeled data (*i.e.* parallel sentences). Studies have demonstrated the benefit of pre-training for the low-resource translation task in which the labeled data is limited (Lewis et al., 2019; Song et al., 2019). All these successes share the same setup: pre-training on abundant unlabeled data and finetuning on limited labeled data.

In many NMT applications, we are confronted with a different setup where abundant labeled data, *e.g.*, millions of parallel sentences, are available for finetuning. For these resource-rich translation tasks, the two-stage approach is less effective and, even worse, sometimes can undermine the performance if improperly utilized (Zhu et al., 2020), in part due to the catastrophic forgetting (French, 1999). More recently, several mitigation techniques have been proposed for the two-stage approach (Edunov et al., 2019; Yang et al., 2019; Zhu et al., 2020), such as freezing the pre-trained representations during finetuning. However, these strategies hinder uncovering the full potential of self-supervised learning since the learned representations are either held fixed or slightly tuned in the supervised learning.

In this paper, we study resource-rich machine translation through a different perspective of joint training where, in contrast to the conventional two-stage approaches, we train NMT models in a single stage using the self-supervised objective (on monolingual sentences) in addition to the supervised objective (on parallel sentences). The challenge for this single-stage training paradigm is that self-supervised learning is less useful in joint training because it provides a much weaker learning signal that can be easily dominated by the supervised learning signal in joint training. As a result, conventional approaches such as combining self-supervised and supervised learning objectives perform not much better than the supervised learning objective by itself.

This paper aims at exploiting the complementary signals in self-supervised learning to facilitate supervised learning. Inspired by chromosomal crossovers (Rieger et al., 2012), we propose an essential new task called crossover encoder-decoder (or *XEnDec*) which takes two training examples as inputs (called parents), shuffles their source sentences,

and produces a sentence by a mixture decoder model. Our method applies *XEnDec* to "deeply" fuse the monolingual (unlabeled) and parallel (labeled) sentences, thereby producing their first and second filial generation (or F_1 and F_2 generation). As we find that the F_2 generation exhibits combinations of traits that differ from those found in the monolingual or the parallel sentence, we train NMT models on the F_2 offspring and name our method F_2-*XEnDec*.

To the best of our knowledge, the proposed method is among the first NMT models on joint self-supervised and supervised learning, and moreover, the first to demonstrate such joint learning substantially benefits resource-rich machine translation. Compared to recent two-stage finetuning approaches (Zhu et al., 2020) and (Yang et al., 2019), our method only needs a single training stage to utilize the complementary signals in self-supervised learning. Empirically, our results show the proposed single-stage approach achieves comparable or better results than previous methods. In addition, our method improves the robustness of NMT models which is known as a critical deficiency in contemporary NMT systems (cf. Section 4.3). It is noteworthy that none of the two-stage training approaches have ever reported this behavior.

We empirically validate our approach on the WMT'14 English-German and WMT'14 English-French translation benchmarks which yields an improvement of 2.13 and 1.78 BLEU points over the vanilla Transformer model (Ott et al., 2018), respectively. It achieves a new state of the art of 46.19 BLEU on the WMT'14 English-French translation task with the back translation technique. In summary, our contributions are as follows:

1. We propose a crossover encoder-decoder (*XEnDec*) which, with appropriate inputs, can reproduce several existing self-supervised and supervised learning objectives.

2. We jointly train self-supervised and supervised objectives in a single stage, and show that our method is able to exploit the complementary signals in self-supervised learning to facilitate supervised learning.

3. Our approach achieves significant improvements on resource-rich translation tasks and exhibits higher robustness against input perturbations such as code-switching noise.

2. Background

2.1. Neural Machine Translation

Under the encoder-decoder paradigm (Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017), the conditional probability $P(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})$ of a target-language sen-

tence $\mathbf{y} = y_1, \cdots, y_J$ given a source-language sentence $\mathbf{x} = x_1, \cdots, x_I$ is modeled as follows: The encoder maps the source sentence $\mathbf{x}$ onto a sequence of I word embeddings $e(\mathbf{x}) = e(x_1), ..., e(x_I)$. Then the word embeddings are encoded into their corresponding continuous hidden representations. The decoder acts as a conditional language model that reads embeddings $e(\mathbf{y})$ for a shifted copy of $\mathbf{y}$ along with the aggregated contextual representations $\mathbf{c}$. For clarity, we denote the input and output in the decoder as $\mathbf{z}$ and $\mathbf{y}$, *i.e.*, $\mathbf{z} = \langle s \rangle, y_1, \cdots, y_{J-1}$, where $\langle s \rangle$ is a start symbol. Conditioned on an aggregated contextual representation $\mathbf{c}_j$ and its partial target input $\mathbf{z}_{\leq j}$, the decoder generates $\mathbf{y}$ as:

$$P(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}) = \prod_{j=1}^{J} P(y_j|\mathbf{z}_{\leq j}, \mathbf{c}; \boldsymbol{\theta}). \tag{1}$$

The aggregated contextual representation $\mathbf{c}$ is often calculated by summarizing the sentence $\mathbf{x}$ with an attention mechanism (Bahdanau et al., 2015). A byproduct of the attention computation is a noisy alignment matrix $\mathbf{A} \in \mathbb{R}^{J \times I}$ which roughly captures the translation correspondence between target and source words (Garg et al., 2019).

Generally, NMT optimizes the model parameters $\boldsymbol{\theta}$ by minimizing the empirical risk over a parallel training set $(\mathbf{x}, \mathbf{y}) \in \mathcal{S}$:

$$\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{(\mathbf{x}, \mathbf{y}) \in \mathcal{S}} [\ell(f(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}), h(\mathbf{y}))], \tag{2}$$

where ℓ is the cross entropy loss between the model prediction $f(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})$ and $h(\mathbf{y})$, and $h(\mathbf{y})$ denotes the sequence of one-hot label vectors with label smoothing in the Transformer (Vaswani et al., 2017).

2.2. Pre-training for Neural Machine Translation

Pre-training sequence-to-sequence models for language generation has been shown to be effective for machine translation (Song et al., 2019; Lewis et al., 2019). These methods generally comprise two stages: pre-training and finetuning. The pre-training takes advantage of an abundant monolingual corpus $\mathcal{U} = \{\mathbf{y}\}$ to learn representations through a self-supervised objective called denoising autoencoder (Vincent et al., 2008) which aims at reconstructing the original sentence $\mathbf{y}$ from one of its corrupted counterparts.

Let $n(\mathbf{y})$ be a corrupted copy of $\mathbf{y}$ where the function $n(\cdot)$ adds noise and/or masks words. $(n(\mathbf{y}), \mathbf{y})$ constitutes the pseudo parallel data and is fed into the NMT model to compute the reconstruction loss. The self-supervised reconstruction loss over the corpus $\mathcal{U}$ is defined as:

$$\mathcal{L}_{\mathcal{U}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\mathbf{y} \in \mathcal{U}} [\ell(f(n(\mathbf{y}), \mathbf{y}; \boldsymbol{\theta}), h(\mathbf{y}))], \tag{3}$$

The optimal model parameters $\boldsymbol{\theta}^{\star}$ are learned via the self-supervised loss $\mathcal{L}_{\mathcal{U}}(\boldsymbol{\theta})$ and used to initialize downstream
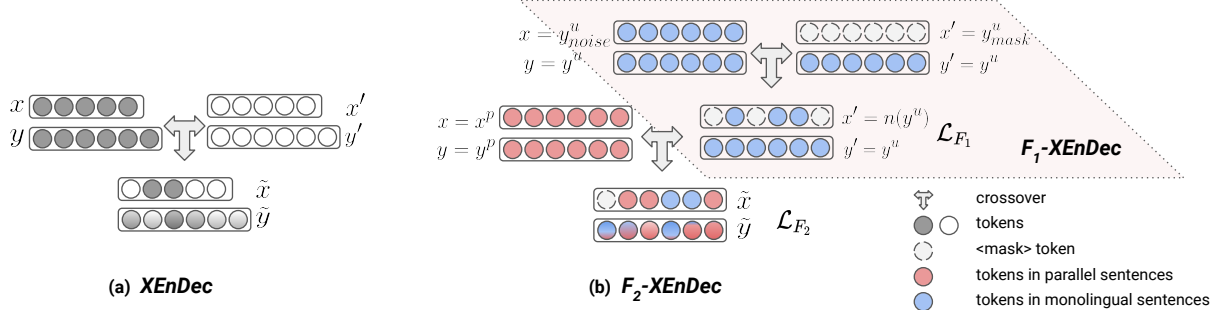
Figure 1. **(a)** Illustration of crossover encoder-decoder (*XEnDec*). It takes two training examples $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$ as inputs, and outputs a sentence pair $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$. **(b)** Our method applies *XEnDec* to fuse the monolingual (blue) and parallel sentences (red). In the first generation, F_1-*XEnDec* generates $(n(\mathbf{y}^u), \mathbf{y}^u)$ incurring a self-supervised loss $\mathcal{L}_{F_1}$, where $(n(\mathbf{y}^u)$ is the function discussed in Section 2.2 that corrupts the monolingual sentence $\mathbf{y}^u$. F_2-*XEnDec* applies another round of *XEnDec* to incorporate parallel data $(\mathbf{x}^p, \mathbf{y}^p)$ to get the F_2 output $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$. $\mathbf{y}^u$: a monolingual sentence. $\mathbf{y}^u_{noise}$: a sentence generated by adding non-masking noise to $\mathbf{y}^u$. $\mathbf{y}^u_{mask}$: a sentence of length $|\mathbf{y}^u|$ containing only "$\langle mask \rangle$" tokens.

models during the finetuning on the parallel training set $\mathcal{S}$.

3. Cross-breeding: F_2-*XEnDec*

For resource-rich translation tasks in which a large parallel corpus and (virtually) unlimited monolingual corpora are available, our goal is to improve translation performance by exploiting self-supervised signals to complement the supervised learning. In the proposed method, we train NMT models jointly with supervised and self-supervised learning objectives in a single stage. This is based on an essential task called *XEnDec*. In the remainder of this section, we first detail the *XEnDec* and then introduce our approach and present the overall algorithm. Finally, we discuss its relationship to some of the previous works.

3.1. Crossover Encoder-Decoder

This section introduces the crossover encoder-decoder (*XEn-Dec*). Different from a conventional encoder-decoder, *XEn-Dec* takes two training examples as inputs (called parents), shuffles the parents' source sentences and produces a virtual example (called offspring) through a mixture decoder model. Fig. 1(a) illustrates this process.

Formally, let $(\mathbf{x}, \mathbf{y})$ denote a training example where $\mathbf{x} = x_1, \cdots, x_I$ represents a source sentence of I words and $\mathbf{y} = y_1, \cdots, y_J$ is the corresponding target sentence of J words. In supervised training, $\mathbf{x}$ and $\mathbf{y}$ are parallel sentences. As we will see in Section 3.2, *XEnDec* can be carried out with and without supervision. We do not distinguish these cases for now and use generic notations to illustrate the idea.

Given a pair of examples $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$ called parents, the crossover encoder shuffles the two source sequences into

a new source sentence $\tilde{\mathbf{x}}$, calculated from:

$$\tilde{x}_i = m_i x_i + (1 - m_i) x'_i, \tag{4}$$

where $\mathbf{m} = m_1, \cdots, m_I \in \{0, 1\}^I$ stands for a series of Bernoulli random variables with each taking the value 1 with probability p called shuffling ratio. If $m_i = 0$, then the i-th word in $\mathbf{x}$ will be substituted with the word in $\mathbf{x}'$ at the same position. For convenience, the lengths of the two sequences are aligned by appending padding tokens to the end of the shorter sentence.

The crossover decoder employs a mixture model to generate the virtual target sentence. The embedding of the decoder's input $\tilde{\mathbf{z}}$ is computed as:

$$e(\tilde{z}_j) = \frac{1}{Z} \Big[e(y_{j-1}) \sum_{i=1}^{I} A_{(j-1)i} m_i \\ + e(y'_{j-1}) \sum_{i=1}^{I} A'_{(j-1)i} (1 - m_i) \Big], \tag{5}$$

where $e(\cdot)$ is the embedding function. $Z = \sum_{i=1}^{I} A_{(j-1)i} m_i + A'_{(j-1)i} (1 - m_i)$ is the normalization term where $\mathbf{A}$ and $\mathbf{A}'$ are the alignment matrices for the source sequences $\mathbf{x}$ and $\mathbf{x}'$, respectively. Eq. (5) averages embeddings of $\mathbf{y}$ and $\mathbf{y}'$ through the latent weights computed by $\mathbf{m}$, $\mathbf{A}$, and $\mathbf{A}'$. The alignment matrix measures the contribution of the source words for generating a specific target word (Och & Ney, 2004; Bahdanau et al., 2015). For example, A_{ji} represents the contribution score of the i-th word in the source sentence for the j-th word in the target sentence. For simplicity, this paper uses the attention matrix learned in the NMT model as a noisy alignment matrix (Garg et al., 2019).

Likewise, the label vector for the crossover decoder is calculated from:

$$h(\tilde{y}_j) = \frac{1}{Z} \Big[h(y_j) \sum_{i=1}^{I} A_{ji} m_i$$
$$+ h(y'_j) \sum_{i=1}^{I} A'_{ji} (1 - m_i) \Big], \tag{6}$$

The $h(\cdot)$ function projects a word onto its label vector, *e.g.*, a one-hot vector. The loss of *XEnDec* is computed over its output $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ using the negative log-likelihood:

$$\ell(f(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}; \boldsymbol{\theta}), h(\tilde{\mathbf{y}})) = -\log P(\tilde{\mathbf{y}} | \tilde{\mathbf{x}}; \boldsymbol{\theta})$$
$$= \sum_j KL(h(\tilde{y}_j) \| P(y | \tilde{\mathbf{z}}_{\leq j}, \mathbf{c}_j; \boldsymbol{\theta})), \tag{7}$$

where $\tilde{\mathbf{z}}$ is a shifted copy of $\tilde{\mathbf{y}}$ as discussed in Section 2.1. Notice that even though we do not directly observe the "virtual sentences" $\tilde{\mathbf{z}}$ and $\tilde{\mathbf{y}}$, we are still able to compute the loss using their embeddings and labels. In practice, the length of $\tilde{\mathbf{x}}$ is set to $\max(|\mathbf{x}|, |\mathbf{x}'|)$ whereas $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{z}}$ share the same length of $\max(|\mathbf{y}|, |\mathbf{y}'|)$.

3.2. Training

The proposed method applies *XEnDec* to deeply fuse the parallel data $\mathcal{S}$ with nonparallel, monolingual data $\mathcal{U}$. As illustrated in Fig. 1(b), the first generation (F_1-*XEnDec* in the figure) uses *XEnDec* to combine monolingual sentences of different views, thereby incurring a self-supervised loss $\mathcal{L}_{F_1}$. We compute the loss $\mathcal{L}_{F_1}$ using Eq. (3). Afterward, the second generation (F_2-*XEnDec* in the figure) applies *XEnDec* to the offspring of the first generation $(n(\mathbf{y}^u), \mathbf{y}^u)$ and a sampled parallel sentence $(\mathbf{x}^p, \mathbf{y}^p)$, yielding a new loss term $\mathcal{L}_{F_2}$. The loss $\mathcal{L}_{F_2}$ is computed over the output of the F_2-*XEnDec* by:

$$\mathcal{L}_{F_2}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\mathbf{y}^u \in \mathcal{U}} \mathop{\mathbb{E}}_{(\mathbf{x}^p, \mathbf{y}^p) \in \mathcal{S}} [\ell(f(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}; \boldsymbol{\theta}), h(\tilde{\mathbf{y}}))], \tag{8}$$

where $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ is the output of the F_2-*XEnDec* in Fig. 1(b).

The final NMT models are optimized jointly on the original translation loss and the above two auxiliary losses.

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) + \mathcal{L}_{F_1}(\boldsymbol{\theta}) + \mathcal{L}_{F_2}(\boldsymbol{\theta}), \tag{9}$$

$\mathcal{L}_{F_2}$ in Eq. (9) is used to deeply fuse monolingual and parallel sentences at instance level rather than combine them mechanically. Section 4.4 empirically verifies the contributions of the $\mathcal{L}_{F_1}$ and $\mathcal{L}_{F_2}$ loss terms.

Algorithm 1 delineates the procedure to compute the final loss $\mathcal{L}(\boldsymbol{\theta})$. Specifically, each time, we sample a monolingual sentence for each parallel sentence to circumvent the expensive enumeration in Eq. (8). To speed up the training,

---

Algorithm 1 Proposed F_2-XEnDec function.

Input: Parallel corpus $\mathcal{S}$, Monolingual corpus $\mathcal{U}$, and Shuffling ratios p_1 and p_2
Output: Batch Loss $\mathcal{L}(\boldsymbol{\theta})$.

1 **Function** F_2-$XEnDec$($\mathcal{S}, \mathcal{U}, p_1, p_2$):
2   **foreach** $(\mathbf{x}^p, \mathbf{y}^p) \in \mathcal{S}$ **do**
3   Sample a $\mathbf{y}^u \in \mathcal{U}$ with similar length as $\mathbf{x}^p$; `// done offline.`
4     $\mathbf{y}^u_{noise} \leftarrow$ add non-masking noise to $\mathbf{y}^u$;
5     $(n(\mathbf{y}^u), \mathbf{y}^u) \leftarrow XEnDec$ over the inputs $(\mathbf{y}^u_{noise}, \mathbf{y}^u)$ and $(\mathbf{y}^u_{mask}, \mathbf{y}^u)$, with the shuffling ratio p_1 and arbitrary alignment matrices;
6     $\mathcal{L}_{\mathcal{S}} \leftarrow$ compute ℓ in Eq. (2) using $(\mathbf{x}^p, \mathbf{y}^p)$ and obtain its attention matrix $\mathbf{A}$;
7     $\mathcal{L}_{F_1} \leftarrow$ compute ℓ in Eq. (3) using $(n(\mathbf{y}^u), \mathbf{y}^u)$ and obtain $\mathbf{A}'$;
8     $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \leftarrow XEnDec$ over the inputs $(\mathbf{x}^p, \mathbf{y}^p)$ and $(n(\mathbf{y}^u), \mathbf{y}^u)$, with the shuffling ratio p_2, $\mathbf{A}$ and $\mathbf{A}'$;
9     $\mathcal{L}_{F_2} \leftarrow$ compute ℓ in Eq. (8);
10   **end**
11   **return** $\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) + \mathcal{L}_{F_1}(\boldsymbol{\theta}) + \mathcal{L}_{F_2}(\boldsymbol{\theta})$; `// Eq. (9).`

---

we group sentences offline by length in Step 3 (cf. batching data in the supplementary document). For adding noise in Step 4, we can follow (Lample et al., 2017) to locally shuffle words while keeping the distance between the original and new position not larger than 3 or set it as a null operation. There are two techniques to boost the final performance.

Computing A: The alignment matrix $\mathbf{A}$ is obtained by averaging the cross-attention weights across all decoder layers and heads. We also add a temperature to control the sharpness of the attention distribution, the reciprocal of which was linearly increased from 0 to 2 during the first $20K$ steps. To avoid overfitting when computing $e(\tilde{\mathbf{z}})$ and $h(\tilde{\mathbf{y}})$, we apply dropout to $\mathbf{A}$ and stop back-propagating gradients through $\mathbf{A}$ when calculating the loss $\mathcal{L}_{F_2}(\boldsymbol{\theta})$.

Computing $h(\tilde{\mathbf{y}})$: Instead of interpolating one-hot labels in Eq. (6), we use the prediction vector $f(\mathbf{x}, \mathbf{y}; \hat{\boldsymbol{\theta}})$ on the sentence pair $(\mathbf{x}, \mathbf{y})$ estimated by the model where $\hat{\boldsymbol{\theta}}$ indicates no gradients are back-propagated through it. However, the predictions made at early stages are usually unreliable. We propose to linearly combine the ground-truth one-hot label with the model prediction using a parameter v, which is computed as $v f_j(\mathbf{x}, \mathbf{y}; \hat{\boldsymbol{\theta}}) + (1 - v) h(y_j)$ where v is gradually annealed from 0 to 1 during the first $20K$ steps [1]. Notice that the prediction vectors are not used in computing the decoder input $e(\tilde{\mathbf{z}})$ which can be clearly distinguished from schedule sampling (Bengio et al., 2015).

3.3. Relation to Other Works

This subsection shows that *XEnDec*, when fed with appropriate inputs, yields learning objectives identical to

---

[1] These two annealing hyperparameters in computing both $\mathbf{A}$ and $h(\tilde{\mathbf{y}})$ are the same for all the models and not elaborately tuned.

Table 1. Comparison with different objectives produced by *XEn-Dec*. Each row shows a set of inputs to *XEnDec* and the corresponding objectives in existing work (the last column). $\mathbf{y}^u_{mask}$ is a sentence of length $|\mathbf{y}^u|$ containing only "$\langle mask \rangle$" tokens. $\mathbf{y}^u_{noise}$ is a sentence obtained by corrupting all the words in $\mathbf{y}^u$ with non-masking noises. $\mathbf{x}^p_{adv}$ and $\mathbf{y}^p_{adv}$ are adversarial sentences in which all the words are substituted with adversarial words.

$(\mathbf{x}$	$\mathbf{y})$	$(\mathbf{x}'$	$\mathbf{y}')$	Objectives
$\mathbf{y}^u$	$\mathbf{y}^u$	$\mathbf{y}^u_{mask}$	$\mathbf{y}^u$	MASS (Song et al., 2019)
$\mathbf{y}^u_{noise}$	$\mathbf{y}^u$	$\mathbf{y}^u_{mask}$	$\mathbf{y}^u$	BART (Lewis et al., 2019)
$\mathbf{x}^p$	$\mathbf{y}^p$	$\mathbf{x}^p_{adv}$	$\mathbf{y}^p_{adv}$	Adv. (Cheng et al., 2019)

two recently proposed self-supervised learning approaches: MASS (Song et al., 2019) and BART (Lewis et al., 2019), as well as a supervised learning approach called *Doubly Adversarial* (Cheng et al., 2019). Table 1 summarizes the inputs of *XEnDec* to recover these approaches.

XEnDec can be used for self-supervised learning. As shown in Table 1, the inputs to *XEnDec* are two pairs of sentences $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$. Given arbitrary alignment matrices, if we set $\mathbf{x}' = \mathbf{y}^u$, $\mathbf{y}' = \mathbf{y}^u$, and $\mathbf{x}$ to be a corrupted copy of $\mathbf{y}^u$, then *XEnDec* is equivalent to the denoising autoencoder which is commonly used to pre-train sequence-to-sequence models such as in MASS (Song et al., 2019) and BART (Lewis et al., 2019). In particular, if we allow $\mathbf{x}'$ to be a dummy sentence of length $|\mathbf{y}^u|$ containing only "$\langle mask \rangle$" tokens ($\mathbf{y}^u_{mask}$ in the table), Eq. (7) yields the learning objective defined in the MASS model (Song et al., 2019) except that losses over unmasked words are not counted in the training loss. Likewise, as shown in Table 1, we can recover BART's objective by setting $\mathbf{x} = \mathbf{y}^u_{noise}$ where $\mathbf{y}^u_{noise}$ is obtained by shuffling tokens or dropping them in $\mathbf{y}^u$. In both cases, *XEnDec* is trained with a self-supervised objective to reconstruct the original sentence from one of its corrupted sentences. Conceptually, denoising autoencoder can be regarded as a degenerated *XEnDec* in which the inputs are two views of its source correspondence for a monolingual sentence, *e.g.*, $n(\mathbf{y})$ and $\mathbf{y}_{mask}$ for $\mathbf{y}$.

XEnDec can also be used in supervised learning. The translation loss proposed in (Cheng et al., 2019) is achieved by letting $\mathbf{x}'$ and $\mathbf{y}'$ be two "adversarial inputs", $\mathbf{x}^p_{adv}$ and $\mathbf{y}^p_{adv}$, both of which consist of adversarial words at each position. For the construction of $\mathbf{x}^p_{adv}$, we refer to Algorithm 1 in (Cheng et al., 2019). In this case, the crossover encoder-decoder is trained with a supervised objective over parallel sentences.

The above connections to existing works illustrate the power of *XEnDec* when it is fed with different kinds of inputs. The results in Section 4.4 show that *XEnDec* is still able to improve the baseline with alternative inputs. However, our experiments show the best configuration found so far

is to use the F_2-*XEnDec* in Algorithm 1 to deeply fuse the monolingual and parallel sentences.

4. Experiments

4.1. Settings

Datasets. We evaluate our approach on two representative, resource-rich translation datasets, WMT'14 English-German and WMT'14 English-French across four translation directions, English→German (En→De), German→English (De→En), English→French (En→Fr), and French→English (Fr→En). To fairly compare with previous state-of-the-art results on these two tasks, we report case-sensitive tokenized BLEU scores calculated by the *multi-bleu.perl* script. The English-German and English-French datasets consist of 4.5M and 36M sentence pairs, respectively. The English, German and French monolingual corpora in our experiments come from the WMT'14 translation tasks. We concatenate all the newscrawl07-13 data for English and German, and newscrawl07-14 for French which results in 90M English sentences, 89M German sentences, and 42M French sentences. We use a word piece model (Schuster & Nakajima, 2012) to split tokenized words into sub-word units. For English-German, we build a shared vocabulary of 32K sub-words units. The validation set is newstest2013 and the test set is newstest2014. The vocabulary for the English-French dataset is also jointly split into 44K sub-word units. The concatenation of newstest2012 and newstest2013 is used as the validation set while newstest2014 is the test set. Refer to the supplementary document for more detailed data pre-processing.

Model and Hyperparameters. We implement our approach on top of the Transformer model (Vaswani et al., 2017) using the *Lingvo* toolkit (Shen et al., 2019). The Transformer models follow the original network settings (Vaswani et al., 2017). In particular, the layer normalization is applied after each residual connection rather than before each sub-layer. The dropout ratios are set to 0.1 for all Transformer models except for the Transformer-big model on English-German where 0.3 is used. We search the hyperparameters using the Transformer-base model on English-German. In our method, the shuffling ratio p_1 is set to 0.50 while 0.25 is used for English-French in Table 6. p_2 is sampled from a Beta distribution $Beta(2,6)$. The dropout ratio of $\mathbf{A}$ is 0.2 for all the models. For decoding, we use a beam size of 4 and a length penalty of 0.6 for English-German, and a beam size of 5 and a length penalty of 1.0 for English-French. We carry out our experiments on a cluster of 128 P100 GPUs and update gradients synchronously. The model is optimized with Adam (Kingma & Ba, 2014) following the same learning rate schedule used in (Vaswani et al., 2017) except for *warmup_steps* which is set to 4000 for both Transform-base and Transformer-big

Table 2. Experiments on WMT'14 English-German and WMT'14 English-French translation.

Models	Methods	En→De	De→En	En→Fr	Fr→En
Base	Reproduced Transformer	28.70	32.23	-	-
	F_2-*XEnDec*	**30.46**	**34.06**	-	-
Big	Reproduced Transformer	29.47	33.12	43.37	39.82
	(Ott et al., 2018)	29.30	-	43.20	-
	(Cheng et al., 2019)	30.01	-	-	-
	(Yang et al., 2019)	30.10	-	42.30	
	(Nguyen et al., 2019)	30.70	-	43.70	-
	(Zhu et al., 2020)	30.75	-	43.78	-
	Joint Training with MASS	30.63	-	43.00	-
	Joint Training with BART	30.88	-	44.18	-
	F_2-*XEnDec*	**31.60**	**34.94**	**45.15**	**41.60**

Table 3. Comparison with the best baseline method in Table 2 in terms of BLEU, BLEURT and YiSi.

Methods	En→De			En→Fr		
	BLEU	BLEURT	YiSi	BLEU	BLEURT	YiSi
Joint Training with BART	30.88	0.225	0.837	44.18	0.488	0.864
F_2-XEnDec	**31.60**	**0.261**	**0.842**	**45.15**	**0.513**	**0.869**

models.

Training Efficiency. When training the vanilla Transformer model, each batch contains 4096×128 tokens of parallel sentences on a 128 P100 GPUs cluster. As there are three losses included in our training objective (Eq. (9)) and the inputs for each of them are different, we evenly spread the GPU memory budget into these three types of data by letting each batch include 2048×128 tokens. Thus the total batch size is $2048 \times 128 \times 3$. The training speed is on average about 60% of the standard training speed. The additional computation cost is partially due to the implementation of the noise function to corrupt the monolingual sentence $\mathbf{y}^u$ and can be reduced by caching noisy data in the data input pipeline. Then the training speed can accelerate to about 80% of the standard training speed.

4.2. Main Results

Table 2 shows the main results on the English-German and English-French datasets. Our method is compared with the following strong baseline methods. (Ott et al., 2018) is the scalable Transformer model. Our reproduced Transformer model performs comparably with their reported results. (Cheng et al., 2019) is a NMT model with adversarial augmentation mechanisms in supervised learning. (Nguyen et al., 2019) boosts NMT performance by adopting multiple rounds of back-translated sentences. Both (Zhu et al., 2020) and (Yang et al., 2019) incorporate the knowledge of pre-trained models into NMT models by treating them as frozen input representations for NMT. We also compare our

approach with MASS (Song et al., 2019) and BART (Lewis et al., 2019). As their goals are to learn generic pre-trained representations from massive monolingual corpora, for fair comparisons, we re-implement their methods using the same backbone model as ours, and jointly optimize their self-supervised objectives together with the supervised objective on the same corpora.

For English-German, our approach achieves significant improvements in both translation directions over the standard Transformer model. Even compared with the strongest baseline on English→German, our approach obtains a $+0.72$ BLEU gain. More importantly, when we apply our approach to a significantly larger dataset, English-French with 36M sentence pairs (vs. English-German with 4.5M sentence pairs), it still yields consistent and notable improvements over the standard Transformer model.

The single-stage approaches (Joint Training with MASS & BART) perform slightly better than the two-stage approaches (Zhu et al., 2020; Yang et al., 2019), which substantiates the benefit of jointly training supervised and self-supervised objectives for resource-rich translation tasks. Among them, BART performs better with stable improvements on English-German and English-French and faster convergence. However, they still lag behind our approach. This is mainly because the $\mathcal{L}_{F_2}$ term in our approach can deeply fuse the supervised and self-supervised objectives instead of simply summing up their training losses. See Section 4.4 for more details.

Furthermore, we evaluate our approach and the best baseline

Table 4. Effect of monolingual corpora sizes.

Methods	Mono. Size	En→De
F_2-XEnDec	×0	28.70
	×1	29.84
	×3	30.36
	×5	30.46
	×10	30.22

Table 5. Finetuning vs. Joint Training.

Methods	En→De
Transformer	28.70
+ Pretrain + Finetune	28.77
F_2-XEnDec (Joint Training)	30.46
+ Pretrain + Finetune	29.70

method (Joint Training with BART) in Table 3 in terms of two additional evaluation metric, BLEURT (Sellam et al., 2020) and YiSi (Lo, 2019), which claim better correlation with human judgement. Results in Table 3 corroborate the superior performance of our approach compared to the best baseline method on both English→German and English→French.

4.3. Analyses

Effect of Monolingual Corpora Sizes. Table 4 shows the impact of monolingual corpora sizes on the performance for our approach. We find that our approach already yields improvements over the baselines when using no monolingual corpora (x0) as well as when using a monolingual corpus with size comparable to the bilingual corpus (1x). As we increase the size of the monolingual corpus to 5x, we obtain the best performance with 30.46 BLEU. However, continuing to increase the data size fails to improve the performance any further. A recent study (Liu et al., 2020a) shows that increasing the model capacity has great potential to exploit extremely large training sets for the Transformer model. We leave this line of exploration as future work.

Finetuning vs. Joint Training. To further study the effect of pre-trained models on the Transformer model and our approach, we use Eq. (3) to pre-train an NMT model on the entire English and German monolingual corpora. Then we finetune the pre-trained model on the parallel English-German corpus. Models finetuned on pre-trained models usually perform better than models trained from scratch at

---

²Our results cannot directly be compared to the numbers in (Edunov et al., 2018) because they use WMT'18 as bilingual data (5.18M) and 10x more monolingual data (226M vs. ours 23M).

Table 6. Results on F_2-XEnDec + Back Translation. Experiments on English-German and English-French are based on the Transformer-big model.

Methods	En→De	En→Fr
Transformer	28.70	43.37
Back Translation	32.09	35.90
(Edunov et al., 2018)	**35.00**[2]	45.60
F_2-XEnDec	31.60	45.15
+ Back Translation	33.70	**46.19**

the early stage of training. However, this advantage gradually vanishes as training progresses (cf. Figure 1 in the supplementary document). As shown in Table 5, Transformer with finetuning achieves virtually identical results as a Transformer trained from scratch. Using the pre-trained model over our approach impairs performance. We believe this may be caused by a discrepancy between the pre-trained loss and our joint training loss.

Back Translation as Noise. One widely applicable method to leverage monolingual data in NMT is back translation (Sennrich et al., 2016). A straightforward way to incorporate back translation into our approach is to treat back-translated corpora as parallel corpora. However, back translation can also be regarded as a type of noise used for constructing $\mathbf{y}_{noise}^u$ in F_2-XEnDec (shown in Fig. 1(b) and Step 4 in Algorithm 1), which can increase the noise diversity. As shown in Table 6, for English→German trained on the Transformer-big model, our approach yields an additional +1.9 BLEU gain when using back translation to noise $\mathbf{y}^u$ and also outperforms the back-translation baseline. When applied to the English-French dataset, we achieve a new state-of-the-art result over the best baseline (Edunov et al., 2018). In contrast, the standard back translation for English-French hurts the performance of Transformer, which is consistent with what was found in previous works, e.g. (Caswell et al., 2019). These results show that our approach is complementary to the back-translation method and performs more robustly when back-translated corpora are less informative although our approach is conceptually different from works related to back translation (Sennrich et al., 2016; Cheng et al., 2016; Edunov et al., 2018).

Robustness to Noisy Inputs. Contemporary NMT systems often suffer from dramatic performance drops when they are exposed to input perturbations (Belinkov & Bisk, 2018; Cheng et al., 2019), even though these perturbations may not be strong enough to alter the meaning of the input sentence. In this experiment, we verify the robustness of the NMT models learned by our approach. Following (Cheng et al., 2019), we evaluate the model performance against word perturbations which specifically includes two types of
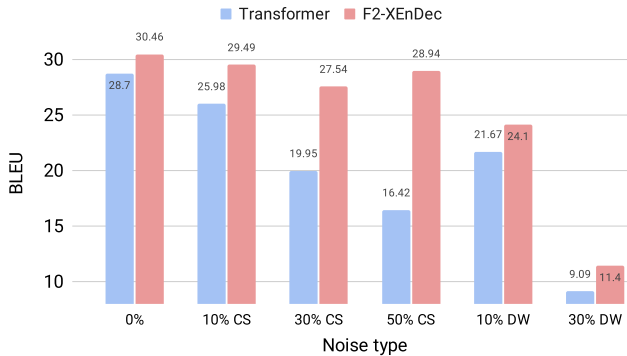
Figure 2. Results on artificial noisy inputs. "CS": code-switching noise. "DW": drop-words noise. We compare our approach to the standard Transformer on different noise types and fractions.

Table 7. Ablation study on English-German.

ID	Different Settings	BLEU
1	Transformer	28.70
2	F_2-*XEnDec*	30.46
3	without $\mathcal{L}_{F_2}$	29.21
4	without $\mathcal{L}_{F_1}$ (a prior alignment is used)	29.55
5	$\mathcal{L}_{\mathcal{S}}$ with *XEnDec* over parallel data	29.23
6	*XEnDec* is replaced by Mixup	29.67
7	without dropout $\mathbf{A}$ and model predictions	29.87
8	without model predictions	30.24

noise to perturb the dataset. The first type of noise is code-switching noise (CS) which randomly replaces words in the source sentences with their corresponding target-language words. Alignment matrices are employed to find the target-language words in the target sentences. The other one is drop-words noise (DW) which randomly discards some words in the source sentences. Figure 2 shows that our approach exhibits higher robustness than the standard Transformer model across all noise types and noise fractions. In particular, our approach performs much more stable for the code-switching noise.

4.4. Ablation Study

Table 7 studies the contributions of the key components and verifies the design choices in our approach.

Contribution of $\mathcal{L}_{F_2}$. We first verify the importance of our core loss term $\mathcal{L}_{F_2}$. When $\mathcal{L}_{F_2}$ is removed in Eq. (9), the training objective is equivalent to summing up supervised ($\mathcal{L}_{\mathcal{S}}$) and self-supervised ($\mathcal{L}_{F_1}$) losses. By comparing Row 2 and 3 in Table 7, we observe a sharp drop (-1.25 BLEU points) caused by the absence of $\mathcal{L}_{F_2}$. This result demonstrates the crucial role of the proposed F_2-*XEnDec* that can

extract complementary signals to facilitate the joint training. We believe this is because of the deep fusion of monolingual and parallel sentences at instance level.

Inputs to *XEnDec*. To validate the proposed task, *XEnDec*, we apply it over different types of inputs. The first one directly combines the parallel and monolingual sentences without using noisy monolingual sentences, which is equivalent to removing $\mathcal{L}_{F_1}$ (Row 4 in Table 7). We achieve this by setting $n(\mathbf{y}^u) = \mathbf{y}^u$ in Algorithm 1. However, we cannot obtain $\mathbf{A}'$ required by the algorithm (Line 7 in Algorithm 1) which leads to the failure of calculating the loss. Thus we design a prior alignment matrix to handle this issue (cf. Section 2 in the supplementary document). The second experiment utilizes *XEnDec* only over parallel sentences (Row 5 in Table 7). We can find these two cases can both achieve better performance compared to the standard Transformer model (Row 1 in the table). These results show the efficacy of the proposed *XEnDec* on different types of inputs. Their gap to our final method shows the rationale of using both parallel and monolingual sentences as inputs. We hypothesize this is because *XEnDec* implicitly regularizes the model by shuffling and reconstructing words in parallel and monolingual sentences.

Comparison to Mixup. We use Mixup (Zhang et al., 2018) to replace our second *XEnDec* to compute $\mathcal{L}_{F_2}$ while keeping the first *XEnDec* untouched. When applying Mixup on a pair of training data $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$, Eq. (4), Eq. (5) and Eq. (6) are replaced by $e(\tilde{x}_i) = \lambda e(x_i) + (1 - \lambda)e(x'_i)$, $e(\tilde{z}_j) = \lambda e(y_{j-1}) + (1 - \lambda)e(y'_{j-1})$ and $h(\tilde{y}_j) = \lambda h(y_j) + (1 - \lambda)h(y'_j)$, respectively, where λ is sampled from a Beta distribution. The comparison between Row 6 and Row 2 in Table 7 shows that Mixup leads to a worse result. Different from Mixup which encourages the model to behave linearly to the linear interpolation of training examples, our task combines training examples in a non-linear way in the source end, and forces the model to decouple the non-linear integration in the target end while predicting the entire target sentence based on its partial source sentence.

Computation of A and $h(\tilde{\mathbf{y}})$. The last two rows in Table 7 verify the impact of two training techniques discussed in Section 3.2. Removing these components would lower the performance.

5. Related Work

The recent past has witnessed an increasing interest in the research community on leveraging pre-training models to boost NMT model performance (Ramachandran et al., 2016; Lample & Conneau, 2019; Song et al., 2019; Lewis et al., 2019; Edunov et al., 2018; Zhu et al., 2020; Yang et al., 2019; Liu et al., 2020b). Most successes come from low-resource and zero-resource translation tasks. (Zhu et al.,

2020) and (Yang et al., 2019) achieve some promising results on resource-rich translations. They propose to combine NMT model representations and frozen pre-trained representations under the common two-stage framework. The bottleneck of these methods is that these two stages are decoupled and separately learned, which exacerbates the difficulty of finetuning self-supervised representations on resource-rich language pairs. Our method, on the other hand, jointly trains self-supervised and supervised NMT models to close the gap between representations learned from either of them with an essential new subtask, *XEnDec*. In addition, our new subtask can be applied to combine different types of inputs. Experimental results show that our method consistently outperforms previous approaches across several translation benchmarks and establishes a new state-of-the-art result on WMT'14 English-French when applying *XEnDec* to back-translated corpora.

Another line of research related to ours originates in computer vision by interpolating images and their labels (Zhang et al., 2018; Yun et al., 2019) which have been shown effective in improving generalization (Arazo et al., 2019; Jiang et al., 2020; Xu et al., 2021; Northcutt et al., 2021) and robustness of convolutional neural network (Hendrycks et al., 2020). Recently, some research efforts have been devoted to introducing this idea to NLP applications (Cheng et al., 2020; Guo et al., 2020; Chen et al., 2020). Our *XEnDec* shares the commonality of combining example pairs. However, *XEnDec*'s focus is on sequence-to-sequence learning for NLP with the aim of using self-supervised learning to complement supervised learning in joint training.

6. Conclusion

This paper has presented a joint training approach, F_2-*XEnDec*, to combine self-supervised and supervised learning in a single stage. The key part is a novel cross encoder-decoder which can be used to "interbreed" monolingual and parallel sentences, which can also be fed with different types of inputs and recover some popular self-supervised and supervised training objectives.

Experiments on two resource-rich translation tasks, WMT'14 English-German and WMT'14 English-French, show that joint training performs favorably against two-stage training approaches when an enormous amount of labeled and unlabeled data is available. When applying *XEnDec* to deeply fuse monolingual and parallel sentences resulting in F_2-*XEnDec*, the joint training paradigm can better exploit the complementary signal from unlabeled data with significantly stronger performance. Finally, F_2-*XEnDec* is capable of improving the NMT robustness against input perturbations such as code-switching noise widely found in social media.

In the future, we plan to further examine the effectiveness of our approach on larger-scale corpora with high-capacity models. We also plan to design more expressive noise functions for our approach.

Acknowledgements

References

Arazo, E., Ortego, D., Albert, P., O'Connor, N., and McGuinness, K. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning (ICML)*, 2019.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, 2015.

Belinkov, Y. and Bisk, Y. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations (ICLR)*, 2018.

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. Scheduled sampling for sequence prediction with recurrent neural networks. *arXiv preprint arXiv:1506.03099*, 2015.

Caswell, I., Chelba, C., and Grangier, D. Tagged back-translation. *arXiv preprint arXiv:1906.06442*, 2019.

Chen, J., Yang, Z., and Yang, D. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Cheng, Y., Xu, W., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. Semi-supervised learning for neural machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

Cheng, Y., Jiang, L., and Macherey, W. Robust neural machine translation with doubly adversarial inputs. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Cheng, Y., Jiang, L., Macherey, W., and Eisenstein, J. Advaug: Robust adversarial augmentation for neural machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of*

the Association for Computational Linguistics (NAACL), 2019.

Edunov, S., Ott, M., Auli, M., and Grangier, D. Understanding back-translation at scale. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Edunov, S., Baevski, A., and Auli, M. Pre-trained language model representations for language generation. *arXiv preprint arXiv:1903.09722*, 2019.

French, R. M. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 1999.

Garg, S., Peitz, S., Nallasamy, U., and Paulik, M. Jointly learning to align and translate with transformer models. *arXiv preprint arXiv:1909.02074*, 2019.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. Convolutional sequence to sequence learning. In *International Conference on Machine Learning (ICML)*, 2017.

Guo, D., Kim, Y., and Rush, A. Sequence-level mixed sample data augmentation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations (ICLR)*, 2020.

Jiang, L., Huang, D., Liu, M., and Yang, W. Beyond synthetic noise: Deep learning on controlled noisy labels. In *International Conference on Machine Learning (ICML)*, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lample, G. and Conneau, A. Cross-lingual language model pretraining. *arXiv*, pp. arXiv–1901, 2019.

Lample, G., Conneau, A., Denoyer, L., and Ranzato, M. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*, 2017.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Liu, X., Duh, K., Liu, L., and Gao, J. Very deep transformers for neural machine translation. *arXiv preprint arXiv:2008.07772*, 2020a.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 2020b.

Lo, C.-k. Yisi-a unified semantic mt quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation*, 2019.

Nguyen, X.-P., Joty, S., Kui, W., and Aw, A. T. Data diversification: An elegant strategy for neural machine translation. *arXiv preprint arXiv:1911.01986*, 2019.

Northcutt, C. G., Jiang, L., and Chuang, I. L. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 2021.

Och, F. J. and Ney, H. The alignment template approach to statistical machine translation. *Computational linguistics*, 2004.

Ott, M., Edunov, S., Grangier, D., and Auli, M. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*, 2018.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training, 2018.

Ramachandran, P., Liu, P. J., and Le, Q. V. Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv:1611.02683*, 2016.

Rieger, R., Michaelis, A., and Green, M. M. *Glossary of genetics and cytogenetics: classical and molecular*. Springer Science & Business Media, 2012.

Schuster, M. and Nakajima, K. Japanese and korean voice search. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

Sellam, T., Das, D., and Parikh, A. Bleurt: Learning robust metrics for text generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Sennrich, R., Haddow, B., and Birch, A. Improving nerual machine translation models with monolingual data. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

Shen, J., Nguyen, P., Wu, Y., Chen, Z., Chen, M. X., Jia, Y., Kannan, A., Sainath, T., Cao, Y., Chiu, C.-C., et al. Lingvo: a modular and scalable framework for sequence-to-sequence modeling. *arXiv preprint arXiv:1902.08295*, 2019.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning (ICML)*, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, 2008.

Xu, Y., Zhu, L., Jiang, L., and Yang, Y. Faster meta update strategy for noise-robust deep learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Yang, J., Wang, M., Zhou, H., Zhao, C., Yu, Y., Zhang, W., and Li, L. Towards making the most of bert in neural machine translation. *arXiv preprint arXiv:1908.05672*, 2019.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018.

Zhu, J., Xia, Y., Wu, L., He, D., Qin, T., Zhou, W., Li, H., and Liu, T.-Y. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*, 2020.