Towards Robust Neural Machine Translation

Yong Cheng[#], Zhaopeng Tu[#], Fandong Meng[#], Junjie Zhai[#], Yang Liu[†]
[#]Tencent AI Lab, China
[†] Department of Computer Science and Technology, Tsinghua University, Beijing, China

Introduction

Input	他们 **不怕** 困难 做出 围棋 AI。
Output	They are not afraid of difficulties to make Go AI.
Input	他们 不**畏** 困难 做出 围棋 AI。
Output	They are not afraid to make Go AI.
Input	毕竟 是 学生， 他 **不得不** 听 学校的 安排。
Output	After all, he was a student, and he had to listen to the arrangements of the school.
Input	毕竟 是 学生， 他 不**的**不 听 学校的 安排。
Output	After all, is a student, he did not listen to the arrangements of the school.

Table 1: The non-robustness problem of neural machine translation. Replacing a Chinese word with its synonym (i.e., "不怕" → "不畏") in example 1 or its homonym (i.e., "不得不"→ "不的不") in example 2 leads to significant erroneous changes in the English translation.

Small perturbations in the input can severely distort intermediate representations and thus impact translation quality of neural machine translation (NMT) models. Due to the introduction of RNN and attention, each contextual word can influence the model prediction in a global context. As shown in Table 1, although we only replace a source word with its synonym or its homonym, the generated translation has been completely distorted. In this paper, we propose to improve the robustness of NMT models with adversarial stability training (AST). 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.

Constructing Perturbed Inputs

Our training framework can be easily extended to arbitrary noisy perturbations. Especially, we can design task-specific perturbation methods. In this paper, we propose two possible strategies to construct the perturbed inputs at different levels of representations.

- At the *lexical* level: we replace words at sampled positions with other words in the vocabulary according to the following distribution:

$$P(x|\mathbf{x}_i) = \frac{\exp\{\cos(\mathbf{E}[\mathbf{x}_i], \mathbf{E}[x])\}}{\sum_{x \in \mathcal{V}_x \setminus \mathbf{x}_i} \exp\{\cos(\mathbf{E}[\mathbf{x}_i], \mathbf{E}[x])\}} \quad (1)$$

- At the *feature* level: we add the Gaussian noise to word embeddings to simulate possible types of perturbations. That is:

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

Translation Experiments

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
Zhang et al. (2018)	MLE	38.38	–	40.02	42.32	38.84	–
	MLE	41.38	43.52	41.50	43.64	41.58	31.60
this work	$\text{AST}_{\text{lexical}}$	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 3: Case-insensitive BLEU scores on Chinese-English translation.

System	Architecture	Training	BLEU
Shen et al. (2015)	Gated RNN with 1 layer	MRT	20.45
Wu et al. (2016)	LSTM with 8 layers	RL	24.60
Gehring et al. (2017)	CNN with 15 layers	MLE	25.16
		MLE	24.06
this work	Gated RNN with 2 layers	$\text{AST}_{\text{lexical}}$	25.17
		$\text{AST}_{\text{feature}}$	**25.26**

Table 4: Case-sensitive BLEU scores on WMT 14 English-German translation.

Synthetic Type	Training	0 Op.	1 Op.	2 Op.	3 Op.	4 Op.	5 Op.
	MLE	41.38	38.86	37.23	35.97	34.61	32.96
Swap	$\text{AST}_{\text{lexical}}$	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
	MLE	41.38	37.21	31.40	27.43	23.94	21.03
Replacement	$\text{AST}_{\text{lexical}}$	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

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

Adversarial Stability Training

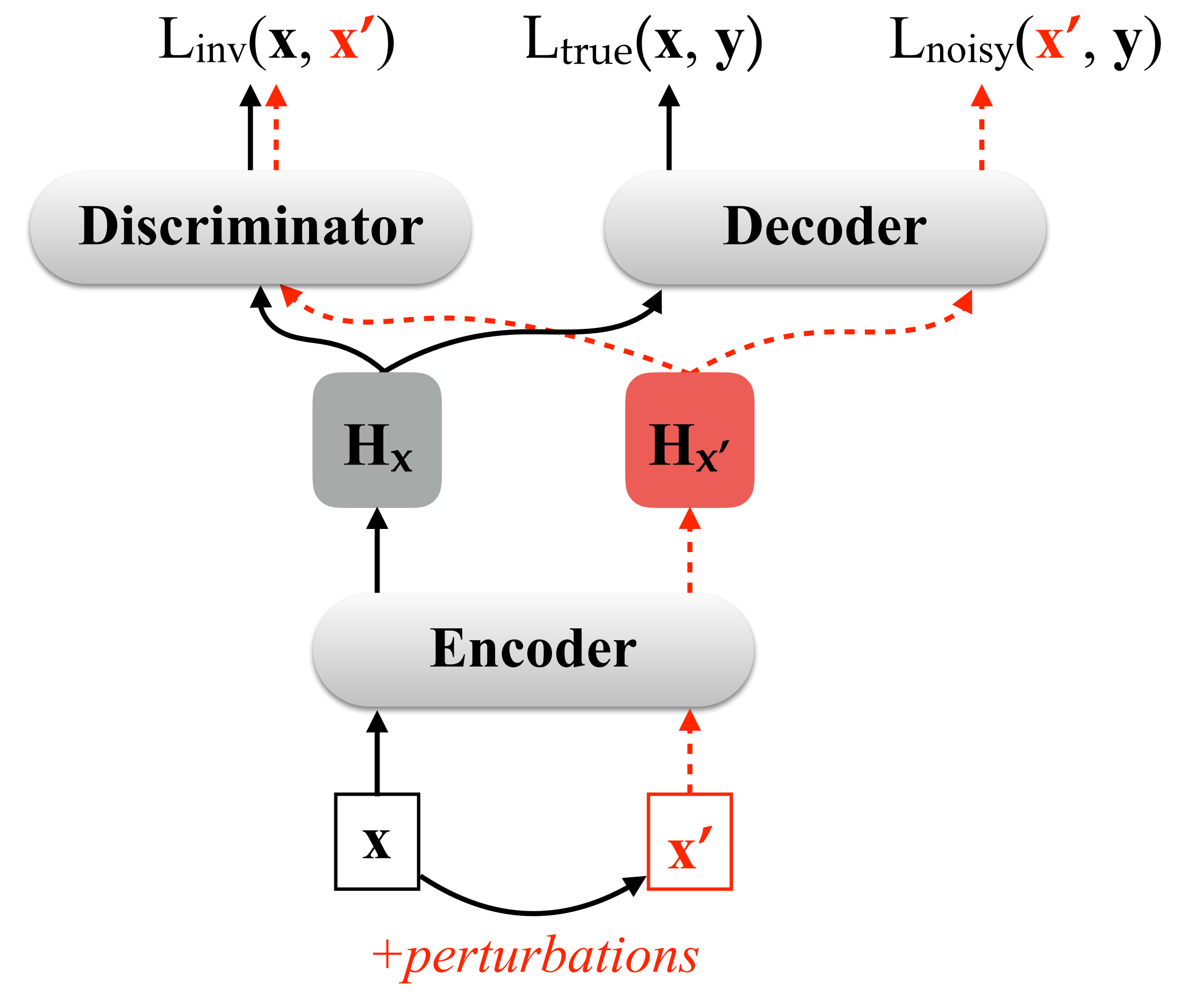
$\mathrm{L}_{\mathrm{inv}}(\mathbf{x}, \mathbf{x}')$
$\mathrm{L}_{\mathrm{true}}(\mathbf{x}, \mathbf{y})$
$\mathrm{L}_{\mathrm{noisy}}(\mathbf{x}', \mathbf{y})$
Discriminator
Decoder
$\mathbf{H}_{\mathbf{x}}$
$\mathbf{H}_{\mathbf{x}'}$
Encoder
$\mathbf{x}$
$\mathbf{x}'$
+perturbations

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 $\mathbf{x}' \in \mathcal{N}(\mathbf{x})$, which is transformed from $\mathbf{x}$ by adding small perturbations.

Besides the standard loss $\mathcal{L}_{\text{true}}$ on the original input $\langle \mathbf{x}, \mathbf{y} \rangle$, we introduce two 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. Formally, the adversarial learning objective is:

$$\begin{aligned} & \mathcal{L}_{\text{inv}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dis}}) \\ = \; & \mathbb{E}_{\mathbf{x} \sim \mathcal{S}}[-\log D(G(\mathbf{x}))] + \\ & \mathbb{E}_{\mathbf{x}' \sim \mathcal{N}(\mathbf{x})}[-\log(1 - D(G(\mathbf{x}')))] \end{aligned} \quad (3)$$

- $\mathcal{L}_{\text{noisy}}(\mathbf{x}', \mathbf{y})$ 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 $P(\mathbf{y}|\mathbf{x}')$.

Given a training corpus $\mathcal{S}$, the adversarial stability training objective is:

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

Ablation Study

$\mathcal{L}_{\text{true}}$	$\mathcal{L}_{\text{noisy}}$	$\mathcal{L}_{\text{inv}}$	BLEU
✓	×	×	41.38
✓	×	✓	41.91
×	✓	×	42.20
✓	✓	×	42.93
✓	✓	✓	43.57

Table 2: Ablation study of adversarial stability training $\text{AST}_{\text{lexical}}$ on Chinese-English translation. "✓" means the loss function is included in the training objective while "×" means it is not.