Towards Robust Neural Machine Translation

Yong Cheng[⋆], **Zhaopeng Tu**[⋆], **Fandong Meng**[⋆], **Junjie Zhai**[⋆] **and Yang Liu**[†]
[⋆]Tencent AI Lab, China
[†]State Key Laboratory of Intelligent Technology and Systems
Beijing National Research Center for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, China
Beijing Advanced Innovation Center for Language Resources
chengyong3001@gmail.com
{zptu, fandongmeng, jasonzhai}@tencent.com
liuyang2011@tsinghua.edu.cn

Abstract

Small perturbations in the input can severely distort intermediate representations and thus impact translation quality of neural machine translation (NMT) models. In this paper, we propose to improve the robustness of NMT models with adversarial stability training. The basic idea is to make both the encoder and decoder in NMT models robust against input perturbations by enabling them to behave similarly for the original input and its perturbed counterpart. Experimental results on Chinese-English, English-German and English-French translation tasks show that our approaches can not only achieve significant improvements over strong NMT systems but also improve the robustness of NMT models.

1 Introduction

Neural machine translation (NMT) models have advanced the state of the art by building a single neural network that can better learn representations (Cho et al., 2014; Sutskever et al., 2014). The neural network consists of two components: an encoder network that encodes the input sentence into a sequence of distributed representations, based on which a decoder network generates the translation with an attention model (Bahdanau et al., 2015; Luong et al., 2015). A variety of NMT models derived from this encoder-decoder framework have further improved the performance of machine translation systems (Gehring et al., 2017; Vaswani et al., 2017). NMT is capable of generalizing better to unseen text by exploiting word similarities in embeddings and capturing long-distance reordering by conditioning on larger contexts in a continuous way.

Input	tamen *bupa* kunnan zuochu weiqi AI.
Output	They are not afraid of difficulties to make Go AI.
Input	tamen *buwei* kunnan zuochu weiqi AI.
Output	They are not afraid to make Go AI.

Table 1: The non-robustness problem of neural machine translation. Replacing a Chinese word with its synonym (i.e., "*bupa*" → "*buwei*") leads to significant erroneous changes in the English translation. Both "*bupa*" and "*buwei*" can be translated to the English phrase "*be not afraid of*."

However, studies reveal that very small changes to the input can fool state-of-the-art neural networks with high probability (Goodfellow et al., 2015; Szegedy et al., 2014). Belinkov and Bisk (2018) confirm this finding by pointing out that NMT models are very brittle and easily falter when presented with noisy input. In NMT, due to the introduction of RNN and attention, each contextual word can influence the model prediction in a global context, which is analogous to the "butterfly effect." As shown in Table 1, although we only replace a source word with its synonym, the generated translation has been completely distorted. We investigate severe variations of translations caused by small input perturbations by replacing one word in each sentence of a test set with its synonym. We observe that 69.74% of translations have changed and the BLEU score is only 79.01 between the translations of the original inputs and the translations of the perturbed inputs, suggesting that NMT models are very sensitive to small perturbations in the input. The vulnerability and instability of NMT models limit their applicability to a broader range of tasks, which require robust performance on noisy inputs. For example, simultaneous translation systems use auto-

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 1756–1766
Melbourne, Australia, July 15 - 20, 2018.

matic speech recognition (ASR) to transcribe input speech into a sequence of hypothesized words, which are subsequently fed to a translation system. In this pipeline, ASR errors are presented as sentences with noisy perturbations (the same pronunciation but incorrect words), which is a significant challenge for current NMT models. Moreover, instability makes NMT models sensitive to misspellings and typos in text translation.

In this paper, we address this challenge with *adversarial stability training* for neural machine translation. The basic idea is to improve the robustness of two important components in NMT: the encoder and decoder. To this end, we propose two approaches to constructing noisy inputs with small perturbations to make NMT models resist them. As important intermediate representations encoded by the encoder, they directly determine the accuracy of final translations. We introduce adversarial learning to make behaviors of the encoder consistent for both an input and its perturbed counterpart. To improve the stability of the decoder, our method jointly maximizes the likelihoods of original and perturbed data. Adversarial stability training has the following advantages:

1. *Improving both the robustness and translation performance*: Our adversarial stability training is capable of not only improving the robustness of NMT models but also achieving better translation performance.

2. *Applicable to arbitrary noisy perturbations*: In this paper, we propose two approaches to constructing noisy perturbations for inputs. However, our training framework can be easily extended to arbitrary noisy perturbations. Especially, we can design task-specific perturbation methods.

3. *Transparent to network architectures*: Our adversarial stability training does not depend on specific NMT architectures. It can be applied to arbitrary NMT systems.

Experiments on Chinese-English, English-French and English-German translation tasks show that adversarial stability training achieves significant improvements across different languages pairs. Our NMT system outperforms the state-of-the-art RNN-based NMT system (GNMT) (Wu et al., 2016) and obtains comparable performance with the CNN-based NMT system (Gehring et al., 2017). Related experimental analyses validate that our training approach can improve the robustness of NMT models.

2 Background

NMT is an end-to-end framework which directly optimizes the translation probability of a target sentence $\mathbf{y} = y_1, ..., y_N$ given its corresponding source sentence $\mathbf{x} = x_1, ..., x_M$:

$$P(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) = \prod_{n=1}^{N} P(y_n|\mathbf{y}_{<n}, \mathbf{x}; \boldsymbol{\theta}) \quad (1)$$

where $\boldsymbol{\theta}$ is a set of model parameters and $\mathbf{y}_{<n}$ is a partial translation. $P(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$ is defined on a holistic neural network which mainly includes two core components: an *encoder* encodes a source sentence $\mathbf{x}$ into a sequence of hidden representations $\mathbf{H}_\mathbf{x} = \mathbf{H}_1, ..., \mathbf{H}_M$, and a *decoder* generates the n-th target word based on the sequence of hidden representations:

$$P(y_n|\mathbf{y}_{<n}, \mathbf{x}; \boldsymbol{\theta}) \propto \exp\{g(y_{n-1}, \mathbf{s}_n, \mathbf{H}_\mathbf{x}; \boldsymbol{\theta})\} \quad (2)$$

where $\mathbf{s}_n$ is the n-th hidden state on target side. Thus the model parameters of NMT include the parameter sets of the encoder $\boldsymbol{\theta}_{\text{enc}}$ and the decoder $\boldsymbol{\theta}_{\text{dec}}$: $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dec}}\}$. The standard training objective is to minimize the negative log-likelihood of the training corpus $\mathcal{S} = \{\langle \mathbf{x}^{(s)}, \mathbf{y}^{(s)} \rangle\}_{s=1}^{|\mathcal{S}|}$:

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \operatorname*{argmin}_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) \\ &= \operatorname*{argmin}_{\boldsymbol{\theta}} \Big\{ \sum_{\langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{S}} -\log P(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) \Big\} \quad (3) \end{aligned}$$

Due to the vulnerability and instability of deep neural networks, NMT models usually suffer from a drawback: small perturbations in the input can dramatically deteriorate its translation results. Belinkov and Bisk (2018) point out that character-based NMT models are very brittle and easily falter when presented with noisy input. We find that word-based and subword-based NMT models also confront with this shortcoming, as shown in Table 1. We argue that the distributed representations should fulfill the stability expectation, which is the underlying concept of the proposed approach. Recent work has shown that adversarially trained models can be made robust to such perturbations (Zheng et al., 2016; Madry et al., 2018). Inspired by this, in this work, we improve the robustness of encoder representations against noisy perturbations with adversarial learning (Goodfellow et al., 2014).

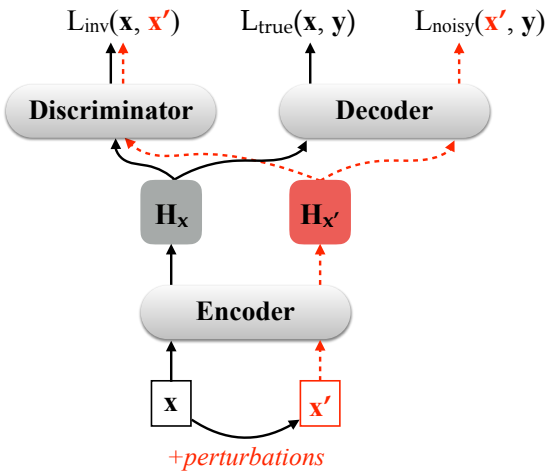

Figure 1: The architecture of NMT with adversarial stability training. The dark solid arrow lines represent the forward-pass information flow for the input sentence $\mathbf{x}$, while the red dashed arrow lines for the noisy input sentence $\mathbf{x}'$, which is transformed from $\mathbf{x}$ by adding small perturbations.

3 Approach

3.1 Overview

The goal of this work is to propose a general approach to make NMT models learned to be more robust to input perturbations. Our basic idea is to maintain the consistency of behaviors through the NMT model for the source sentence $\mathbf{x}$ and its perturbed counterpart $\mathbf{x}'$. As aforementioned, the NMT model contains two procedures for projecting a source sentence $\mathbf{x}$ to its target sentence $\mathbf{y}$: the encoder is responsible for encoding $\mathbf{x}$ as a sequence of representations $\mathbf{H}_\mathbf{x}$, while the decoder outputs $\mathbf{y}$ with $\mathbf{H}_\mathbf{x}$ as input. We aim at learning the perturbation-invariant encoder and decoder.

Figure 1 illustrates the architecture of our approach. Given a source sentence $\mathbf{x}$, we construct a set of perturbed sentences $\mathcal{N}(\mathbf{x})$, in which each sentence $\mathbf{x}'$ is constructed by adding small perturbations to $\mathbf{x}$. We require that $\mathbf{x}'$ is a subtle variation from $\mathbf{x}$ and they have similar semantics. Given the input pair $(\mathbf{x}, \mathbf{x}')$, we have two expectations: (1) the encoded representation $\mathbf{H}_{\mathbf{x}'}$ should be close to $\mathbf{H}_\mathbf{x}$; and (2) given $\mathbf{H}_{\mathbf{x}'}$, the decoder is able to generate the robust output $\mathbf{y}$. To this end, we introduce two additional objectives to improve the robustness of the encoder and decoder:

- $\mathcal{L}_{\text{inv}}(\mathbf{x}, \mathbf{x}')$ to encourage the encoder to output similar intermediate representations $\mathbf{H}_\mathbf{x}$ and $\mathbf{H}_{\mathbf{x}'}$ for $\mathbf{x}$ and $\mathbf{x}'$ to achieve an invariant encoder, which benefits outputting the same translations. We cast this objective in the adversarial learning framework.

- $\mathcal{L}_{\text{noisy}}(\mathbf{x}', \mathbf{y})$ to guide the decoder to generate output $\mathbf{y}$ given the noisy input $\mathbf{x}'$, which is modeled as $-\log P(\mathbf{y}|\mathbf{x}')$. It can also be defined as KL divergence between $P(\mathbf{y}|\mathbf{x})$ and $P(\mathbf{y}|\mathbf{x}')$ that indicates using $P(\mathbf{y}|\mathbf{x})$ to teach $P(\mathbf{y}|\mathbf{x}')$.

As seen, the two introduced objectives aim to improve the robustness of the NMT model which can be free of high variances in target outputs caused by small perturbations in inputs. It is also natural to introduce the original training objective $\mathcal{L}(\mathbf{x}, \mathbf{y})$ on $\mathbf{x}$ and $\mathbf{y}$, which can guarantee good translation performance while keeping the stability of the NMT model.

Formally, given a training corpus $\mathcal{S}$, the adversarial stability training objective is

$$\begin{aligned} & \mathcal{J}(\boldsymbol{\theta}) \\ = & \sum_{\langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{S}} \Big(\mathcal{L}_{\text{true}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dec}}) \\ & + \alpha \sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x})} \mathcal{L}_{\text{inv}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dis}}) \\ & + \beta \sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x})} \mathcal{L}_{\text{noisy}}(\mathbf{x}', \mathbf{y}; \boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dec}}) \Big) \end{aligned} \quad (4)$$

where $\mathcal{L}_{\text{true}}(\mathbf{x}, \mathbf{y})$ and $\mathcal{L}_{\text{noisy}}(\mathbf{x}', \mathbf{y})$ are calculated using Equation 3, and $\mathcal{L}_{\text{inv}}(\mathbf{x}, \mathbf{x}')$ is the adversarial loss to be described in Section 3.3. α and β control the balance between the original translation task and the stability of the NMT model. $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dec}}, \boldsymbol{\theta}_{\text{dis}}\}$ are trainable parameters of the encoder, decoder, and the newly introduced discriminator used in adversarial learning. As seen, the parameters of encoder $\boldsymbol{\theta}_{\text{enc}}$ and decoder $\boldsymbol{\theta}_{\text{dec}}$ are trained to minimize both the translation loss $\mathcal{L}_{\text{true}}(\mathbf{x}, \mathbf{y})$ and the stability losses ($\mathcal{L}_{\text{noisy}}(\mathbf{x}', \mathbf{y})$ and $\mathcal{L}_{\text{inv}}(\mathbf{x}, \mathbf{x}')$).

Since $\mathcal{L}_{\text{noisy}}(\mathbf{x}', \mathbf{y})$ evaluates the translation loss on the perturbed neighbour $\mathbf{x}'$ and its corresponding target sentence $\mathbf{y}$, it means that we augment the training data by adding perturbed neighbours, which can potentially improve the translation performance. In this way, our approach not only makes the output of NMT models more robust, but also improves the performance on the original translation task.

In the following sections, we will first describe how to construct perturbed inputs with different strategies to fulfill different goals (Section 3.2), followed by the proposed adversarial learning mechanism for the perturbation-invariant encoder (Section 3.3). We conclude this section with the training strategy (Section 3.4).

3.2 Constructing Perturbed Inputs

At each training step, we need to generate a perturbed neighbour set $\mathcal{N}(\mathbf{x})$ for each source sentence $\mathbf{x}$ for adversarial stability training. In this paper, we propose two strategies to construct the perturbed inputs at multiple levels of representations.

The first approach generates perturbed neighbours at the *lexical* level. Given an input sentence $\mathbf{x}$, we randomly sample some word positions to be modified. Then we replace words at these positions with other words in the vocabulary according to the following distribution:

$$P(x|\mathbf{x}_i) = \frac{\exp\{\cos(\mathbf{E}[\mathbf{x}_i], \mathbf{E}[x])\}}{\sum_{x \in \mathcal{V}_x \backslash \mathbf{x}_i} \exp\{\cos(\mathbf{E}[\mathbf{x}_i], \mathbf{E}[x])\}} \quad (5)$$

where $\mathbf{E}[\mathbf{x}_i]$ is the word embedding for word $\mathbf{x}_i$, $\mathcal{V}_x \backslash \mathbf{x}_i$ is the source vocabulary set excluding the word $\mathbf{x}_i$, and $\cos(\mathbf{E}[\mathbf{x}_i], \mathbf{E}[x])$ measures the similarity between word $\mathbf{x}_i$ and x. Thus we can change the word to another word with similar semantics.

One potential problem of the above strategy is that it is hard to enumerate all possible positions and possible types to generate perturbed neighbours. Therefore, we propose a more general approach to modifying the sentence at the *feature* level. Given a sentence, we can obtain the word embedding for each word. We add the Gaussian noise to a word embedding to simulate possible types of perturbations. That is

$$\mathbf{E}[\mathbf{x}'_i] = \mathbf{E}[\mathbf{x}_i] + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathbf{N}(0, \sigma^2 \mathbf{I}) \quad (6)$$

where the vector $\boldsymbol{\epsilon}$ is sampled from a Gaussian distribution with variance σ^2. σ is a hyper-parameter. We simply introduce Gaussian noise to all of word embeddings in $\mathbf{x}$.

The proposed scheme is a general framework where one can freely define the strategies to construct perturbed inputs. We just present two possible examples here. The first strategy is potentially useful when the training data contains noisy words, while the latter is a more general strategy to improve the robustness of common NMT models. In practice, one can design specific strategies for particular tasks. For example, we can replace correct words with their homonyms (same pronunciation but different meanings) to improve NMT models for simultaneous translation systems.

3.3 Adversarial Learning for the Perturbation-invariant Encoder

The goal of the perturbation-invariant encoder is to make the representations produced by the encoder indistinguishable when fed with a correct sentence $\mathbf{x}$ and its perturbed counterpart $\mathbf{x}'$, which is directly beneficial to the output robustness of the decoder. We cast the problem in the adversarial learning framework (Goodfellow et al., 2014). The encoder serves as the generator G, which defines the policy that generates a sequence of hidden representations $\mathbf{H}_\mathbf{x}$ given an input sentence $\mathbf{x}$. We introduce an additional discriminator D to distinguish the representation of perturbed input $\mathbf{H}_{\mathbf{x}'}$ from that of the original input $\mathbf{H}_\mathbf{x}$. The goal of the generator G (i.e., encoder) is to produce similar representations for $\mathbf{x}$ and $\mathbf{x}'$ which could fool the discriminator, while the discriminator D tries to correctly distinguish the two representations.

Formally, the adversarial learning objective is

$$\begin{aligned} & \mathcal{L}_{\text{inv}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dis}}) \\ = \; & \mathbb{E}_{\mathbf{x} \sim \mathcal{S}}[-\log D(G(\mathbf{x}))] + \\ & \mathbb{E}_{\mathbf{x}' \sim \mathcal{N}(\mathbf{x})}\left[-\log(1 - D(G(\mathbf{x}')))\right] \end{aligned} \quad (7)$$

The discriminator outputs a classification score given an input representation, and tries to maximize $D(G(\mathbf{x}))$ to 1 and minimize $D(G(\mathbf{x}'))$ to 0. The objective encourages the encoder to output similar representations for $\mathbf{x}$ and $\mathbf{x}'$, so that the discriminator fails to distinguish them.

The training procedure can be regarded as a min-max two-player game. The encoder parameters $\boldsymbol{\theta}_{\text{enc}}$ are trained to maximize the loss function to fool the discriminator. The discriminator parameters $\boldsymbol{\theta}_{\text{dis}}$ are optimized to minimize this loss for improving the discriminating ability. For efficiency, we update both the encoder and the discriminator simultaneously at each iteration, rather than the periodical training strategy that is commonly used in adversarial learning. Lamb et al. (2016) also propose a similar idea to use Professor Forcing to make the behaviors of RNNs be indistinguishable when training and sampling the networks.

3.4 Training

As shown in Figure 1, our training objective includes three sets of model parameters for three modules. We use mini-batch stochastic gradient descent to optimize our model. In the forward pass, besides a mini-batch of $\mathbf{x}$ and $\mathbf{y}$, we also construct a mini-batch consisting of the perturbed neighbour $\mathbf{x}'$ and $\mathbf{y}$. We propagate the information to calculate these three loss functions according to arrows. Then, gradients are collected to update three sets of model parameters. Except for the gradients of $\mathcal{L}_{\text{inv}}$ with respect to $\boldsymbol{\theta}_{\text{enc}}$ are multiplying by -1, other gradients are normally back-propagated. Note that we update $\boldsymbol{\theta}_{\text{inv}}$ and $\boldsymbol{\theta}_{\text{enc}}$ simultaneously for training efficiency.

4 Experiments

4.1 Setup

We evaluated our adversarial stability training on translation tasks of several language pairs, and reported the 4-gram BLEU (Papineni et al., 2002) score as calculated by the *multi-bleu.perl* script.

Chinese-English We used the LDC corpus consisting of 1.25M sentence pairs with 27.9M Chinese words and 34.5M English words respectively. We selected the best model using the NIST 2006 set as the validation set (hyper-parameter optimization and model selection). The NIST 2002, 2003, 2004, 2005, and 2008 datasets are used as test sets.

English-German We used the WMT 14 corpus containing 4.5M sentence pairs with 118M English words and 111M German words. The validation set is newstest2013, and the test set is newstest2014.

English-French We used the IWSLT corpus which contains 0.22M sentence pairs with 4.03M English words and 4.12M French words. The IWLST corpus is very dissimilar from the NIST and WMT corpora. As they are collected from TED talks and inclined to spoken language, we want to verify our approaches on the non-normative text. The IWSLT 14 test set is taken as the validation set and 15 test set is used as the test set.

For English-German and English-French, we tokenize both English, German and French words using `tokenize.perl` script. We follow Sennrich et al. (2016b) to split words into subword units. The numbers of merge operations in byte pair encoding (BPE) are set to 30K, 40K and 30K respectively for Chinese-English, English-German, and English-French. We report the case-sensitive tokenized BLEU score for English-German and English-French and the case-insensitive tokenized BLEU score for Chinese-English.

Our baseline system is an in-house NMT system. Following Bahdanau et al. (2015), we implement an RNN-based NMT in which both the encoder and decoder are two-layer RNNs with residual connections between layers (He et al., 2016b). The gating mechanism of RNNs is gated recurrent unit (GRUs) (Cho et al., 2014). We apply layer normalization (Ba et al., 2016) and dropout (Hinton et al., 2012) to the hidden states of GRUs. Dropout is also added to the source and target word embeddings. We share the same matrix between the target word embeedings and the pre-softmax linear transformation (Vaswani et al., 2017). We update the set of model parameters using Adam SGD (Kingma and Ba, 2015). Its learning rate is initially set to 0.05 and varies according to the formula in Vaswani et al. (2017).

Our adversarial stability training initializes the model based on the parameters trained by maximum likelihood estimation (MLE). We denote adversarial stability training based on lexical-level perturbations and feature-level perturbations respectively as $\text{AST}_{\text{lexical}}$ and $\text{AST}_{\text{feature}}$. We only sample one perturbed neighbour $\mathbf{x}' \in \mathcal{N}(\mathbf{x})$ for training efficiency. For the discriminator used in $\mathcal{L}_{\text{inv}}$, we adopt the CNN discriminator proposed by Kim (2014) to address the variable-length problem of the sequence generated by the encoder. In the CNN discriminator, the filter windows are set to 3, 4, 5 and rectified linear units are applied after convolution operations. We tune the hyper-parameters on the validation set through a grid search. We find that both the optimal values of α and β are set to 1.0. The standard variance in Gaussian noise used in the formula (6) is set to 0.01. The number of words that are replaced in the sentence $\mathbf{x}$ during lexical-level perturbations is taken as $\max(0.2|\mathbf{x}|, 1)$ in which $|\mathbf{x}|$ is the length of $\mathbf{x}$. The default beam size for decoding is 10.

4.2 Translation Results

4.2.1 NIST Chinese-English Translation

Table 2 shows the results on Chinese-English translation. Our strong baseline system significantly outperforms previously reported results on

System	Training	MT06	MT02	MT03	MT04	MT05	MT08
Shen et al. (2016)	MRT	37.34	40.36	40.93	41.37	38.81	29.23
Wang et al. (2017)	MLE	37.29	–	39.35	41.15	38.07	–
Zhang et al. (2018)	MLE	38.38	–	40.02	42.32	38.84	–
this work	MLE	41.38	43.52	41.50	43.64	41.58	31.60
	$\text{AST}_{\text{lexical}}$	43.57	44.82	42.95	45.05	43.45	34.85
	$\text{AST}_{\text{feature}}$	**44.44**	**46.10**	**44.07**	**45.61**	**44.06**	**34.94**

Table 2: Case-insensitive BLEU scores on Chinese-English translation.

System	Architecture	Training	BLEU
Shen et al. (2016)	Gated RNN with 1 layer	MRT	20.45
Luong et al. (2015)	LSTM with 4 layers	MLE	20.90
Kalchbrenner et al. (2017)	ByteNet with 30 layers	MLE	23.75
Wang et al. (2017)	DeepLAU with 4 layers	MLE	23.80
Wu et al. (2016)	LSTM with 8 layers	RL	24.60
Gehring et al. (2017)	CNN with 15 layers	MLE	25.16
Vaswani et al. (2017)	Self-attention with 6 layers	MLE	28.40
this work	Gated RNN with 2 layers	MLE	24.06
		$\text{AST}_{\text{lexical}}$	25.17
		$\text{AST}_{\text{feature}}$	**25.26**

Table 3: Case-sensitive BLEU scores on WMT 14 English-German translation.

Training	tst2014	tst2015
MLE	36.92	36.90
$\text{AST}_{\text{lexical}}$	37.35	37.03
$\text{AST}_{\text{feature}}$	**38.03**	**37.64**

Table 4: Case-sensitive BLEU scores on IWSLT English-French translation.

Chinese-English NIST datasets trained on RNN-based NMT. Shen et al. (2016) propose minimum risk training (MRT) for NMT, which directly optimizes model parameters with respect to BLEU scores. Wang et al. (2017) address the issue of severe gradient diffusion with linear associative units (LAU). Their system is deep with an encoder of 4 layers and a decoder of 4 layers. Zhang et al. (2018) propose to exploit both left-to-right and right-to-left decoding strategies for NMT to capture bidirectional dependencies. Compared with them, our NMT system trained by MLE outperforms their best models by around 3 BLEU points. We hope that the strong baseline systems used in this work make the evaluation convincing.

We find that introducing adversarial stability training into NMT can bring substantial improvements over previous work (up to +3.16 BLEU points over Shen et al. (2016), up to +3.51 BLEU points over Wang et al. (2017) and up to +2.74 BLEU points over Zhang et al. (2018)) and our system trained with MLE across all the datasets. Compared with our baseline system, $\text{AST}_{\text{lexical}}$ achieves +1.75 BLEU improvement on average. $\text{AST}_{\text{feature}}$ performs better, which can obtain +2.59 BLEU points on average and up to +3.34 BLEU points on NIST08.

4.2.2 WMT 14 English-German Translation

In Table 3, we list existing NMT systems as comparisons. All these systems use the same WMT 14 English-German corpus. Except that Shen et al. (2016) and Wu et al. (2016) respectively adopt MRT and reinforcement learning (RL), other systems all use MLE as training criterion. All the systems except for Shen et al. (2016) are deep NMT models with no less than four layers. Google's neural machine translation (GNMT) (Wu et al., 2016) represents a strong RNN-based NMT system. Compared with other RNN-based NMT systems except for GNMT, our baseline system with two layers can achieve better performance than theirs.

When training our NMT system with $\text{AST}_{\text{leixcal}}$, significant improvement (+1.11

Synthetic Type	Training	0 Op.	1 Op.	2 Op.	3 Op.	4 Op.	5 Op.
Swap	MLE	41.38	38.86	37.23	35.97	34.61	32.96
	$\text{AST}_{\text{lexical}}$	43.57	41.18	39.88	37.95	37.02	36.16
	$\text{AST}_{\text{feature}}$	44.44	42.08	40.20	38.67	36.89	35.81
Replacement	MLE	41.38	37.21	31.40	27.43	23.94	21.03
	$\text{AST}_{\text{lexical}}$	43.57	40.53	37.59	35.19	32.56	30.42
	$\text{AST}_{\text{feature}}$	44.44	40.04	35.00	30.54	27.42	24.57
Deletion	MLE	41.38	38.45	36.15	33.28	31.17	28.65
	$\text{AST}_{\text{lexical}}$	43.57	41.89	38.56	36.14	34.09	31.77
	$\text{AST}_{\text{feature}}$	44.44	41.75	39.06	36.16	33.49	30.90

Table 5: Translation results of synthetic perturbations on the validation set in Chinese-English translation. "1 Op." denotes that we conduct one operation (swap, replacement or deletion) on the original sentence.

Source	zhongguo dianzi yinhang yewu guanli xingui jiangyu sanyue yiri qi shixing
Reference	china's new management rules for e-banking operations to take effect on march 1
MLE	china's electronic bank rules to be implemented on march 1
$\text{AST}_{\text{lexical}}$	new rules for business administration of china 's electronic banking industry will come into effect on march 1 .
$\text{AST}_{\text{feature}}$	new rules for business management of china 's electronic banking industry to come into effect on march 1
Perturbed Source	*zhongfang* dianzi yinhang yewu guanli xingui jiangyu sanyue yiri qi shixing
MLE	china to implement new regulations on business management
$\text{AST}_{\text{lexical}}$	the new regulations for the business administrations of the chinese electronics bank will come into effect on march 1 .
$\text{AST}_{\text{feature}}$	new rules for business management of china's electronic banking industry to come into effect on march 1

Table 6: Example translations of a source sentence and its perturbed counterpart by replacing a Chinese word "zhongguo" with its synonym "zhongfang."

BLEU points) can be observed. $\text{AST}_{\text{feature}}$ can obtain slightly better performance. Our NMT system outperforms the state-of-the-art RNN-based NMT system, GNMT, with $+0.66$ BLEU point and performs comparably with Gehring et al. (2017) which is based on CNN with 15 layers. Given that our approach can be applied to any NMT systems, we expect that the adversarial stability training mechanism can further improve performance upon the advanced NMT architectures. We leave this for future work.

4.2.3 IWSLT English-French Translation

Table 4 shows the results on IWSLT English-French Translation. Compared with our strong baseline system trained by MLE, we observe that our models consistently improve translation performance in all datasets. $\text{AST}_{\text{feature}}$ can achieve significant improvements on the tst2015 although $\text{AST}_{\text{lexical}}$ obtains comparable results. These demonstrate that our approach maintains good performance on the non-normative text.

4.3 Results on Synthetic Perturbed Data

In order to investigate the ability of our training approaches to deal with perturbations, we experiment with three types of synthetic perturbations:

- **Swap**: We randomly choose N positions from a sentence and then swap the chosen words with their right neighbours.
- **Replacement**: We randomly replace sampled words in the sentence with other words.
- **Deletion**: We randomly delete N words from each sentence in the dataset.

As shown in Table 5, we can find that our training approaches, $\text{AST}_{\text{lexical}}$ and $\text{AST}_{\text{feature}}$, consistently outperform MLE against perturbations on all the numbers of operations. This means that our

$\mathcal{L}_{\text{true}}$	$\mathcal{L}_{\text{noisy}}$	$\mathcal{L}_{\text{adv}}$	BLEU
$\surd$	$\times$	$\times$	41.38
$\surd$	$\times$	$\surd$	41.91
$\times$	$\surd$	$\times$	42.20
$\surd$	$\surd$	$\times$	42.93
$\surd$	$\surd$	$\surd$	43.57

Table 7: Ablation study of adversarial stability training $\text{AST}_{\text{lexical}}$ on Chinese-English translation. "$\surd$" means the loss function is included in the training objective while "$\times$" means it is not.

approaches have the capability of resisting perturbations. Along with the number of operations increasing, the performance on MLE drops quickly. Although the performance of our approaches also drops, we can see that our approaches consistently surpass MLE. In $\text{AST}_{\text{lexical}}$, with 0 operation, the difference is +2.19 (43.57 Vs. 41.38) for all synthetic types, but the differences are enlarged to +3.20, +9.39, and +3.12 respectively for the three types with 5 operations.

In the *Swap* and *Deletion* types, $\text{AST}_{\text{lexical}}$ and $\text{AST}_{\text{feature}}$ perform comparably after more than four operations. Interestingly, $\text{AST}_{\text{lexical}}$ performs significantly better than both of MLE and $\text{AST}_{\text{feature}}$ after more than one operation in the *Replacement* type. This is because $\text{AST}_{\text{lexical}}$ trains the model specifically on perturbation data that is constructed by replacing words, which agrees with the *Replacement* Type. Overall, $\text{AST}_{\text{lexical}}$ performs better than $\text{AST}_{\text{feature}}$ against perturbations after multiple operations. We speculate that the perturbation method for $\text{AST}_{\text{lexical}}$ and synthetic type are both discrete and they keep more consistent. Table 6 shows example translations of a Chinese sentence and its perturbed counterpart.

These findings indicate that we can construct specific perturbations for a particular task. For example, in simultaneous translation, an automatic speech recognition system usually generates wrong words with the same pronunciation of correct words, which dramatically affects the quality of machine translation system. Therefore, we can design specific perturbations aiming for this task.

4.4 Analysis

4.4.1 Ablation Study

Our training objective function Eq. (4) contains three loss functions. We perform an ablation study on the Chinese-English translation to understand the importance of these loss functions by choosing $\text{AST}_{\text{lexical}}$ as an example. As Table 7 shows, if we remove $\mathcal{L}_{\text{adv}}$, the translation performance decreases by 0.64 BLEU point. However, when $\mathcal{L}_{\text{noisy}}$ is excluded from the training objective function, it results in a significant drop of 1.66 BLEU point. Surprisingly, only using $\mathcal{L}_{\text{noisy}}$ is able to lead to an increase of 0.88 BLEU point.

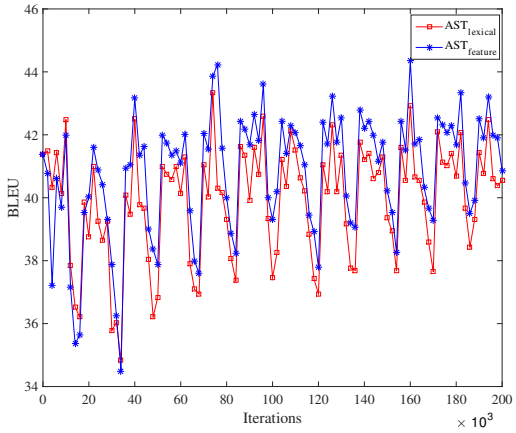

Figure 2: BLEU scores of $\text{AST}_{\text{lexical}}$ over iterations on Chinese-English validation set.

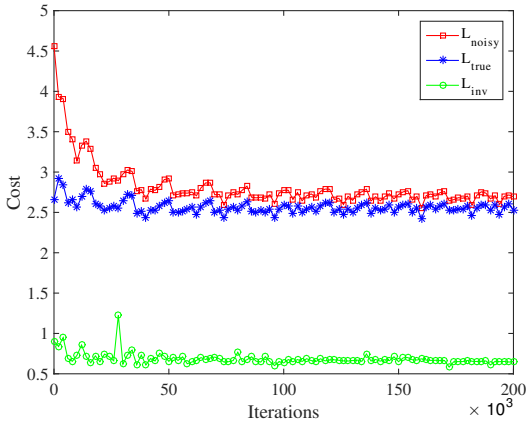

Figure 3: Learning curves of three loss functions, $\mathcal{L}_{\text{true}}$, $\mathcal{L}_{\text{inv}}$ and $\mathcal{L}_{\text{noisy}}$ over iterations on Chinese-English validation set.

4.4.2 BLEU Scores over Iterations

Figure 2 shows the changes of BLEU scores over iterations respectively for $\text{AST}_{\text{lexical}}$ and $\text{AST}_{\text{feature}}$. They behave nearly consistently. Initialized by the model trained by MLE, their performance drops rapidly. Then it starts to go up quickly. Compared with the starting point, the

maximal dropping points reach up to about 7.0 BLEU points. Basically, the curves present the state of oscillation. We think that introducing random perturbations and adversarial learning can make the training not very stable like MLE.

4.4.3 Learning Curves of Loss Functions

Figure 3 shows the learning curves of three loss functions, $\mathcal{L}_{\text{true}}$, $\mathcal{L}_{\text{inv}}$ and $\mathcal{L}_{\text{noisy}}$. We can find that their costs of loss functions decrease not steadily. Similar to the Figure 2, there still exist oscillations in the learning curves although they do not change much sharply. We find that $\mathcal{L}_{\text{inv}}$ converges to around 0.68 after about $100K$ iterations, which indicates that discriminator outputs probability 0.5 for both positive and negative samples and it cannot distinguish them. Thus the behaviors of the encoder for $\mathbf{x}$ and its perturbed neighbour $\mathbf{x}'$ perform nearly consistently.

5 Related Work

Our work is inspired by two lines of research: (1) adversarial learning and (2) data augmentation.

Adversarial Learning Generative Adversarial Network (GAN) (Goodfellow et al., 2014) and its related derivative have been widely applied in computer vision (Radford et al., 2015; Salimans et al., 2016) and natural language processing (Li et al., 2017; Yang et al., 2018). Previous work has constructed adversarial examples to attack trained networks and make networks resist them, which has proved to improve the robustness of networks (Goodfellow et al., 2015; Miyato et al., 2016; Zheng et al., 2016). Belinkov and Bisk (2018) introduce adversarial examples to training data for character-based NMT models. In contrast to theirs, adversarial stability training aims to stabilize both the encoder and decoder in NMT models. We adopt adversarial learning to learn the perturbation-invariant encoder.

Data Augmentation Data augmentation has the capability to improve the robustness of NMT models. In NMT, there is a number of work that augments the training data with monolingual corpora (Sennrich et al., 2016a; Cheng et al., 2016; He et al., 2016a; Zhang and Zong, 2016). They all leverage complex models such as inverse NMT models to generate translation equivalents for monolingual corpora. Then they augment the parallel corpora with these pseudo corpora to improve NMT models. Some authors have recently endeavored to achieve zero-shot NMT through transferring knowledge from bilingual corpora of other language pairs (Chen et al., 2017; Zheng et al., 2017; Cheng et al., 2017) or monolingual corpora (Lample et al., 2018; Artetxe et al., 2018). Our work significantly differs from these work. We do not resort to any complicated models to generate perturbed data and do not depend on extra monolingual or bilingual corpora. The way we exploit is more convenient and easy to implement. We focus more on improving the robustness of NMT models.

6 Conclusion

We have proposed adversarial stability training to improve the robustness of NMT models. The basic idea is to train both the encoder and decoder robust to input perturbations by enabling them to behave similarly for the original input and its perturbed counterpart. We propose two approaches to construct perturbed data to adversarially train the encoder and stabilize the decoder. Experiments on Chinese-English, English-German and English-French translation tasks show that the proposed approach can improve both the robustness and translation performance.

As our training framework is not limited to specific perturbation types, it is interesting to evaluate our approach in natural noise existing in practical applications, such as homonym in the simultaneous translation system. It is also necessary to further validate our approach on more advanced NMT architectures, such as CNN-based NMT (Gehring et al., 2017) and Transformer (Vaswani et al., 2017).

Acknowledgments

We thank the anonymous reviewers for their insightful comments and suggestions. We also thank Xiaoling Li for analyzing experimental results and providing valuable examples. Yang Liu is supported by the National Key R&D Program of China (No. 2017YFB0202204), National Natural Science Foundation of China (No. 61761166008, No. 61522204), Beijing Advanced Innovation Center for Language Resources, and the NExT++ project supported by the National Research Foundation, Prime Ministers Office, Singapore under its IRC@Singapore Funding Initiative.

References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *Proceedings of ICLR*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proceedings of ICLR*.

Yun Chen, Yang Liu, Yong Cheng, and Victor OK Li. 2017. A teacher-student framework for zero-resource neural machine translation. In *Proceedings of ACL*.

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

Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, and Wei Xu. 2017. Joint training for pivot-based neural machine translation. In *Proceedings of IJCAI*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of EMNLP*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of ICML*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of NIPS*.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of ICLR*.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016a. Dual learning for machine translation. In *Proceedings of NIPS*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016b. Deep residual learning for image recognition. In *Proceedings of CVPR*.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2017. Neural machine translation in linear time. In *Proceedings of ICML*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *Proceedings of NIPS*.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *Proceedings of ICLR*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of EMNLP*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP*.

Aleksander Madry, Makelov Aleksandar, Schmidt Ludwig, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *Proceedings of ICLR*.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. 2016. Distributional smoothing with virtual adversarial training. In *Proceedings of ICLR*.

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

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Proceedings of NIPS*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving nerual machine translation models with monolingual data. In *Proceedings of ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of ACL*.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of ACL*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceddings of NIPS*.

Christian Szegedy, Wojciech Zaremba, Sutskever Ilya, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of ICML*.

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 *Proceedings of NIPS*.

Mingxuan Wang, Zhengdong Lu, Jie Zhou, and Qun Liu. 2017. Deep neural machine translation with linear associative unit. In *Proceedings of ACL*.

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*.

Z. Yang, W. Chen, F. Wang, and B. Xu. 2018. Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets. In *Proceedings of NAACL*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of EMNLP*.

Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang. 2018. Asynchronous Bidirectional Decoding for Neural Machine Translation. In *Proeedings of AAAI*.

Hao Zheng, Yong Cheng, and Yang Liu. 2017. Maximum expected likelihood estimation for zero-resource neural machine translation. In *Proceedings of IJCAI*.

Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. 2016. Improving the robustness of deep neural networks via stability training. In *Proceedings of CVPR*.