PAEG: Phrase-level Adversarial Example Generation for Neural Machine Translation

Juncheng Wan[1*], **Jian Yang**[2*], **Shuming Ma**[3], **Dongdong Zhang**[3], **Weinan Zhang**[1†], **Yong Yu**[1], **Zhoujun Li**[2]

[1]Shanghai Jiao Tong University
[2]State Key Lab of Software Development Environment, Beihang University
[3]Microsoft Research Asia
{junchengwan,wnzhang,yyu}@apex.sjtu.edu.cn;
{jiaya, lizj}@buaa.edu.cn; {shumma, dozhang}@microsoft.com

Abstract

While end-to-end neural machine translation (NMT) has achieved impressive progress, noisy input usually leads models to become fragile and unstable. Generating adversarial examples as the augmented data has been proved to be useful to alleviate this problem. Existing methods for adversarial example generation (AEG) are word-level or character-level, which ignore the ubiquitous phrase structure. In this paper, we propose a **P**hrase-level **A**dversarial **E**xample **G**eneration (PAEG) framework to enhance the robustness of the translation model. Our method further improves the gradient-based word-level AEG method by adopting a phrase-level substitution strategy. We verify our method on three benchmarks, including LDC Chinese-English, IWSLT14 German-English, and WMT14 English-German tasks. Experimental results demonstrate that our approach significantly improves translation performance and robustness to noise compared to previous strong baselines.

1 Introduction

Recently, neural machine translation (NMT) has effectively improved translation quality. NMT has shown state-of-the-art performance for many language pairs (Wu et al., 2016; Hassan et al., 2018; Vaswani et al., 2017). Various architectures (Sutskever et al., 2014; Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017) bring many appealing properties. Most NMT systems heavily rely on high-quality parallel data and perform poorly in noisy input. With the noise rising in the source sentence, NMT tends to be more vulnerable (Szegedy et al., 2014; Goodfellow et al., 2015), due to the output prediction of the decoder easily intervened by the other words (Cheng et al., 2018). A slight disturbance like a random permutation can damage the translation quality dramatically (Belinkov and Bisk, 2018). Even replacing a word with a synonym in the source input, the NMT model can be cheated and the target output can not be translated correctly.

Original Sentence	A cooked hot dog in a bun with ketchup and relish.
Word-level AEG	A cooked ***warm*** dog in a bun with ketchup and relish.
Phrase-level AEG	A cooked ***sausage rolls*** in a bun with ketchup and relish.

Table 1: An example of adversarial example generation (AEG). When the word "hot" is selected, word-level adversarial example generation method substitutes "hot" to "warm". The phrase-level method substitutes the whole phrase "hot dog" to "sausage rolls".

To improve the robustness of the NMT model, previous works propose to construct the adversarial examples by manipulating hidden features or discrete text input. These adversarial examples are used as augmented data for the training of the NMT model. To attack hidden features, Cheng et al. (2018) added perturbations in the input at the feature level for adversarial stability training. To generate discrete adversarial input, Ebrahimi et al. (2018) employed differentiable string-edit operations to rank adversarial changes. Belinkov and Bisk (2018) and Vaibhav et al. (2019) emulated naturally occurring errors in clean data as synthetic noise. Cheng et al. (2019) proposed a gradient-based method to craft adversarial examples, considering the similarity between the gradient related to the translation loss of input and the embedding difference of words.

Previous methods of adversarial example generation (AEG) are limited at the low level, like word-level, not considering the relationship between different words within a phrase. There is one example in Table 1. The word-level AEG method selects a

[*]Equal Contribution.
[†]Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 5085–5097
October 12–17, 2022.

vulnerable position then substitutes the corresponding word, omitting that the substituted word is in a phrase. Sometimes, the examples of this improper substitution can even harm the translation model.

Therefore, we propose a phrase-level adversarial example generation (PAEG) method, which improves a gradient-based word-level AEG method to phrase-level. Specifically, this method builds phrase-level candidates efficiently and substitutes phrases wholly with these candidates. We also propose to further improve this method with a bidirectional generation algorithm, as target-to-source adversarial pairs are a kind of slight perturbation of the original source-to-target translation. In practice, we generate adversarial examples after fixed intervals of NMT model updating (to convergence) and use them as new augmented data for the continual training of the model.

To verify the effectiveness of our method, we conduct experiments on three common benchmarks, i.e, LDC Chinese-English, IWSLT14 German-English, and WMT14 English-German. Experimental results demonstrate our method achieves significant improvements on translation quality and robustness to noisy inputs over the previous baselines including outstanding adversarial examples generation methods.

2 Phrase-level Adversarial Example Generation

In this section, we formulate the problem of adversarial example generation mathematically. First, our proposed method provides reliable candidates with the pre-trained model. Then, we use the gradient-based method to select vulnerable positions and substitutes at phrase level to generate adversarial examples. These examples are used as augmented data for the training of the NMT model. To further improve the performance, we extend our method to the bidirectional generation.

2.1 Problem Formulation

Let $A = \{(\mathbf{x}, \mathbf{z}), \mathbf{y}\}$ denotes the training data in NMT, where $(\mathbf{x}, \mathbf{z})$ are the encoder input and the decoder input, $\mathbf{y}$ the corresponding decoder output. To generate the corresponding adversarial examples $B = \{(\mathbf{x}', \mathbf{z}'), \mathbf{y}\}$, where only the input is slightly different from A, we need to limit the adversarial input $(\mathbf{x}', \mathbf{z}')$ semantically close to the original data.

Adversarial examples aim to cheat the model, making it predict wrong words. Therefore, given real output words $\mathbf{y}$, we construct an input to make the model predict the incorrect word $\mathbf{y}'(\mathbf{y}' \neq \mathbf{y})$. The process of adversarial example generation in NMT can be formulated as solving the following optimization problem:

$$\{(\mathbf{x}', \mathbf{z}') : \arg\max_{(\mathbf{x}', \mathbf{z}')} P(\mathbf{x}', \mathbf{z}'; \mathbf{y}, \theta), \quad dist((\mathbf{x}', \mathbf{z}'), (\mathbf{x}, \mathbf{z})) < \epsilon\} \tag{1}$$

where $dist$ is a measure function of the input, such as the semantic distance of sentence embeddings or edit distance, $P(\mathbf{x}', \mathbf{z}'; \mathbf{y}, \theta)$ is the maximal probability that the model predicts a wrong word $\mathbf{y}'$ such that $\mathbf{y}' \neq \mathbf{y}$ when the model is fed with $(\mathbf{x}', \mathbf{z}')$, θ is the model parameters, and ϵ is a sufficiently small distance.

Algorithm 1 Phrase-level Adversarial Example

Input: $\{(\mathbf{x}, \mathbf{z}), \mathbf{y}\}$ denotes input and output, θ_l^s and θ_l^t denote parameters of LMs, θ_m denotes parameters of the model, $\mathbb{D}$ denotes the phrase dictionary.
Output: phrase-level adversarial input: $(\mathbf{x}', \mathbf{z}')$.

```
Compute $\{\mathbf{g}_{x_i}\}_{i=1}^{|\mathbf{x}|}$ with x, z, y by Eq.(2).
$pos_x \longleftarrow$ positions of maximal $\{||\mathbf{g}_{x_i}||_2\}_{i=1}^{|\mathbf{x}|}$
for $i$ in $pos_x$ do
    Get $cand(\mathbf{x}_{ij})$ by $\mathbb{D}$ and $\theta_l^s$.
    Substitute $\mathbf{x}_{ij}$ to $\mathbf{x}'_{ij}$ as Eq.(4).
end for
Compute $\{\mathbf{g}_{z_i}\}_{i=1}^{|\mathbf{z}|}$ with x', z, y by Eq.(2).
Get attention matrix $\mathcal{M}$ by x', z, y, and $\theta_m$.
Compute $\{P(j)\}_{j=1}^{|\mathbf{y}|}$ with $\mathcal{M}$ by Eq.(3).
$pos_z \longleftarrow$ sampling by $\{P(j)\}_{j=1}^{|\mathbf{y}|}$
for $i$ in $pos_z$ do
    Get $cand(\mathbf{z}_{ij})$ by $\mathbb{D}$, $\theta_l^t$ and $\theta_m$.
    Substitute $\mathbf{z}_{ij}$ to $\mathbf{z}'_{ij}$ as Eq.(4).
end for
return $(\mathbf{x}', \mathbf{z}')$
```

2.2 Phrase Candidates from PLM

To guarantee the generated example $(\mathbf{x}', \mathbf{z}')$ is similar to the original example $(\mathbf{x}, \mathbf{z})$, two aspects are taken into account. One aspect is that the information in sentences should not change a lot. The other is to guarantee that words are similar. Therefore, high-quality candidates for words or phrases to be substituted should have similar semantic meanings to their original ones and be more fluent in the whole sentence.

To achieve this, one intuitive method is to select words with maximal prediction probability in the language model (LM), since LM predicts words based on the context. Cheng et al. (2019) uses a bidirectional LM trained on the monolingual part of the parallel corpus. However, a high-quality LM often needs billions of monolingual data to train like BERT (Devlin et al., 2019). It is unacceptable to spend much time and computational resources training reliable LMs. Therefore, we propose to utilize the knowledge of the pre-trained LM (PLM). In this paper, we use BERT as PLM.

In our paper, we use the notation $\mathbf{x}_{ij}$ as the phrase from position i to j in sentence $\mathbf{x}$, $cand(\mathbf{x}_{ij})$ as the phrase candidates of phrase $\mathbf{x}_{ij}$. When $i = j$, $\mathbf{x}_{ij}$ indicates the word x_i and $cand(\mathbf{x}_{ij})$ indicates the word candidates $cand(x_i)$. Besides, we use $\mathbb{D}_n$ as the n-gram phrase dictionary and $\mathbb{D}$ the union set of all $\mathbb{D}_n$.

In the i^{th} position of the source input, we construct $cand(x_i)$ by selecting the top n_s tokens with maximal prediction probability in BERT when fed with $\mathbf{x}$, where x_i is masked. For the target input side, candidates consist of two parts. The first part is from BERT, which provides with n_t^l candidates. The second part is from the trained NMT model, which provides with n_t^m candidates. In this way, the candidate set $cand(z_j)$ of target input side consists of words fluent in the sentence and words conforming the translation of $\mathbf{x}$. In this paper, we set $n_s^l = 10$ and $n_t^l = n_t^m = 5$.

Given a phrase $\mathbf{x}_{ij}$, we construct phrase-level candidates $cand(\mathbf{x}_{ij})$. We first build the set of all probable phrase candidates as the Cartesian product of all $cand(x_k)$ $(k = i, i+1, \dots, j)$. Then, we screen out unreasonable phrase candidates by the phrase dictionary $\mathbb{D}$. Candidates not in this dictionary are discarded.

To obtain the phrase dictionary $\mathbb{D}$, we introduce two methods. The first one is to use the syntax parser to parse the sentence into a syntax tree. Then, the leaf nodes of an n-leaf subtree is an n-gram phrase. The phrase dictionary $\mathbb{D}$ is the union of these n-gram phrase dictionary $\mathbb{D}_n$. The second method is to utilize the existing phrase extraction tool directly. In this paper, we take both of these two methods. The syntax parser we used is `nltk.parse`[1] and $n = 2, 3, 4$. The phrase extraction tool we used is `TextBlob`[2].

2.3 Select Vulnerable Positions

Instead of randomly selecting positions, we propose that the adversarial examples should select the most vulnerable positions in the sentence. Given a certain sentence, some NMT models may get worse translations when certain words or phrases are substituted.

Given that we train an NMT model with parameters θ_m and use negative log likelihood as the loss function with the input $\mathbf{x}$, $\mathbf{z}$ and the output $\mathbf{y}$, we can get the gradient vector $\mathbf{g}_{x_i}$ of token x_i over the training loss:

$$\mathbf{g}_{x_i} = \nabla_{e(x_i)} - \log P(\mathbf{y}|\mathbf{x}, \mathbf{z}; \theta_m) \qquad (2)$$

where $e(x_i)$ is the embedding vector of token x_i.

Previous methods randomly choose positions in the source input. Since different positions have different gradient norms $||\mathbf{g}_{x_i}||_2$, if the gradient norm is large, the position is more unstable. Therefore, positions with large gradient norm are more vulnerable. For the source input, we select the top $\alpha_s|\mathbf{x}|$ positions with maximal gradient norm, where $\alpha_s \in (0, 1)$ is a ratio. [3]

To construct the target input $\mathbf{z}'$, we teach the model how to defend the attack from the source $\mathbf{x}'$. It is a reason that we choose n_t^m candidates from the NMT model on the target side. Selected target side positions should have the target counterpart of substituted source words in $\mathbf{x}'$. For example, if we substitute the word "drawing" to "eating" in the source input "Cezanne loved drawing apples ." ("Cezanne malt gerne äpfel ." in German), then we need to find the position of the corresponding translation "drawing" ("malt") and substitute it to an English word related to "eating", such as "isst".

This process is the inverse process of attention in NMT. Following (Cheng et al., 2019), we sample $\alpha_t|\mathbf{y}|$ $(\alpha_t \in (0, 1))$ relevant words influenced by the perturbed words in the source input $\mathbf{x}'$ as by sampling function $P(\cdot)$:

$$P(j) = \frac{\sum_i \mathcal{M}_{ij}\delta_{x_i \neq x'_i}}{\sum_k \sum_i \mathcal{M}_{ik}\delta_{x_i \neq x'_i}}, j = 1, \dots, |\mathbf{y}| \qquad (3)$$

where $\mathcal{M}_{ij}$ is the value of attention matrix between token x_i and token y_j from NMT model, $\delta_{x_i \neq x'_i}$ is 1 if $x_i \neq x'_i$ and 0 otherwise.

[1] https://www.nltk.org

[2] https://textblob.readthedocs.io/en/dev/

[3] The experimental results of how to choose positions are discussed in Appendix A.1.

2.4 Phrase-level Substitution

Since words in the same phrase have a close relationship, we substitute words at phrase level in the adversarial example generation. There are two aspects to consider. First, since synonymy phrases sharing the same meaning may have variant lengths, the feature representation of a phrase should be irrelevant to the length of the phrase. Besides, we need to choose the phrase from the candidate set that disturbs the model the most.

For the first consideration, we simply extract phrase-level features by averaging the word embeddings. For the second aspect, we adopt the gradient-based approach in (Cheng et al., 2019). To represent the whole gradient of the phrase, we also average all the gradients of words. Other feature engineering methods like max-pooling, concatenation, and element-wise product are also viable.

Formally, for the substitution of phrase $\mathbf{x}_{ij}$, the greedy approach based on the gradient is:

$$\mathbf{x}'_{ij} = \underset{\mathbf{c}\in cand(\mathbf{x}_{ij})}{\arg\max}\ sim(f^e(\mathbf{c}) - f^e(\mathbf{x}_{ij}), f^g(\mathbf{c})) \tag{4}$$

where sim is the similarity function, f^e is the feature representation of the phrase, f^g is the feature representation of the gradient of the phrase, and $cand(\mathbf{x}_{ij})$ the phrase candidates of $\mathbf{x}_{ij}$. In this paper, we use the "average" function for f^e and f^g [4] and cosine similarity as the similarity function.

Our phrase-level adversarial example generation process is shown in Algorithm 1. During the training of the NMT model, we generate adversarial examples periodically as augmented data. Note that we do not need training LMs for the source and target languages.

2.5 Bidirectional Generation

In practice, reversed adversarial examples from target-to-source translation can also be used as augmented data for the source-to-target translation. Therefore, we introduce a bidirectional generation method to boost our phrase-level adversarial example generation method.

Our bidirectional generation has two translation directions, source-to-target, and target-to-source. We use a universal encoder and decoder for these two directions as (Johnson et al., 2017). From the original data, we generate the adversarial examples for two directions. In each iteration, the adversarial examples are reversed and added to the dataset. The model is trained on the augmented dataset.

Algorithm 2 Bidirectional Generation

Input: $\{(\mathbf{x},\mathbf{z}_l),\mathbf{y}\}$ denotes source-to-target input and output. $\{(\mathbf{y},\mathbf{z}_r),\mathbf{x}\}$ denotes target-to-source input and output. **Gen** is the adversarial examples generator.

Output: augmented source-to-target data D_l and target-to-source data D_r.

```
Compute $\{\mathbf{g}_{x_i}\}_{i=1}^{|\mathbf{x}|}$ with x, z, y by Eq.(2).
$D_l \longleftarrow \{(\mathbf{x},\mathbf{z}_l),\mathbf{y}\}$, $D_r \longleftarrow \{(\mathbf{y},\mathbf{z}_r),\mathbf{x}\}$.
$\mathbf{x}',\mathbf{z}'_l \longleftarrow \mathrm{Gen}(\mathbf{x},\mathbf{z}_l)$, $\mathbf{y}',\mathbf{z}'_r \longleftarrow \mathrm{Gen}(\mathbf{y},\mathbf{z}_r)$
Add $\{(\mathbf{x}',\mathbf{z}'_l),\mathbf{y}\} \bigcup \{(\mathbf{z}'_r,\mathbf{y}'),\mathbf{y}\}$ to $D_l$ and
  add $\{(\mathbf{y}',\mathbf{z}'_r),\mathbf{x}\} \bigcup \{(\mathbf{z}'_l,\mathbf{x}'),\mathbf{x}\}$ to $D_r$.
return $(D_l, D_r)$
```

Formally, we notate $(\mathbf{x},\mathbf{z}_l,\mathbf{y})$ as the encoder input, decoder input and decoder output for source-to-target translation, ($\mathbf{y}$, $\mathbf{z}_r$, $\mathbf{x}$) as the encoder input, decoder input and decoder output for target-to-source translation. After generating the adversarial examples, we get $(\mathbf{x}',\mathbf{z}'_l,\mathbf{y})$ and $(\mathbf{y}',\mathbf{z}'_r,\ \mathbf{x})$. Then, the adversarial examples input are reversed and added to the training data of the other direction. For source-to-target training, we have three pairs of data $(\mathbf{x},\mathbf{z},\mathbf{y})$, $(\mathbf{x}',\mathbf{z}'_l,\mathbf{y})$, $(\mathbf{z}'_r,\mathbf{y}',\mathbf{y})$. They are respectively the original training data, the adversarial examples and the reversed adversarial examples from the other direction.

The phrase-level adversarial example generation of these two directions help mutually during the training. Our bidirectional generation algorithm is shown in Algorithm 2. It is worth noting that, in general, the training time of PAEG is not as much as double of PAEG without bidirectional generation, as the data from bidirectional generation has a similar distribution of the original adversarial samples.[5]

3 Experiments

We evaluate our method on three datasets, LDC Chinese-English, IWSLT14 German-English, and WMT14 English-German translation datasets. Then, we compare our method with baselines. At last, we do a detailed analysis of the different components of our method.

Limited by the number of pages, we have included the description of three datasets and the

[4] The reason for using "average" function is explained in Appendix A.2.

[5] There are more discussions about time consumption of bidirectional generation in Appendix 4.

Method	MT06	MT02	MT03	MT05	MT08	MT12	Avg.
Transformer (Vaswani et al., 2017)	43.52	43.17	44.06	44.45	36.27	35.07	41.09
Multilingual NMT (Johnson et al., 2017)	43.54	43.46	44.63	44.40	36.13	35.00	41.19
Word Dropout (Sennrich et al., 2016)†	43.96	44.02	44.55	44.70	36.49	35.33	41.51
SwitchOut (Wang et al., 2018a)†	43.83	44.36	45.02	44.85	36.53	35.45	41.67
AdvGen (Cheng et al., 2019)†	44.74	45.12	46.49	45.95	37.29	36.02	42.60
PAEG (this work)†	**45.49**	**45.76**	**47.58**	**46.83**	**38.18**	**36.91**	**43.46**

Table 2: Case-insensitive BLEU-4 scores (%) on LDC Zh→En task. Our method is compared with other baselines and *Transformer_base* model. Methods with "†" use adversarial examples for training.

Method	BLEU
Transformer (Vaswani et al., 2017)	34.20
Multilingual NMT (Johnson et al., 2017)	34.13
NT^{2}MT (Feng et al., 2018)	31.75
LightConv (Wu et al., 2019)	34.80
DynamicConv (Wu et al., 2019)	35.20
Word Dropout (Sennrich et al., 2016)†	34.72
SwitchOut (Wang et al., 2018a)†	34.83
AdvGen (Cheng et al., 2019)†	35.25
PAEG (this work)†	**35.65**

Table 3: Case-insensitive BLEU-4 scores (%) on IWSLT14 De→En task. Our method is compared with other baselines and *Transformer_small* model. Methods with "†" use adversarial examples for training.

Method	BLEU
Transformer (Vaswani et al., 2017)	28.40
Multilingual NMT (Johnson et al., 2017)	29.11
RNMT+ (Chen et al., 2018)	28.49
LightConv (Wu et al., 2019)	28.90
DynamicConv (Wu et al., 2019)	29.70
Word Dropout (Sennrich et al., 2016)†	29.30
SwitchOut (Wang et al., 2018a)†	29.40
AdvGen (Cheng et al., 2019)†	30.01
PAEG (this work)†	**30.49**

Table 4: Case-insensitive BLEU-4 scores (%) on WMT14 En→De task. Our method is compared with other baselines and *Transformer_big* model. Methods with "†" use adversarial examples for training.

training details in the Appendix B and Appendix C respectively.

3.1 Comparisons to Baseline Methods

We compare our method with NMT models without adversarial examples (Non-adv NMT) and using adversarial examples (Adv NMT). Our method gets significant translation improvement by statistical significance testing ($p < 0.05$) compared to relevant baselines.

Non-adv NMT **Multilingual NMT** (Johnson et al., 2017) is implemented with the Transformer model as the universal encoder and decoder. **NT^{2}MT** (Feng et al., 2018) uses a phrase attention mechanism with backbone model LSTM. We report the maximal result with out-of-domain dictionaries in the paper. **RNMT+** (Chen et al., 2018) is an enhanced version of RNN-based NMT model. **LightConv** (Wu et al., 2019) uses a lightweight convolution performing competitively to the Transformer. **DynamicConv** (Wu et al., 2019) leverages a dynamic convolution predicting separate convolution kernels.

Adv NMT **Word Dropout** (Sennrich et al., 2016) drops words randomly. We implement it on the token level, as recommended by the paper. **SwitchOut** (Wang et al., 2018a) randomly replaces words in both the source and target sentence with words from the vocabulary. We implement the hamming distance sampling method in the paper. **AdvGen** (Cheng et al., 2019) is an adversarial example generation method at the word-level. This method uses doubly adversarial input. We implement this method with the Transformer backbone, $\alpha_s = 25\%, \alpha_t = 50\%$ for LDC Chinese-English task, and $\alpha_s = 20\%, \alpha_t = 20\%$ for IWSLT14 German-English and WMT14 English-German.

Table 2 demonstrates the comparisons between our method with the above five baseline methods on LDC Chinese-English translation task. First, we compare our method with the Transformer. On average, PAEG can improve +2.37 BLEU points significantly. Then, we compare our method with methods of training with adversarial examples. On average, adversarial example generation methods (AdvGen and PAEG) utilizing the training information of the model greatly surpass the other methods (Word Dropout and SwitchOut). The reason is that the former approach is better at attacking vulnerable parts of the NMT model. Compared with the

Method	MT06	MT02	MT03	MT05	MT08	MT12	Avg.
PAEG	45.49	45.76	47.58	46.83	38.18	36.91	43.46
w/o bidirectional generation	45.52	45.53	46.96	46.72	38.10	36.85	43.24
w/o phrase-level substitution	44.03	44.02	45.63	45.35	37.21	35.51	41.96
w/o candidates from BERT	43.52	43.17	44.06	44.45	36.27	35.07	41.09

Table 5: Experiments on LDC Zh→En dataset to analyze the effect of different components of PAEG. We removed three components of PAEG step by step. The results show that phrase-level substitution is the most effective part.

state-of-the-art AEG method AdvGen, PAEG gets an improvement of +0.86 BLEU points.

In Table 3, we compare our method with the above eight baseline methods on the IWSLT14 German-English translation task. Compared with the backbone model Transformer, PAEG gets the gain of +1.45 BLEU points. Compared with methods built on top of Transformer, NT^{2}MT (Feng et al., 2018) with out-of-domain dictionaries suffers from a worse backbone model (LSTM). Multilingual NMT (Johnson et al., 2017) has a similar performance to the Transformer model. Compared with the other methods of training with adversarial examples, PAEG has the best performance. PAEG gets +0.8∼0.9 BLEU points improvement compared with AEG methods which do not leverage the training information of the model.

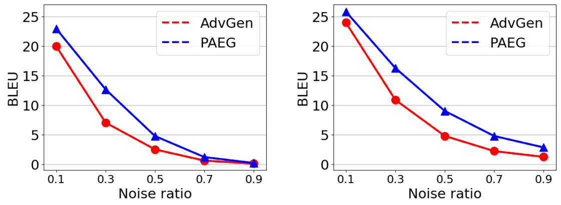
The comparisons on the WMT14 English-German task are in Table 4. Compared with *Transformer_big* model, PAEG has a notable gain of +2.09 BLEU points. PAEG consistently outperforms all three baselines training with adversarial examples, having around +0.5∼1.0 BLEU points improvement in this commonly used dataset.

3.2 Ablation Studies

Our proposed method PAEG is mainly affected by three components, the use of the pre-trained model, the phrase-level substitution, and the bidirectional adversarial example generation. We analyze the different components of PAEG by ablation studies.

Effect of Phrase-level Substitution We use the phrase-level substitution and there is +1.28 BLEU points improvement in Table 5, which is significant. Substituting words randomly from the top 10 word-level candidates can not guarantee consistency between words. What is worse is that random substitution may destroy the phrase structure and semantic consistency in the sentence.

For common languages, such as Chinese, English, and German, the ratio of phrases is non-negligible. Substituting at the phrase level does make the adversarial input more fluent and thus more closely approximates the real-world data. In this way, our method can teach the model to defend against the attack on the target side better.



(a) (b)

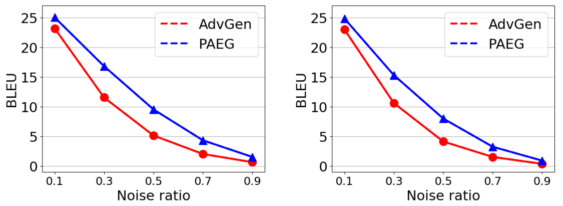
Figure 1: Results on (a) random replacement noise and (b) random switch noise.

Effect of PLM To find out the impact of the pre-trained model, we use BERT to generate pseudo data. In Table 5, with the use of the pre-trained model BERT, the Transformer model has +0.87 BLEU points improvement. This proves that BERT provides more reliable candidates by pre-training on amounts of data. Compared with the LMs trained on millions of monolingual data, BERT can significantly leverage the contextual information to make the candidates appear fluent in the sentence.

Effect of Bidirectional Generation In Table 5, we add the bidirectional generation method to PAEG and there is +0.22 BLEU points improvement. This shows that the bidirectional generation has slight improvements. Considering that PAEG (without bidirectional generation) itself achieves a high BLEU score, the further improvement of bidirectional generation cannot be ignored.

3.3 Robustness to Noisy Inputs

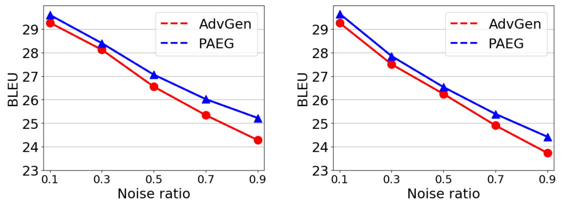
To compare the robustness of different NMT models, we conduct three groups of experiments to simulate machine translation scenarios with noisy inputs by word replacement and switch. All experiments are conducted in the WMT14 English-German test set. Our method is compared with the representative word-level augmentation method AdvGen with the *Transformer_big* backbone.



(a)

(b)

Figure 2: Results on (a) word-level most similar synonym noise and (b) word-level least similar synonym noise.



(a)

(b)

Figure 3: Results on (a) phrase-level most similar synonym noise and (b) phrase-level least similar synonym noise.

Word Type	Word-level	Phrase-level
NP (%)	18.69	20.11 (**+1.42**)
VP (%)	7.56	7.48 (-0.08)
PP (%)	16.43	17.53 (+1.10)
ADJP (%)	1.76	1.98 (+0.22)

Table 6: Experiments on the LDC Zh→En task to compare the phrase-level and word-level AEG methods by the ratio of noun/verb/prepositional/adjective phrases (NP/VP/PP/ADJP), in the hypothesis.

Random Word Replacement/Switch Noise We first simulate the random replacement and switch noise, where a specific proportion (γ) of *positions* of the source sentence are selected uniformly and replace with random words in the source vocabulary (also uniformly). Such phenomenon is common in real-world scenarios, like onomatopoeia in speech recognition. We set $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ to indicate the level of noise and test the sensitivity of NMT models in Figure 1(a) and Figure 1(b).

The analytic results show that our method PAEG improves the robustness of NMT models more than AdvGen, both to random replacement noise and switch noise. When the ratio of noise increases, the BLEU improvement gets consistently larger, which proves the effectiveness of PAEG. When the ratio of noise is high (0.7∼0.9), both methods degenerate into random translation machines. It can be attributed that excessive random noise impairs the source-side encoding.

Word-level Synonym Noise Another common noise in the translation system is synonym substitution, where the translation system is required to translate sentences consistently with subtle synonym difference. We first simulate this scenario with moderate word replacements. The selection of noisy positions is the same as random word replacement. Each to-be-replaced word matches the top 5 similar words by word similarity as the candidate set.[6] We add the most/least similar synonym noise by selecting the most/least similar word in the candidate set as the replacement. The noise ratio $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and sensitivity results are in Figure 2(a) and Figure 2(b).

The noise-BLEU curves have almost the same trend as random word replacement, which again proves the superior robustness of PAEG over AdvGen at word-level synonym noises. This is understandable because PAEG is inclusion of word-level alternatives.

Phrase-level Synonym Noise In addition, we would like to verify how robust our method is to phrase-level synonym noises, where phrase structures are destroyed by word-level synonyms replacement, such as the case in Table 1. For this purpose, we select ϵ ratio of *phrases* uniformly and replace them with similar words in the source language vocabulary. The noise ratio $\epsilon \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Figure 3(a) and Figure 3(b) show that with the increase of phrase-level noise, PAEG gets more BLEU improvement both in most and least similar synonym noise settings. Our method is more resistant to the destruction of phrase structures, which is proved again in the following section.

3.4 Analysis of Phrase-level Substitution

Phrase-level substitution shows remarkable improvement of the BLEU scores on average. In this subsection, we analyze the translation details and discuss the reason for such an improvement.

Phrase Translation First, we make the statistics of the ratio of phrases η of the (generated) hypothesis in LDC Chinese-English translation in Table 6. In a text $\mathbf{x}$, the ratio of phrases η is defined as

[6] We use word embedding cosine similarity by pre-trained word embeddings GloVe (100 dimension) from flairNLP.

Original	SRC: 对正在实施的(家庭/family)(暴力/violence)，(受害人/victim)可以(请求/ask)公安(机关/organ)救助。 TGT: With regard to ongoing family violence, the victim may ask the public security organ for help.
AdvGen	SRC: 对正在实施的(家庭/family)(迫害/abuse)，(受害人/victim)可以(要求/require)公安(机关/organ)救助。 TGT: With regard to ongoing family **abuse**, the **victim** may **require** the public security organ for help.
PAEG	SRC: 对正在实施的(家庭/family)(虐待/abuse)，(受害人/victim)可以(请求/ask)公安(警察/police)救助。 TGT: With regard to ongoing **domestic abuse**, the **sufferer** may ask the public security **police** for help.

Table 7: Comparison of our PAEG method and AdvGen method on the LDC Zh→En dataset. Tokens with underline are substituted by the model as a word. Tokens with wave lines are substituted by the model as a phrase entirely. Chinese tokens and their English counterparts are in brackets (Chinese/English).

N-gram	Word-level	Phrase-level
1-gram BLEU	79.56	79.78 (+0.22)
2-gram BLEU	52.87	53.59 (+0.72)
3-gram BLEU	34.49	35.65 (**+1.16**)
4-gram BLEU	24.22	25.03 (+0.81)

Table 8: Experiments on the LDC Zh→En to compare the phrase-level and word-level AEG method in n-gram BLEU scores. Phrase-level method improves n-gram ($n > 1$) BLEU scores more.

the sum of the phrase lengths in $\mathbf{x}$ divided by the text length $|\mathbf{x}|$. For the word-level AEG method, the η of noun phrases (NP) is 18.69% on average. While for the phrase-level method, the ratio of NP is remarkably $20.11\%(+1.42\%)$. Besides, the η of prepositional phrases (PP) also increases 1.1% by phrase-level substitution.

These results show that the NMT model trained on PAEG considers more about phrases, especially NP and PP. Phrase-level substitution prevents the damage to the structure of phrases, guarantee the normal ratio of phrases in the augmented dataset, and thus teaches the decoder to generate phrases.

N-gram Accuracy Besides, we analyze the improvements for different n-gram BLEU scores in Table 8. PAEG improve the 3-gram BLEU greatly (+1.16 points) over the word-level method. 2-gram and 3-gram BLEU also get moderate improvements (+0.7∼0.8 points), much greater than 1-gram BLEU. These results verify that, using phrase-level strategy, longer grams can be translated more accurately (to match the phrases in the references).

Case Study In Table 7, there is a case of the example generating process from the LDC dataset. On the target side, AdvGen substituted “violence” to “abuse”. PAEG selected the 6-th position of the target sentence and substituted “family violence” to “domestic abuse” entirely. Though “family abuse” does not violate the original meaning, the substitution “domestic abuse” is more reasonable.

4 Related Work

Adversarial training for neural networks has been studied recently (Szegedy et al., 2014; Goodfellow et al., 2015). Similar ideas are applied into natural language processing (Goyal et al., 2016; Li et al., 2017; Yang et al., 2018; Cheng et al., 2018, 2019, 2020; Namysl et al., 2020; Croce et al., 2020; Wang et al., 2020a; Zang et al., 2020; Ding et al., 2020). Specifically, adversarial example generation (Fadaee et al., 2017; Ebrahimi et al., 2018; Wang et al., 2018b; Cheng et al., 2020; Zou et al., 2020; Zheng et al., 2020; Hidey et al., 2020; Zhang et al., 2021; Lai et al., 2022) is proved to be useful to train a robust NMT system. Recently, Cheng et al. (2019) adopted a gradient-based method to craft adversarial examples at word level, using the adversarial source input to attack while the target input to defend the model.

Our bidirectional generation method is similar to multilingual NMT training. Multilingual NMT models (Dong et al., 2015; Luong et al., 2016; Johnson et al., 2017; Wang et al., 2020b; Zhang et al., 2020; Zhu et al., 2020; Siddhant et al., 2020) are trained over multiple language pairs with parameter sharing, such as using the same encoder/decoder for different source/target languages (Johnson et al., 2017), using one encoder and separate decoders to translate one language to multiple languages (Dong et al., 2015), and sharing an attention mechanism (Firat et al., 2016) across multiple language pairs. In this work, we use the adversarial examples generated from the other direction to improve the robustness of the original translation direction.

5 Conclusion

In this work, we propose a phrase-level adversarial example generation method. Our goal is to improve the fluency of the adversarial examples. We improve a gradient-based word-level method with phrase-level candidate construction, overall substitution strategy, and bidirectional generation. We verify our method on Chinese-English, German-English, and English-German corpus, and the results show that PAEG can improve both translation quality and robustness to noisy inputs significantly.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86, Melbourne, Australia. Association for Computational Linguistics.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. AdvAug: Robust adversarial augmentation for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5961–5970, Online. Association for Computational Linguistics.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1766, Melbourne, Australia. Association for Computational Linguistics.

Danilo Croce, Giuseppe Castellucci, and Roberto Basili. 2020. GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2114–2119, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Dingkun Long, Guangwei Xu, Muhua Zhu, Pengjun Xie, Xiaobin Wang, and Haitao Zheng. 2020. Coupling distant annotation and adversarial training for cross-domain Chinese word segmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6662–6671, Online. Association for Computational Linguistics.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Jiangtao Feng, Lingpeng Kong, Po-Sen Huang, Chong Wang, Da Huang, Jiayuan Mao, Kan Qiao, and Dengyong Zhou. 2018. Neural phrase-to-phrase machine translation. *CoRR*, abs/1811.02172.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron C. Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4601–4609.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567.

Christopher Hidey, Tuhin Chakrabarty, Tariq Alhindi, Siddharth Varia, Kriste Krstovski, Mona Diab, and Smaranda Muresan. 2020. DeSePtion: Dual sequence prediction and adversarial examples for improved fact-checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8593–8606, Online. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Siyu Lai, Zhen Yang, Fandong Meng, Xue Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2022. Generating authentic adversarial examples beyond meaning-preserving with doubly round-trip translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4256–4266. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169, Copenhagen, Denmark. Association for Computational Linguistics.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Marcin Namysl, Sven Behnke, and Joachim Köhler. 2020. NAT: Noise-aware training for robust neural sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1501–1517, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2827–2835, Online. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1916–1920, Minneapolis, Minnesota. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Rui Wang, Xuemeng Hu, Deyu Zhou, Yulan He, Yuxuan Xiong, Chenchen Ye, and Haiyang Xu. 2020a. Neural topic modeling with bidirectional adversarial training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 340–350, Online. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018a. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018b. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020b. Balancing training for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8526–8537, Online. Association for Computational Linguistics.

Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Improving neural machine translation with conditional sequence generative adversarial nets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1346–1355, New Orleans, Louisiana. Association for Computational Linguistics.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, Online. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Xinze Zhang, Junzhe Zhang, Zhenhua Chen, and Kun He. 2021. Crafting adversarial examples for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1967–1977. Association for Computational Linguistics.

Xiaoqing Zheng, Jiehang Zeng, Yi Zhou, Cho-Jui Hsieh, Minhao Cheng, and Xuanjing Huang. 2020. Evaluating and enhancing the robustness of neural network-based dependency parsing models with adversarial examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6600–6610, Online. Association for Computational Linguistics.

Changfeng Zhu, Heng Yu, Shanbo Cheng, and Weihua Luo. 2020. Language-aware interlingua for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1650–1655, Online. Association for Computational Linguistics.

Wei Zou, Shujian Huang, Jun Xie, Xinyu Dai, and Jiajun Chen. 2020. A reinforced generation of adversarial examples for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3486–3497, Online. Association for Computational Linguistics.

A Details of PAEG

A.1 Vulnerable Positions

In this work, there is an assumption that substituting words in vulnerable positions (positions with greater gradient norm) is more likely to add perturbation to model training. In our experiments, we have tried sampling the positions of source phrases randomly and found that vulnerable positions is better (+0.2∼0.3 BLEU points).

A.2 Phrase Embedding

In the experiments, "max-pooling" has been explored to get the phrase embedding/gradients from word embedding/gradients, and it has a similar result as the "average" operation, within 0.2 BLEU points. In the implementation, "max-pooling" is slower than the "average" (using PyTorch 1.7), therefore we choose "average" for convenience.

B Dataset

LDC Chinese-English Task This is a dataset of 1.2M training sequence pairs. The LDC numbers are 2002E17, 2002E18, 2004T08, 2005T10, 2005T34, 2006E17, 2006T06, and 2008T18[7]. We choose the NIST 2006 as the validation set, which has 1664 sentences, and the NIST 2002, NIST 2003, NIST 2005, NIST 2008, NIST 2012 as the test sets, which contain 877, 919, 1082, 1357, 2190 sentences respectively.

IWSLT14 German-English Task This dataset comes from translated TED talks. This dataset contains roughly 160K pairs as the training set, 7K pairs as the validation set, and 7K pairs as the test set, respectively. We take the IWSLT14 test set as the test set.

WMT14 English-German Task The training data has 4.5M sentence pairs. We use the newstest2013 as the valid set and the newstest2014 as the test set.

C Training Details

Our backbone model is the Transformer model (Vaswani et al., 2017). The NMT model consists of a Transformer encoder and a Transformer decoder. The pre-trained LM is BERT-based[8]. We use `nltk.parse` to build the syntax tree and extract the phrases of length $2, 3, 4$. Besides, we use `TextBlob` to extract the noun phradses and merge other phrases (from `nltk.parse`) to build the phrase dictionary.

LDC Chinese-English Translation We use our in-house Chinese word-breaker toolkit to segment Chinese data. We use byte pair encoding (BPE) to encode sentences with a shared token vocabulary of 51K sub-word tokens. The size of the phrase vocabulary is 1.2M for Chinese and 0.9M for English. We limit the maximum sentence length up to 256 words. We apply Adam (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$ to train models for 80 epochs and select the best model parameters according to the model performance on the valid set. We use *Transformer_base* setting: embedding size as 512, feed-forward network (FFN) size as 2048, attention heads as 8, learning rate as 0.1, batch size as 6144, and dropout rate as 0.1. We use the warm-up strategy with 4000 warm-up steps. We report case-insensitive tokenized BLEU-4 scores with Moses[9].

IWSLT14 German-English Translation We use BPE to encode sentences with a shared vocabulary of 10K sub-word tokens. The phrase vocabulary of German is of size 0.4M and English of size 0.4M. We limit the maximum sentence length up to 256 words. We apply Adam with $\beta_1 = 0.9$ and $\gamma_2 = 0.98$ to train models for 100 epochs and select the best model parameters according to the model performance on the valid set. We use *Transformer_small* setting: embedding size as 512, FFN size as 1024, attention heads as 4, learning rate as 0.1, batch size as 6144, and dropout rate as 0.3. We use the warm-up strategy with 4000 warm-up steps.

WMT14 English-German Translation We use BPE to encode sentences with a shared vocabulary of 10K sub-word tokens. The phrase vocabulary of German is of size 0.7M and English of size 0.4M. We limit the maximum sentence length up to 256 words. We apply Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.98$ to train models for 50 epochs and select the best model parameters according to the model performance on the valid set. We use *Transformer_big* setting: embedding size as 1024, FFN

[7] https://catalog.ldc.upenn.edu/byproject

[8] https://github.com/huggingface/transformers

[9] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

size as 4096, attention heads as 16, learning rate as 0.1, batch size as 6144, and dropout rate as 0.1. We use the warm-up strategy with 4000 warm-up steps.

D Training Time Analysis

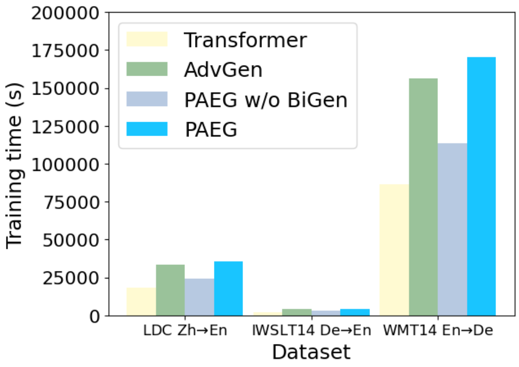


Figure 4: Training time of NMT models. All experiments on the three translation datasets are conducted on 8 NVIDIA 32G V100 GPUs and we set the batch size to fill the GPU memory.

As our method uses augmented data, one concern is whether the training time increases too much. We record the time consumption of our method as well as AdvGen and Transformer. All experiments on the three translation datasets are conducted on 8 NVIDIA 32G V100 GPUs and we set the batch size to fill the GPU memory.

The results are shown in Figure 4. The experiments show that AdvGen uses around the double time of training a Transformer, as it trains two (source and target) language models and generates adversarial data. Our method utilizes a pre-trained language model and thus saves the time of training the language model. Our method without bidirectional generation (BiGen) is faster than AdvGen. Even using bidirectional generation, our method is only slightly slower than AdvGen. Besides, the training time of PAEG is not exactly double of that of PAEG w/o BiGen, which is reasonable as the data from bidirectional generation do not deviate too much from the distribution of the original adversarial samples.