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).

Input	他(她)一个残疾人，我女儿身体好好地。
Original Output	**he is a** handicapped **person**, my daughter is in good health. ✓
Perturbed Output	**one of her** handicapped **people**, my daughter is in good health. ×

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

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". Perturbations originate from two sources: (a) natural noise in the annotation and (b) artificial deviations generated by attack models. In this paper, we do not distinguish the source of a perturbation and term perturbed examples as adversarial examples. The presence of such adversarial examples can lead to significant degradation of the generalization performance of the NMT model.

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

to use some synthetic noise and/or natural noise. Cheng et al. (2018) proposed adversarial stability training to improve the robustness on arbitrary noise type including feature-level and word-level noise. Liu et al. (2018) examined the homophonic noise for Chinese translation.

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 $\mathcal{S} = \{(\mathbf{x}^{(s)}, \mathbf{y}^{(s)})\}_{s=1}^{|\mathcal{S}|}$. 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 $\mathbf{x} = x_1, ..., x_I$ to a sequence of I word embeddings $e(\mathbf{x}) = e(x_1), ..., e(x_I)$. Then the word embeddings are encoded to their corresponding continuous hidden representations $\mathbf{h}$ by the transformation layer. Similarly, the decoder maps its target input sentence $\mathbf{z} = z_1, ..., z_J$ to a sequence of J word embeddings. For clarity, we denote the input and output in the decoder as $\mathbf{z}$ and $\mathbf{y}$. $\mathbf{z}$ is a shifted copy of $\mathbf{y}$ in the standard NMT model, *i.e.* $\mathbf{z} = \langle sos \rangle, \mathbf{y}_1, \cdots, \mathbf{y}_{J-1}$, where $\langle sos \rangle$ is a start symbol. Conditioned on the hidden representations $\mathbf{h}$ and the target input $\mathbf{z}$, the decoder generates $\mathbf{y}$ as:

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

where $\boldsymbol{\theta}_{mt}$ is a set of model parameters and $\mathbf{z}_{<j}$ is a partial target input. The training loss on $\mathcal{S}$ is defined as:

$$\mathcal{L}_{clean}(\boldsymbol{\theta}_{mt}) = \frac{1}{|S|} \sum_{(\mathbf{x},\mathbf{y}) \in S} -\log P(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_{mt}) \quad (2)$$

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 $\mathcal{Z}(\mathbf{x}, y)$ is generated with respect to a training sample $(\mathbf{x}, y)$ by solving an optimization problem:

$$\left\{\mathbf{x}' \,|\, \mathcal{R}(\mathbf{x}', \mathbf{x}) \leq \epsilon, \operatorname*{argmax}_{\mathbf{x}'} J(\mathbf{x}', y; \boldsymbol{\theta})\right\} \quad (3)$$

where $J(\cdot)$ measures the possibility of a sample being adversarial, and $\mathcal{R}(\mathbf{x}', \mathbf{x})$ 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 $\mathbf{x}'$. Although it is difficult to give a precise definition of the degree of imperceptibility $\mathcal{R}(\mathbf{x}', \mathbf{x})$, l_∞ 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 $\mathbf{x}'$ from the original input $\mathbf{x}$ to attack the NMT model. We then use *AdvGen* to find an adversarial target input $\mathbf{z}'$ from the decoder input $\mathbf{z}$ to improve the NMT model robustness to adversarial perturbations in the source input $\mathbf{x}'$. Thereby we hope the NMT model will be robust against both the source adversarial input $\mathbf{x}'$ and adversarial perturbations in target predictions $\mathbf{z}'$. 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 $(\mathbf{x}, \mathbf{y})$, according to Eq. (3), we generate a set of adversarial examples $\mathcal{A}(\mathbf{x}, \mathbf{y})$ specific to the NMT model by:

$$\left\{\mathbf{x}' \,|\, \mathcal{R}(\mathbf{x}', \mathbf{x}) \leq \epsilon, \operatorname*{argmax}_{\mathbf{x}'} -\log P(\mathbf{y}|\mathbf{x}'; \boldsymbol{\theta}_{mt})\right\} \quad (4)$$

where we use the negative log translation probability $-\log P(\mathbf{y}|\mathbf{x}'; \boldsymbol{\theta}_{mt})$ 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 $\mathbf{x}$, we induce a possible adversarial word x_i' for the word x_i in $\mathbf{x}$:

$$x_i' = \operatorname*{argmax}_{x \in \mathcal{V}_x} \operatorname{sim}\left(e(x) - e(x_i), \mathbf{g}_{x_i}\right) \quad (5)$$

$$\mathbf{g}_{x_i} = \nabla_{e(x_i)} -\log P(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}) \quad (6)$$

where $\mathbf{g}_{x_i}$ is a gradient vector wrt. $e(x_i)$, $\mathcal{V}_x$ is the vocabulary for the source language, and $\operatorname{sim}(\cdot, \cdot)$ 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 $Q(x_i, \mathbf{x}) \in \mathbb{R}^{|\mathcal{V}|}$ denote the likelihood of the i-th word in the sentence $\mathbf{x}$. Define $\mathcal{V}_{x_i} = top\_n(Q(x_i, \mathbf{x}))$ as the set of the n most probable words among the top n scores in terms of $Q(x_i, \mathbf{x})$, where n is a small constant integer and $|\mathcal{V}_{x_i}| \ll |\mathcal{V}_x|$. For the source, we estimate it from:

$$Q_{src}(x_i, \mathbf{x}) = P_{lm}(x|\mathbf{x}_{<i}, \mathbf{x}_{>i}; \boldsymbol{\theta}^x_{lm}) \quad (7)$$

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

Algorithm 1: The $AdvGen$ Function.

Input: $\mathbf{s}$: Input sentence, Q: Likelihood function, $\mathcal{D}_{pos}$: Distribution for word sampling, $\mathcal{L}$: translation loss.
Output: $\mathbf{s}'$: Output adversarial sentence

Function `AdvGen` ($\mathbf{s}$, Q, D_{pos}, $\mathcal{L}$) **:**
  $POS \leftarrow$ sample $\gamma|\mathbf{s}|$ positions from $\{1, ..., |\mathbf{s}|\}$ according to D_{pos} `// γ is a sampling ratio`
  foreach $i \in \{1, ..., |\mathbf{s}|\}$ **do**
    if $i \in POS$ **then**
      $\mathcal{V}_{s_i} \leftarrow top\_n(Q(s_i, \mathbf{s})) - \{s_i\}$;
      $\mathbf{g}_{s_i} \leftarrow \nabla_{e(s_i)}\mathcal{L}$;
      Compute s'_i by Eq. (5);
    else
      $s'_i \leftarrow s_i$;
    end
  end
  return $\mathbf{s}'$

language model can retain the semantic similarity between the original words and their adversarial counterparts to strengthen the constraint $\mathcal{R}$ in Eq. (4). Finally, it prevents word representations from being degenerative because replacements with adversarial words usually affect the context information around them.

Algorithm 1 describes the function $AdvGen$ for generating an adversarial sentence $\mathbf{s}'$ from an input sentence $\mathbf{s}$. The function inputs are: Q is a likelihood function for the candidate set generation, and for the source, it is Q_{src} from Eq. (7). D_{pos} is a distribution over the word position $\{1, .., |\mathbf{x}|\}$ from which the adversarial word is sampled. For the source, we use the simple uniform distribution $\mathcal{U}$. Following the constraint $\mathcal{R}$, we want the output sentence not to deviate too much from the input sentence and thus only change a small fraction of its constituent words based on a hyper-parameter $\gamma \in [0, 1]$.

3.2 Defense with Adversarial Target Inputs

After generating an adversarial example $\mathbf{x}'$, we treat $(\mathbf{x}', \mathbf{y})$ 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 $\mathbf{z}$ be the decoder input for the sentence pair $(\mathbf{x}, \mathbf{y})$. We use the same $AdvGen$ function to generate an adversarial target input $\mathbf{z}'$ from $\mathbf{z}$ by:

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

Note that for the target, the translation loss in Eq. (6) is replaced by $-\log P(\mathbf{y}|\mathbf{x}')$. 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 $P_{lm}(\mathbf{y}; \boldsymbol{\theta}^y_{lm})$ as follow:

$$\begin{aligned} Q_{trg}(z_i, \mathbf{z}) = \; & \lambda P(z|\mathbf{z}_{<i}, \mathbf{z}_{>i}; \boldsymbol{\theta}^y_{lm}) \\ & + (1-\lambda) P(z|\mathbf{z}_{<i}, \mathbf{x}'; \boldsymbol{\theta}_{mt}) \end{aligned} \quad (9)$$

where λ 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 $(\mathbf{x}, \mathbf{y}, \mathbf{x}')$ by:

$$P(j) = \frac{\sum_i \mathcal{M}_{ij} \delta(x_i, x'_i)}{\sum_k \sum_i \mathcal{M}_{ik} \delta(x_i, x'_i)}, j \in \{1, .., |\mathbf{y}|\} \quad (10)$$

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

3.3 Training

Algorithm 2 details the entire procedure to calculate the robustness loss for a parallel sentence pair $(\mathbf{x}, \mathbf{y})$. We run $AdvGen$ twice to obtain $\mathbf{x}'$ and $\mathbf{z}'$. 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:

$$\mathcal{L}_{robust}(\boldsymbol{\theta}_{mt}) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{S}} -\log P(\mathbf{y}|\mathbf{x}', \mathbf{z}'; \boldsymbol{\theta}_{mt}) \quad (11)$$

Algorithm 2: Computing Robustness Loss.

Input: $(\mathbf{x}, \mathbf{y})$: a parallel sentence pair
Output: $loss$: a robustness loss for $(\mathbf{x}, \mathbf{y})$

Function `RobustLoss(`$\mathbf{x}, \mathbf{y}$`)`**:**
- Initialize the sampling ratio γ_{src} and γ_{trg};
- Compute Q_{src} by Eq. (7);
- Set D_{src} as a uniform distribution;
- $\mathbf{x}' \leftarrow AdvGen(\mathbf{x}, Q_{src}, D_{src}, -\log P(\mathbf{y}|\mathbf{x}))$;
- Q_{trg} is computed as Eq. (9);
- D_{trg} is computed as Eq. (10);
- $\mathbf{z}' \leftarrow AdvGen(\mathbf{z}, Q_{trg}, D_{trg}, -\log P(\mathbf{y}|\mathbf{x}'))$;
- $loss \leftarrow -\log P(\mathbf{y}|\mathbf{x}', \mathbf{z}'; \boldsymbol{\theta}_{mt})$
- **return** $loss$

The final training objective $\mathcal{L}$ is a combination of four loss functions:

$$\mathcal{L}(\boldsymbol{\theta}_{mt}, \boldsymbol{\theta}^x_{lm}, \boldsymbol{\theta}^y_{lm}) = \mathcal{L}_{clean}(\boldsymbol{\theta}_{mt}) + \mathcal{L}_{lm}(\boldsymbol{\theta}^x_{lm}) + \mathcal{L}_{robust}(\boldsymbol{\theta}_{mt}) + \mathcal{L}_{lm}(\boldsymbol{\theta}^y_{lm}) \quad (12)$$

where $\boldsymbol{\theta}^x_{lm}$ and $\boldsymbol{\theta}^y_{lm}$ are two sets of model parameters for source and target bidirectional language models, respectively. The word embeddings are shared between $\boldsymbol{\theta}_{mt}$ and $\boldsymbol{\theta}^x_{lm}$ and likewise between $\boldsymbol{\theta}_{mt}$ and $\boldsymbol{\theta}^y_{lm}$.

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 compromises 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 $AdvGen$, We modified multiple positions in the source and target input sentences in parallel. The bidirectional language model used in $AdvGen$ 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, λ was set to 0.5. The n in 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-German translation. We compare our approach with Transformer for different numbers of hidden

Method	Model	MT06	MT02	MT03	MT04	MT05	MT08
Vaswani et al. (2017)	Trans.-Base	44.59	44.82	43.68	45.60	44.57	35.07
Miyato et al. (2017)	Trans.-Base	45.11	45.95	44.68	45.99	45.32	35.84
Sennrich et al. (2016a)	Trans.-Base	44.96	46.03	44.81	46.01	45.69	35.32
Wang et al. (2018)	Trans.-Base	45.47	46.31	45.30	46.45	45.62	35.66
Cheng et al. (2018)	RNMT$_{lex.}$	43.57	44.82	42.95	45.05	43.45	34.85
	RNMT$_{feat.}$	44.44	46.10	44.07	45.61	44.06	34.94
Cheng et al. (2018)	Trans.-Base$_{feat.}$	45.37	46.16	44.41	46.32	45.30	35.85
	Trans.-Base$_{lex.}$	45.78	45.96	45.51	46.49	45.73	36.08
Sennrich et al. (2016b)*	Trans.-Base	46.39	47.31	47.10	47.81	45.69	36.43
Ours	Trans.-Base	46.95	47.06	46.48	47.39	46.58	37.38
Ours + BackTranslation*	Trans.-Base	**47.74**	**48.13**	**47.83**	**49.13**	**49.04**	**38.61**

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

Method	Model	MT06	MT02	MT03	MT04	MT05	MT08
Vaswani et al. (2017)	Trans.-Base	44.59	44.82	43.68	45.60	44.57	35.07
Ours	Trans.-Base	**46.95**	**47.06**	**46.48**	**47.39**	**46.58**	**37.38**

Table 3: Results on NIST Chinese-English translation.

Method	Model	BLEU
Vaswani et al.	Trans.-Base	27.30
	Trans.-Big	28.40
Chen et al.	RNMT+	28.49
Ours	Trans.-Base	28.34
	Trans.-Big	**30.01**

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

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 re-trained the Chinese-English model using back-translated data as well as original parallel data.

Input & Noisy Input	这体现了中俄两国和两国议会间密切(紧密)的友好合作关系。
Reference	this expressed the relationship of close friendship and cooperation between China and Russia and between our parliaments.
Vaswani et al. on Input	this reflects the close friendship and cooperation between **China and Russia** and **between** the parliaments **of** the two countries.
Vaswani et al. on Noisy Input	this reflects the close friendship and cooperation between the two countries and the **two** parliaments.
Ours on Input	this **reflects** the close relations of friendship and cooperation between China and Russia and between their parliaments.
Ours on Noisy Input	this **embodied** the close relations of friendship and cooperation between China and Russia and between their parliaments.

Table 5: Comparison of translation results of Transformer and our model for an input and its perturbed input.

Method	0.00	0.05	0.10	0.15
Vaswani et al.	44.59	41.54	38.84	35.71
Miyato et al.	45.11	42.11	39.39	36.44
Cheng et al.	45.78	42.90	40.58	38.46
Ours	**46.95**	**44.20**	**41.71**	**39.89**

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

Method	0.00	0.05	0.10	0.15
Vaswani et al.	100	77.08	62.00	52.50
Miyato et al.	100	79.19	63.12	53.51
Cheng et al.	100	79.66	65.16	56.11
Ours	100	**82.76**	**69.23**	**60.70**

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

Table 2 shows the comparisons to the above five baseline methods. Among all methods trained without extra corpora, our approach achieves the best result across datasets. After incorporating the back-translated corpus, our method yields an additional gain of 1-3 points over (Sennrich et al., 2016b) trained on the same back-translated corpus. Since all methods are built on top of the same backbone, the result substantiates the efficacy of our method on the standard benchmarks that contain natural noise. Compared to (Miyato et al., 2017), we found that continuous gradient-based perturbations to word embeddings can be absorbed quickly, often resulting in a worse BLEU score than the proposed discrete perturbations by word replacement.

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-

$\mathcal{L}_{clean}$	$\mathcal{L}_{robust}$		$\mathcal{L}_{lm}$	BLEU
	$\mathbf{x}' \neq \mathbf{x}$	$\mathbf{z}' \neq \mathbf{z}$		
✓				44.59
✓			✓	45.08
✓	✓		✓	45.23
✓		✓	✓	46.26
✓	✓	✓		46.61
✓	✓	✓	✓	**46.95**

Table 8: Ablation study on Chinese-English translation. ✓ means that it is included in training.

γ_{src} \ γ_{trg}	0.00	0.25	0.50	0.75
0.00	44.59	46.19	46.26	46.14
0.25	45.23	46.72	**46.95**	46.52
0.50	44.25	45.34	45.39	45.94
0.75	44.18	44.98	45.35	45.37

Table 9: Effect of the ratio value γ_{src} and γ_{trg} on Chinese-English Translation.

ally the same meaning, where "密切" and "紧密" both mean "close" in Chinese. Our model retains very important words such as "China and Russia", which are missing in the Transformer results.

4.5 Ablation Studies

Table 8 shows the importance of different components in our approach, which include $\mathcal{L}_{clean}$, $\mathcal{L}_{robust}$ and $\mathcal{L}_{lm}$. As for $\mathcal{L}_{robust}$, it includes the source adversarial input, *i.e.* $\mathbf{x}' \neq \mathbf{x}$ and the target source adversarial input, *i.e.* $\mathbf{z}' \neq \mathbf{z}$. In the fourth row with $\mathbf{x}' = \mathbf{x}$ and $\mathbf{z}' \neq \mathbf{z}$, we randomly choose replacement positions of $\mathbf{z}$ since no changes in $\mathbf{x}$ 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 ($\mathbf{z}' \neq \mathbf{z}$) 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 γ_{src} and γ_{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 γ_{src} and the column is γ_{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 $\gamma_{src} = 0.25$ and $\gamma_{trg} = 0.50$. We found that a non-zero γ_{trg} always yields improvements when compared to the result of $\gamma_{trg} = 0$. While $\gamma_{src} = 0.25$ increases BLEU scores for all the values of γ_{trg}, a larger γ_{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 attach 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

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Empirical Methods in Natural Language Processing*.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Ankur Bapna, Mia Xu Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training deeper neural machine translation models with transparent attention. *arXiv preprint arXiv:1808.07561*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Association for Computational Linguistics*.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Association for Computational Linguistics*.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. In *Association for Computational Linguistics*.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018a. On adversarial examples for character-level neural machine translation. In *Proceedings of COLING*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018b. Hotflip: White-box adversarial examples for text classification. In *Association for Computational Linguistics*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Empirical Methods in Natural Language Processing*.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Association for Computational Linguistics*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Empirical Methods in Natural Language Processing*.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. 2015. Self-paced curriculum learning. In *AAAI Conference on Artificial Intelligence*.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. *arXiv preprint arXiv:1902.01509*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Empirical Methods in Natural Language Processing*.

Hairong Liu, Mingbo Ma, Liang Huang, Hao Xiong, and Zhongjun He. 2018. Robust neural machine translation with joint textual and phonetic embedding. *arXiv preprint arXiv:1810.06729*.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. *arXiv preprint arXiv:1809.00388*.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *International Conference on Learning Representations*.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a methof for automatic evaluation of machine translation. In *Association for Computational Linguistics*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving nerual machine translation models with monolingual data. In *Association for Computational Linguistics*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Association for Computational Linguistics*.

Matthias Sperber, Jan Niehues, and Alex Waibel. 2017. Toward robust neural machine translation for noisy input sequences. In *International Workshop on Spoken Language Translation*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*.

Christian Szegedy, Wojciech Zaremba, Sutskever Ilya, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Machine Learning*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. Switchout: an efficient data augmentation algorithm for neural machine translation. In *Empirical Methods in Natural Language Processing*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *International Conference on Learning Representations*.