Robust Neural Machine Translation with Doubly Adversarial Inputs

Yong Cheng, Lu Jiang, and Wolfgang Macherey
Google AI
{chengyong, lujiang, wmach}@google.com

Abstract
Neural machine translation (NMT) often suffers from the vulnerability to noisy perturbations in the input. We propose an approach to improving the robustness of NMT models, which consists of two parts: (1) attack the translation model with adversarial source examples; (2) defend the translation model with adversarial target inputs to improve its robustness against the adversarial source inputs. For the generation of adversarial inputs, we propose a gradient-based method to craft adversarial examples informed by the translation loss over the clean inputs. Experimental results on Chinese-English and English-German translation tasks demonstrate that our approach achieves significant improvements (2.8 and 1.6 BLEU points) over Transformer on standard clean benchmarks as well as exhibiting higher robustness on noisy data.

1 Introduction
In recent years, neural machine translation (NMT) has achieved tremendous success in advancing the quality of machine translation (Wu et al., 2016; Hieber et al., 2017). As an end-to-end sequence learning framework, NMT consists of two important components, the encoder and decoder, which are usually built on similar neural networks of different types, such as recurrent neural networks (Sutskever et al., 2014; Bahdanau et al., 2015; Chen et al., 2018), convolutional neural networks (Gehring et al., 2017), and more recently on transformer networks (Vaswani et al., 2017). To overcome the bottleneck of encoding the entire input sentence into a single vector, an attention mechanism was introduced, which further enhanced translation performance (Bahdanau et al., 2015). Deeper neural networks with increased model capacities in NMT have also been explored and shown promising results (Bapna et al., 2018).

Despite these successes, NMT models are still vulnerable to perturbations in the input sentences. For example, Belinkov and Bisk (2018) found that NMT models can be immensely brittle to small perturbations applied to the inputs. Even if these perturbations are not strong enough to alter the meaning of an input sentence, they can nevertheless result in different and often incorrect translations. Consider the example in Table 1, the Transformer model will generate a worse translation (revealing gender bias) for a minor change in the input from “he” to “she”.

<table>
<thead>
<tr>
<th>Input</th>
<th>he is a handicapped person, my daughter is in good health. ✓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>one of her handicapped people, my daughter is in good health. ×</td>
</tr>
</tbody>
</table>

Table 1: An example of Transformer NMT translation result for an input and its perturbed input by replacing “他(he)” to “她(she)”.

A few studies have been proposed in other natural language processing (NLP) tasks aiming to tackle this issue in classification tasks, e.g. in (Miyato et al., 2017; Alzantot et al., 2018; Ebrahimi et al., 2018b; Zhao et al., 2018). As for NMT, previous approaches relied on prior knowledge to generate adversarial examples to improve the robustness, neglecting specific downstream NMT models. For example, Belinkov and Bisk (2018) and Karpukhin et al. (2019) studied how...

This paper studies learning a robust NMT model that is able to overcome small perturbations in the input sentences. Different from prior work, our work deals with the perturbed examples jointly generated by a white-box NMT model, which means that we have access to the parameters of the attacked model. To the best of our knowledge, the only previous work on this topic is from (Ebrahimi et al., 2018a) on character-level NMT. Overcoming adversarial examples in NMT is a challenging problem as the words in the input are represented as discrete variables, making them difficult to be switched by imperceptible perturbations. Moreover, the characteristics of sequence generation in NMT further intensify this difficulty. To tackle this problem, we propose a gradient-based method, AdvGen, to construct adversarial examples guided by the final translation loss from the clean inputs of a NMT model. AdvGen is applied to both encoding and decoding stages: (1) we attack a NMT model by generating adversarial source inputs that are sensitive to the training loss; (2) we then defend the NMT model with the adversarial target inputs, aiming at reducing the prediction errors for the corresponding adversarial source inputs.

Our contribution is threefold:

1. A white-box method to generate adversarial examples is explored for NMT. Our method is a gradient-based approach guided by the translation loss.

2. We propose a new approach to improving the robustness of NMT with doubly adversarial inputs. The adversarial inputs in the encoder aim at attacking the NMT models, while those in the decoder are capable of defending the errors in predictions.

3. Our approach achieves significant improvements over the previous state-of-the-art Transformer model on two common translation benchmarks.

Experimental results on the standard Chinese-English and English-German translation benchmarks show that our approach yields an improvement of 2.8 and 1.6 BLEU points over the state-of-the-art models including Transformer (Vaswani et al., 2017). This result substantiates that our model improves the generalization performance over the clean benchmark datasets. Further experiments on noisy text verify the ability of our approach to improving robustness. We also conduct ablation studies to gain further insight into which parts of our approach matter the most.

2 Background

Neural Machine Translation NMT is typically a neural network with an encoder-decoder architecture. It aims to maximize the likelihood of a parallel corpus \( S = \{ (x^{(s)}, y^{(s)}) \}_{s=1}^{|S|} \). Different variants derived from this architecture have been proposed recently (Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017). This paper focuses on the recent Transformer model (Vaswani et al., 2017) due to its superior performance, although our approach seems applicable to other models, too.

The encoder in NMT maps a source sentence \( x = x_1, \ldots, x_I \) to a sequence of \( I \) word embeddings \( e(x) = e(x_1), \ldots, e(x_I) \). Then the word embeddings are encoded to their corresponding continuous hidden representations \( h \) by the transformation layer. Similarly, the decoder maps its target input sentence \( z = z_1, \ldots, z_J \) to a sequence of \( J \) word embeddings. For clarity, we denote the input and output in the decoder as \( z \) and \( y \). \( z \) is a shifted copy of \( y \) in the standard NMT model, \( i.e. \ z = (sos), y_1, \ldots, y_{J-1}, \) where \( \langle sos \rangle \) is a start symbol. Conditioned on the hidden representations \( h \) and the target input \( z \), the decoder generates \( y \) as:

\[
P(y|x; \theta_{mt}) = \prod_{j=1}^{J} P(y_j|z_{\leq j}, h; \theta_{mt})
\]

(1)

where \( \theta_{mt} \) is a set of model parameters and \( z_{\leq j} \) is a partial target input. The training loss on \( S \) is defined as:

\[
L_{clean}(\theta_{mt}) = \frac{1}{|S|} \sum_{(x,y) \in S} - \log P(y|x; \theta_{mt})
\]

(2)

Adversarial Examples Generation An adversarial example is usually constructed by corrupting the original input with a small perturbation such that the difference to the original input remains
less perceptible but dramatically distorts the model output. The adversarial examples can be generated by a white-box or black-box model, where the latter does not have access to the attacked models and often relies on prior knowledge. The former white-box examples are generated using the information of the attacked models. Formally, a set of adversarial examples $\mathcal{Z}(x, y)$ is generated with respect to a training sample $(x, y)$ by solving an optimization problem:

$$\left\{ x' \mid \mathcal{R}(x', x) \leq \epsilon, \arg\max_{x'} J(x', y; \theta) \right\}$$

(3)

where $J(\cdot)$ measures the possibility of a sample being adversarial, and $\mathcal{R}(x', x)$ captures the degree of imperceptibility for a perturbation. For example, in the classification task, $J(\cdot)$ is a function outputting the most possible target class $y'$ ($y' \neq y$) when fed with the adversarial example $x'$. Although it is difficult to give a precise definition of the degree of imperceptibility $\mathcal{R}(x', x)$, $l_\infty$ norm is usually used to bound the perturbations in image classification (Goodfellow et al., 2015).

3 Approach

Our goal is to learn robust NMT models that can overcome small perturbations in the input sentences. As opposed to images, where small perturbations to pixels are imperceptible, even a single word change in natural languages can be perceived. NMT is a sequence generation model wherein each output word is conditioned on all previous predictions. Thus, one question is how to design meaningful perturbation operations for NMT.

We propose a gradient-based approach, called AdvGen, to construct adversarial examples and use these examples to both attack as well as defend the NMT model. Our intuition is that an ideal model would generate similar translation results for similar input sentences despite any small difference caused by perturbations.

The attack and defense are carried out in the end-to-end training of the NMT model. We first use AdvGen to construct an adversarial example $x'$ from the original input $x$ to attack the NMT model. We then use AdvGen to find an adversarial target input $z'$ from the decoder input $x$ to improve the NMT model robustness to adversarial perturbations in the source input $x'$. Thereby we hope the NMT model will be robust against both the source adversarial input $x'$ and adversarial perturbations in target predictions $z'$. The rest of this section will discuss the attack and defense procedures in detail.

3.1 Attack with Adversarial Source Inputs

Following (Goodfellow et al., 2015; Miyato et al., 2017; Ebrahimi et al., 2018b), we study the white-box method to generate adversarial examples tightly guided by the training loss. Given a parallel sentence pair $(x, y)$, according to Eq. (3), we generate a set of adversarial examples $\mathcal{A}(x, y)$ specific to the NMT model by:

$$\left\{ x' \mid \mathcal{R}(x', x) \leq \epsilon, \arg\max_{x'} -\log P(y|x'; \theta_{mt}) \right\}$$

(4)

where we use the negative log translation probability $-\log P(y|x'; \theta_{mt})$ to estimate $J(\cdot)$ in Eq. (3). The formula constructs adversarial examples that are expected to distort the current prediction and retain semantic similarity bounded by $\mathcal{R}$.

It is intractable to obtain an exact solution for Eq. (4). We therefore resort to a greedy approach based on the gradient to circumvent it. For the original input $x$, we induce a possible adversarial word $x'_i$ for the word $x_i$ in $x$:

$$x'_i = \arg\max_{x \in \mathcal{V}_x} (e(x) - e(x_i), g_{x_i})$$

(5)

$$g_{x_i} = \nabla_{e(x_i)} - \log P(y|x; \theta)$$

(6)

where $g_{e_{x_i}}$ is a gradient vector wrt. $e(x_i)$, $\mathcal{V}_x$ is the vocabulary for the source language, and $\sim e(\cdot, \cdot)$ denotes the similarity function by calculating the cosine distance between two vectors.

Eq. (5) enumerates all words in $\mathcal{V}_x$ incurring formidable computational cost. We hence substitute it with a dynamic set $\mathcal{V}_{x_i}$ that is specific for each word $x_i$. Let $Q(x_i, x) \in \mathbb{R}^{\mathcal{V}}$ denote the likelihood of the $i$-th word in the sentence $x$. Define $\mathcal{V}_{x_i} = \text{top}_n(Q(x_i, x))$ as the set of the $n$ most probable words among the top $n$ scores in terms of $Q(x_i, x)$, where $n$ is a small constant integer and $|\mathcal{V}_{x_i}| \ll |\mathcal{V}_x|$. For the source, we estimate it from:

$$Q_{\text{src}}(x_i, x) = P_{lm}(x|x_{<i}, x_{>i}; \theta_{lm}^x)$$

(7)

Here, $P_{lm}$ is a bidirectional language model for the source language.

The introduction of language model has three benefits. First, it enables a computationally feasible way to approximate Eq. (5). Second, the
After generating an adversarial example $x'$, we treat $(x', y)$ as a new training data point to improve the model’s robustness. These adversarial examples in the source tend to introduce errors which may accumulate and cause drastic changes to the decoder prediction. To defend the model from errors in the decoder predictions, we generate an adversarial target input by $AdvGen$, similar to what we discussed in Section 3.1. The decoder trained with the adversarial target input is expected to be more robust to the small perturbations introduced in the source input. The ablation study results in Table 8 substantiate the benefit of this defense mechanism.

Formally, let $z$ be the decoder input for the sentence pair $(x, y)$. We use the same $AdvGen$ function to generate an adversarial target input $z'$ from $z$ by:

$$z' = AdvGen(z, Q_{trg}, D_{trg}, -\log P(y|x')) \quad (8)$$

Note that for the target, the translation loss in Eq. (6) is replaced by $-\log P(y|x')$. $Q_{trg}$ is the likelihood for selecting the target word candidate set $\mathcal{V}_z$. To compute it, we combine the NMT model prediction with a language model $P_{lm}(y; \theta_{lm})$ as follow:

$$Q_{trg}(z_i, z) = \lambda P(z_i|z_{<i}, z_{>i}; \theta_{lm}) + (1 - \lambda) P(z_i|x_i, x'; \theta_{mt}) \quad (9)$$

where $\lambda$ balances the importance between two models.

$D_{trg}$ is a distribution for sampling positions for the target input. Different from the uniform distribution used in the source, in the target sentence we want to change those relevant words influenced by the perturbed words in the source input. To do so, we use the attention matrix $\mathcal{M}$ learned in the NMT model, obtained at the current mini-batch, to compute the distribution over $(x, y, x')$ by:

$$P(j) = \frac{\sum_j M_{ij} \delta(x_i, x'_j)}{\sum_k \sum_{i'} M_{ik} \delta(x_{i'}, x'_j)}, j \in \{1, ..., |y|\} \quad (10)$$

where $M_{ij}$ is the attention score between $x_i$ and $y_j$ and $\delta(x, x')$ is an indicator function that yields 1 if $x_i \neq x'_j$ and 0 otherwise.

### 3.3 Training

Algorithm 2 details the entire procedure to calculate the robustness loss for a parallel sentence pair $(x, y)$. We run $AdvGen$ twice to obtain $x'$ and $z'$. We do not backpropagate gradients over $AdvGen$ when updating parameters, which just plays a role of data generator. In our implementation, this function incurs at most a 20% time overhead compared to the standard Transformer model. Accordingly, we compute the robustness loss on $\mathcal{S}$ as:

$$\mathcal{L}_{robust}(\theta_{mt}) = \frac{1}{|\mathcal{S}|} \sum_{(x, y) \in \mathcal{S}} -\log P(y|x', z'; \theta_{mt}) \quad (11)$$
The final training objective $L$ is a combination of four loss functions:

$$L(\theta_{mt}, \theta_{lm}^{x}, \theta_{lm}^{y}) = L_{\text{clean}}(\theta_{mt}) + L_{lm}(\theta_{lm}^{x}) + L_{\text{robust}}(\theta_{mt}) + L_{lm}(\theta_{lm}^{y})$$

where $\theta_{lm}^{x}$ and $\theta_{lm}^{y}$ are two sets of model parameters for source and target bidirectional language models, respectively. The word embeddings are shared between $\theta_{mt}$ and $\theta_{lm}^{x}$ and likewise between $\theta_{mt}$ and $\theta_{lm}^{y}$.

## 4 Experiments

### 4.1 Setup

We conducted experiments on Chinese-English and English-German translation tasks. The Chinese-English training set is from the LDC corpus that comprises 1.2M sentence pairs. We used the NIST 2006 dataset as the validation set for model selection and hyper-parameters tuning, and NIST 2002, 2003, 2004, 2005, 2008 as test sets. For the English-German translation task, we used the WMT’14 corpus consisting of 4.5M sentence pairs. The validation set is newstest2013, and the test set is newstest2014.

In both translation tasks, we merged the source and target training sets and used byte pair encoding (BPE) (Sennrich et al., 2016c) to encode words through sub-word units. We built a shared vocabulary of 32K sub-words for English-German and created shared BPE codes with 60K operations for Chinese-English that induce two vocabularies with 46K Chinese sub-words and 30K English sub-words. We report case-sensitive tokenized BLEU scores for English-German and case-insensitive tokenized BLEU scores for Chinese-English (Papineni et al., 2002). For a fair comparison, we did not average multiple checkpoints (Vaswani et al., 2017), and only report results on a single converged model.

We implemented our approach based on the Transformer model (Vaswani et al., 2017). In $\text{AdvGen}$, We modified multiple positions in the source and target input sentences in parallel. The bidirectional language model used in $\text{AdvGen}$ consists of left-to-right and right-to-left Transformer networks, a linear layer to combine final representations from these two networks, and a softmax layer to make predictions. The Transformer network was built using six transformation layers which keeps consistent with the encoder in the Transformer model. The hyperparameters in the Transformer model were set according to the default values described in (Vaswani et al., 2017). We denote the Transformer model with 512 hidden units as Trans.-Base and 1024 hidden units as Trans.-Big.

We tuned the hyperparameters in our approach on the validation set via a grid search. Specifically, $\lambda$ was set to 0.5. The $n$ in $\text{top}_n$ to select word candidates was set to 10. The ratio pair $(\gamma_{src}, \gamma_{trg})$ was set to $(0.25, 0.50)$ with the exception of Trans.-Base on English-German where it was set to $(0.15, 0.15)$. We treated the single part of parallel corpus as monolingual data to train bidirectional language models without introducing additional data. The model parameters in our approach were trained from scratch except for the parameters in language models initialized by the models pre-trained on the single part of parallel corpus. The parameters of language models were still updated during robustness training.

### 4.2 Main Results

Table 3 shows the BLEU scores on the NIST Chinese-English translation task. We first compare our approach with the Transformer model (Vaswani et al., 2017) on which our model is built. As we see, the introduction of our method to the standard backbone model (Trans.-Base) leads to substantial improvements across the validation and test sets. Specifically, our approach achieves an average gain of 2.25 BLEU points and up to 2.8 BLEU points on NIST03.

Table 4 shows the results on WMT’14 English-
<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>MT06</th>
<th>MT02</th>
<th>MT03</th>
<th>MT04</th>
<th>MT05</th>
<th>MT08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vaswani et al. (2017)</td>
<td>Trans.-Base</td>
<td>44.59</td>
<td>44.82</td>
<td>43.68</td>
<td>45.60</td>
<td>44.57</td>
<td>35.07</td>
</tr>
<tr>
<td>Miyato et al. (2017)</td>
<td>Trans.-Base</td>
<td>45.11</td>
<td>45.95</td>
<td>44.68</td>
<td>45.99</td>
<td>45.32</td>
<td>35.84</td>
</tr>
<tr>
<td>Sennrich et al. (2016a)</td>
<td>Trans.-Base</td>
<td>44.96</td>
<td>46.03</td>
<td>44.81</td>
<td>46.01</td>
<td>45.69</td>
<td>35.32</td>
</tr>
<tr>
<td>Wang et al. (2018)</td>
<td>Trans.-Base</td>
<td>45.47</td>
<td>46.31</td>
<td>45.30</td>
<td>46.45</td>
<td>45.62</td>
<td>35.66</td>
</tr>
<tr>
<td>Cheng et al. (2018)</td>
<td>RNMT&lt;sub&gt;lex.&lt;/sub&gt;</td>
<td>43.57</td>
<td>44.82</td>
<td>42.95</td>
<td>45.05</td>
<td>43.45</td>
<td>34.85</td>
</tr>
<tr>
<td></td>
<td>RNMT&lt;sub&gt;feat.&lt;/sub&gt;</td>
<td>44.44</td>
<td>46.10</td>
<td>44.07</td>
<td>45.61</td>
<td>44.06</td>
<td>34.94</td>
</tr>
<tr>
<td>Cheng et al. (2018)</td>
<td>Trans.-Base&lt;sub&gt;feat.&lt;/sub&gt;</td>
<td>45.37</td>
<td>46.16</td>
<td>44.41</td>
<td>46.32</td>
<td>45.30</td>
<td>35.85</td>
</tr>
<tr>
<td></td>
<td>Trans.-Base&lt;sub&gt;lex.&lt;/sub&gt;</td>
<td>45.78</td>
<td>45.96</td>
<td>45.51</td>
<td>46.49</td>
<td>45.73</td>
<td>36.08</td>
</tr>
<tr>
<td>Sennrich et al. (2016b)*</td>
<td>Trans.-Base</td>
<td>46.39</td>
<td>47.31</td>
<td>47.10</td>
<td>47.81</td>
<td>45.69</td>
<td>36.43</td>
</tr>
<tr>
<td>Ours</td>
<td>Trans.-Base</td>
<td>46.95</td>
<td>47.06</td>
<td>46.48</td>
<td>47.39</td>
<td>46.58</td>
<td>37.38</td>
</tr>
<tr>
<td>Ours + BackTranslation*</td>
<td>Trans.-Base</td>
<td>47.74</td>
<td>48.13</td>
<td>47.83</td>
<td>49.13</td>
<td>49.04</td>
<td>38.61</td>
</tr>
</tbody>
</table>

Table 2: Comparison with baseline methods trained on different backbone models (second column). * indicates the method trained using an extra corpus.

<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>MT06</th>
<th>MT02</th>
<th>MT03</th>
<th>MT04</th>
<th>MT05</th>
<th>MT08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vaswani et al. (2017)</td>
<td>Trans.-Base</td>
<td>44.59</td>
<td>44.82</td>
<td>43.68</td>
<td>45.60</td>
<td>44.57</td>
<td>35.07</td>
</tr>
<tr>
<td>Ours</td>
<td>Trans.-Base</td>
<td><strong>46.95</strong></td>
<td><strong>47.06</strong></td>
<td><strong>46.48</strong></td>
<td><strong>47.39</strong></td>
<td><strong>46.58</strong></td>
<td><strong>37.38</strong></td>
</tr>
</tbody>
</table>

Table 3: Results on NIST Chinese-English translation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vaswani et al.</td>
<td>Trans.-Base</td>
<td>27.30</td>
</tr>
<tr>
<td></td>
<td>Trans.-Big</td>
<td>28.40</td>
</tr>
<tr>
<td>Chen et al.</td>
<td>RNMT+</td>
<td>28.49</td>
</tr>
<tr>
<td>Ours</td>
<td>Trans.-Base</td>
<td>28.34</td>
</tr>
<tr>
<td></td>
<td>Trans.-Big</td>
<td><strong>30.01</strong></td>
</tr>
</tbody>
</table>

Table 4: Results on WMT’14 English-German translation.

German translation. We compare our approach with Transformer for different numbers of hidden units (i.e. 1024 and 512) and a related RNN-based NMT model RNMT+ (Chen et al., 2018). As is shown in Table 4, our approach achieves improvements over the Transformer for the same number of hidden units, i.e. 1.04 BLEU points over Trans.-Base, 1.61 BLEU points over Trans.-Big, and 1.52 BLEU points over RNMT+ model. Recall that our approach is built on top of the Transformer model. The notable gain in terms of BLEU verifies our English-German translation model.

4.3 Comparison to Baseline Methods

To further verify our method, we compare to recent related techniques for robust NMT learning methods. For a fair comparison, we implemented all methods on the same Transformer backbone.

Miyato et al. (2017) applied perturbations to word embeddings using adversarial learning in text classification tasks. We apply this method to the NMT model.

Sennrich et al. (2016a) augmented the training data with word dropout. We follow their method to randomly set source word embeddings to zero with the probability of 0.1. This simple technique performs reasonably well on the Chinese-English translation.

Wang et al. (2018) introduced a data-augmentation method for NMT called SwitchOut to randomly replace words in both source and target sentences with other words.

Cheng et al. (2018) employed adversarial stability training to improve the robustness of NMT. We cite their numbers reported in the paper for the RNN-based NMT backbone and implemented their method on the Transformer backbone. We consider two types of noisy perturbations in their method and use subscripts *lex.* and *fea.* to denote them.

Sennrich et al. (2016b) is a common data-augmentation method for NMT. The method back-translates monolingual data by an inverse translation model. We sampled 1.2M English sentences from the Xinhua portion of the GIGAWORD corpus as monolingual data. We then back-translated them with an English-Chinese NMT model and
Table 5: Comparison of translation results of Transformer and our model for an input and its perturbed input.

<table>
<thead>
<tr>
<th>Method</th>
<th>0.00</th>
<th>0.05</th>
<th>0.10</th>
<th>0.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vaswani et al.</td>
<td>44.59</td>
<td>41.54</td>
<td>38.84</td>
<td>35.71</td>
</tr>
<tr>
<td>Miyato et al.</td>
<td>45.11</td>
<td>42.11</td>
<td>39.39</td>
<td>36.44</td>
</tr>
<tr>
<td>Cheng et al.</td>
<td>45.78</td>
<td>42.90</td>
<td>40.58</td>
<td>38.46</td>
</tr>
<tr>
<td>Ours</td>
<td>46.95</td>
<td>44.20</td>
<td>41.71</td>
<td>39.89</td>
</tr>
</tbody>
</table>

Table 6: Results on artificial noisy inputs. The column lists results for different noise fractions.

<table>
<thead>
<tr>
<th>Method</th>
<th>0.00</th>
<th>0.05</th>
<th>0.10</th>
<th>0.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vaswani et al.</td>
<td>100</td>
<td>77.08</td>
<td>62.00</td>
<td>52.50</td>
</tr>
<tr>
<td>Miyato et al.</td>
<td>100</td>
<td>79.19</td>
<td>63.12</td>
<td>53.51</td>
</tr>
<tr>
<td>Cheng et al.</td>
<td>100</td>
<td>79.66</td>
<td>65.16</td>
<td>56.11</td>
</tr>
<tr>
<td>Ours</td>
<td>100</td>
<td>82.76</td>
<td>69.23</td>
<td>60.70</td>
</tr>
</tbody>
</table>

Table 7: BLEU scores computed using the zero noise fraction output as a reference.

4.4 Results on Noisy Data

We have shown improvements on the standard clean benchmarks. This subsection validates the robustness of the NMT models over artificial noise. To this end, we added synthetic noise to the clean validation set by randomly replacing a word with a relevant word according to the similarity of their word embeddings. We repeated the process in a sentence according to a pre-defined noise fraction where a noise level of 0.0 yields the original clean dataset while 1.0 provides an entirely altered set. For each sentence, we generated 100 noisy sentences. We then re-scored those sentences using a pre-trained bidirectional language model, and picked the best one as the noisy input.

Table 6 shows results on artificial noisy inputs. BLEU scores were computed against the ground-truth translation result. As we see, our approach outperforms all baseline methods across all noise levels. The improvement is generally more evident when the noise fraction becomes larger.

To further analyze the prediction stability, we compared the model outputs for clean and noisy inputs. To do so, we selected the output of a model on clean input (noise fraction equals 0.0) as a reference and computed the BLEU score against this reference. Table 7 presents the results where the second column 100 means that the output is exactly the same as the reference. The relative drop of our model, as the noise level grows, is smaller compared to other baseline methods. The results in Table 6 and Table 7 together suggest our model is more robust toward the input noise.

Table 5 shows an example translation (More examples are shown in the Appendix). In this example, the original and noisy input have liter-
4.5 Ablation Studies

Table 8 shows the importance of different components in our approach, which include $L_{\text{clean}}$, $L_{\text{robust}}$ and $L_{\text{lm}}$. As for $L_{\text{robust}}$, it includes the source adversarial input, i.e. $x' \neq x$ and the target source adversarial input, i.e. $z' \neq z$. In the fourth row with $x' = x$ and $z' \neq z$, we randomly choose replacement positions of $z$ since no changes in $x$ leads not to form the distribution in Eq. (10). We can find removing any component leads to a notable decrease in BLEU. Among those, the adversarial target input ($z' \neq z$) shows the greatest decrease of 1.87 BLEU points, and removing language models have the least impact on the BLEU score. However, language models are still important in reducing the size of the candidate set, regularizing word embeddings and generating fluent sentences.

The hyper-parameters $\gamma_{\text{src}}$ and $\gamma_{\text{trg}}$ control the ratio of word replacement in the source and target inputs. Table 9 shows their sensitive study result where the row corresponds to $\gamma_{\text{src}}$ and the column is $\gamma_{\text{trg}}$. As we see, the performance is relatively insensitive to the values of these hyper-parameters, and the best configuration on the Chinese-English validation set is obtained at $\gamma_{\text{src}} = 0.25$ and $\gamma_{\text{trg}} = 0.50$. We found that a non-zero $\gamma_{\text{trg}}$ always yields improvements when compared to the result of $\gamma_{\text{trg}} = 0$. While $\gamma_{\text{src}} = 0.25$ increases BLEU scores for all the values of $\gamma_{\text{trg}}$, a larger $\gamma_{\text{src}}$ seems to be damaging.

5 Related Work

Robust Neural Machine Translation Improving robustness has been receiving increasing attention in NMT. For example, Belinkov and Bisk (2018); Liu et al. (2018); Karpukhin et al. (2019); Sperber et al. (2017) focused on designing effective synthetic and/or natural noise for NMT using black-box methods. Cheng et al. (2018) proposed adversarial stability training to improve the robustness on arbitrary noise type. Ebrahimi et al. (2018a) used white-box methods to generate adversarial examples on character-level NMT. Different from prior work, our work uses a white-box method for the word-level NMT model and introduces a new method using doubly adversarial inputs to both attack and defend the model.

We noticed that Michel and Neubig (2018) proposed a dataset for testing the machine translation on noisy text. Meanwhile they adopt a domain adaptation method to first train a NMT model on a clean dataset and then finetune it on noisy data. This is different from our setting in which no noisy training data is available. Another difference is that one of our primary goals is to improve NMT models on the standard clean test data. This differs from Michel and Neubig (2018) whose goal is to improve models on noisy test data. We leave the extension to their setting for future work.

Adversarial Examples Generation Our work is inspired by adversarial examples generation, a popular research area in computer vision, e.g. in (Szegedy et al., 2014; Goodfellow et al., 2015; Moosavi-Dezfooli et al., 2016). In NLP, many authors endeavored to apply similar ideas to a variety of NLP tasks, such as text classification (Miyato et al., 2017; Ebrahimi et al., 2018b), machine comprehension (Jia and Liang, 2017), dialogue generation (Li et al., 2017), machine translation (Belinkov and Bisk, 2018), etc. Closely related to (Miyato et al., 2017) which attacked the text classification models in the embedding space, ours generates adversarial examples based on discrete word replacements. The experiments show that ours achieve better performance on both clean and
noisy data.

**Data Augmentation** Our approach can be viewed as a data-augmentation technique using adversarial examples. In fact, incorporating monolingual corpora into NMT has been an important topic (Sennrich et al., 2016b; Cheng et al., 2016; He et al., 2016; Edunov et al., 2018). There are also papers augmenting a standard dataset based on the parallel corpora by dropping words (Sennrich et al., 2016a), replacing words (Wang et al., 2018), editing rare words (Fadaee et al., 2017), etc. Different from these about data-augmentation techniques, our approach is only trained on parallel corpora and outperforms a representative data-augmentation work (Sennrich et al., 2016b) trained with extra monolingual data. When monolingual data is included, our approach yields further improvements.

6 Conclusion

In this work, we have presented an approach to improving the robustness of the NMT models with doubly adversarial inputs. We have also introduced a white-box method to generate adversarial examples for NMT. Experimental results on Chinese-English and English-German translation tasks demonstrate the capability of our approach to improving both the translation performance and the robustness. In future work, we plan to explore the direction to generate more natural adversarial examples dispensing with word replacements and more advanced defense approaches such as curriculum learning (Jiang et al., 2018, 2015).

References


