Proceedings of Machine Learning Research 95:534-549, 2018 ACML 2018

Adversarial Neural Machine Translation

Lijun Wu WULIJUN3@MAIL2.SYSU.EDU.CN
Sun Yat-sen University

Yingce Xia YINGCE.XIA@GMAIL.COM
University of Science and Technology of China

Fei Tian FETIA@MICROSOFT.COM
Li Zhao LIZO@MICROSOFT.COM
Tao Qin TAOQIN@MICROSOFT.COM
Microsoft Research

Jianhuang Lai STSLJH@SYSU.EDU.CN
Sun Yat-sen University

Tie-Yan Liu TYLIU@MICROSOFT.COM
Microsoft Research

Editors: Jun Zhu and Ichiro Takeuchi

Abstract

In this paper, we study a new learning paradigm for neural machine translation (NMT). Instead of maximizing the likelihood of the human translation as in previous works, we minimize the distinction between human translation and the translation given by an NMT model. To achieve this goal, inspired by the recent success of generative adversarial networks (GANs), we employ an adversarial training architecture and name it as Adversarial-NMT. In Adversarial-NMT, the training of the NMT model is assisted by an adversary, which is an elaborately designed $2D$ convolutional neural network (CNN). The goal of the adversary is to differentiate the translation result generated by the NMT model from that by human. The goal of the NMT model is to produce high quality translations so as to cheat the adversary. A policy gradient method is leveraged to co-train the NMT model and the adversary. Experimental results on English→French and German→English translation tasks show that Adversarial-NMT can achieve significantly better translation quality than several strong baselines.

Keywords: Adversarial training, Generative adversarial networks, Neural machine translation.

1. Introduction

Neural machine translation (NMT) (Bahdanau et al., 2015; Cho et al., 2014) has drawn more and more attention in both academia and industry (Jean et al., 2015; Luong and Manning, 2016; Sennrich et al., 2016; Wu et al., 2016). Compared with traditional statistical machine translation (SMT) (Koehn et al., 2003), NMT achieves similar or even better translation results in an end-to-end framework. The maximum likelihood estimation (MLE) training and encoder-decoder framework, together with attention mechanisms (Bahdanau et al., 2015) grant NMT with the ability to better translate sentences.

Despite its success, the translation quality of latest NMT systems is still far from satisfactory and there remains large room for improvement. For example, NMT usually adopts the maximum likelihood estimation principle for training, i.e., to maximize the probability of the target ground-truth sentence conditioned on the source sentence. Such an objective does not guarantee the translation results from NMT model to be natural, sufficient, and accurate compared with ground-truth translation written by human. There are previous works (Bahdanau et al., 2017; Ranzato et al., 2016; Shen et al., 2016) that aim to alleviate such limitations of maximum likelihood training, by adopting sequence level objectives (e.g., directly maximizing BLEU (Papineni et al., 2002)), to reduce the objective inconsistency between NMT training and inference. Yet somewhat improved, such objectives still cannot fully bridge the gap between NMT translations and ground-truth translations.

In this paper, we adopt a thoroughly different training objective for NMT, targeting at directly minimizing the difference between human translation and the translation given by an NMT model. To achieve this goal, inspired by the recent success of generative adversarial networks (GANs) (Goodfellow et al., 2014a), we design an adversarial training protocol for NMT and name it as Adversarial-NMT. In Adversarial-NMT, besides the typical NMT model, an adversary is introduced to distinguish the translation generated by NMT from that by human (i.e., ground-truth). Meanwhile, the NMT model tries to improve its translation results so that it can successfully *cheat* the adversary.

These two modules in Adversarial-NMT are jointly trained, and their performances get mutually improved. In particular, the discriminative power of the adversary can be improved by learning from more and more training samples (both positive ones generated by human and negative ones sampled from NMT); and the ability of the NMT model in cheating the adversary can be improved by taking the output of the adversary as reward. In this way, the NMT translations are *professor forced* (Lamb et al., 2016) to be as close as possible to the human translations.

Different from previous GANs, which assume the existence of a generator in continuous space, in our proposed framework, the NMT model is not a typical generative model in continuous space, but instead, a probabilistic transformation that maps a source language sentence to a target language sentence, both in discrete space. Such differences make it necessary to design both new network architectures and optimization methods to make adversarial training possible for NMT. We therefore on one aspect, leverage a specially designed $2D$ pconvolutional neural network (CNN) model as adversary, which takes the (source, target) sentence pair as input; on the other aspect, we turn to a policy gradient method named REINFORCE (Williams, 1992), widely used in reinforcement learning literature (Sutton and Barto, 1998), to guarantee the two modules are effectively optimized in an adversarial manner. We conduct extensive experiments, which demonstrate that Adversarial-NMT can achieve significantly better translation results than traditional NMT models with even much larger vocabulary size and higher model complexity.

2. Related Work

End-to-end neural machine translation (NMT) (Bahdanau et al., 2015; Jean et al., 2015; Sutskever et al., 2014; Wu et al., 2016) has drawn a lot of attention from the community. A typical NMT system is built on the RNN based encoder-decoder framework. In such

a framework, the encoder RNN sequentially processes the words in the source language sentence into fixed length vectors, and then the decoder RNN works on the output vectors of the encoder to generate the translation sentence in the target language. NMT typically adopts the principle of maximum likelihood estimation (MLE) for training, i.e., maximizing the per-word likelihood of target sentence. Other training criteria, such as minimum risk training (MRT) based on reinforcement learning (Ranzato et al., 2016; Shen et al., 2016) and translation reconstruction (Tu et al., 2016), are shown to improve over such word level MLE principle since these objectives take the translation sentence as a whole.

The training principle we propose is based on the spirit of generative adversarial networks (GANs) (Goodfellow et al., 2014a; Salimans et al., 2016), or more generally, adversarial training (Goodfellow et al., 2014b). In adversarial training, a discriminator and a generator competes with each other, forcing the generator to produce high quality outputs that are able to fool the discriminator. Adversarial training typically is widely used in image generation (Goodfellow et al., 2014a; Reed et al., 2016), with few attempts in natural language processing tasks (Yu et al., 2017), in which it is difficult to propagate the error signals from the discriminator to the generator through the discretely generated natural language tokens. Yu et al. (2017) alleviates such a difficulty by reinforcement learning approach for speech language generation, poem generation and music generation, and Zhang et al. (2017) proposes approximated discretization with soft-argmax function in the adversarial training to generate text. However, there are few efforts on adversarial training for sequence-to-sequence task. Li et al. (2017) adopts the adversarial training with recurrent neural network based discriminator in the dialogue generation task. In parallel to our work, Yang et al. (2018) also adapts the generative adversarial networks into neural machine translation. However, our work is different from the above works in the following aspects: 1) In terms of the discriminator, Yang et al. (2018) follows Yu et al. (2017) and uses two separate CNNs as discriminator to model the source sentence and target sentence representation independently, and then the two sentence representations are concatenated for classification. Different from their two independent CNNs, we adopt a $2D$ convolutional architecture which is built directly on the interaction space between the source and target sentences. Acting in this way, the two sentences meet before their own high-level representations, while still retaining the space for the individual development of abstraction of each sentence. Our method therefore can better model the semantic relationship between source sentence and target sentence, which has been verified in previous work (Hu et al., 2014). 2) In terms of optimization efficiency, to apply reinforcement learning approach, the Monte-Carlo search is used in every step to get the intermediate action-value score (Yang et al., 2018; Yu et al., 2017), which is computationally cost. We only use one sample from a trajectory to estimate the terminal reward. To avoid bringing high variance into training procedure, we use a moving average of the historical reward values to set as a reward baseline (Weaver and Tao, 2001). Therefore, our model is faster and computationally efficient, while maintaining good accuracy. Detailed training strategies and model structures will be described in the next Section 3.

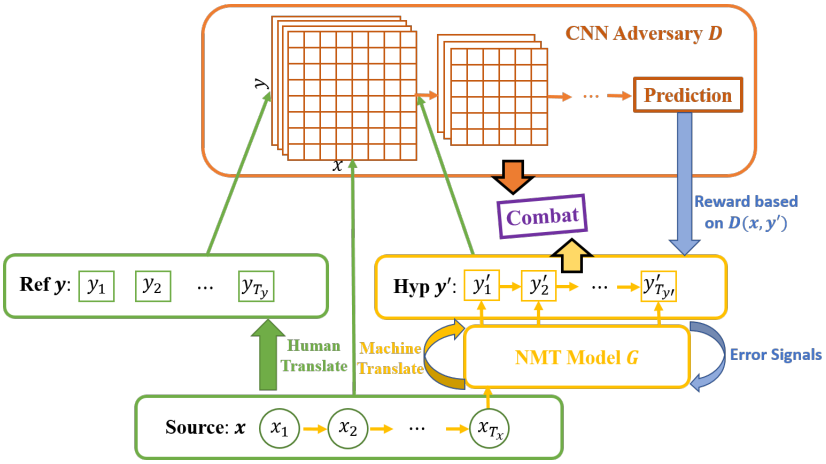


Figure 1: The Adversarial-NMT framework. 'Ref' is short for 'Reference' which means the ground-truth translation and 'Hyp' is short for 'Hypothesis', denoting model translation sentence. All the yellow parts denote the NMT model G, which maps a source sentence x to a translation sentence. The red parts are the adversary network D, which predicts whether a given target sentence is the ground-truth translation of the given source sentence x. G and D combat with each other, generating both sampled translation y' to train D, and the reward signals to train G by policy gradient (the blue arrows).

3. Adversarial-NMT

The overall framework of our proposed Adversarial-NMT is shown in Figure 1. There are two main components in our framework, a generator G, which is an NMT model used to translate sentences, and an adversary model D, which is used to distinguish the translation generated by NMT model from that by human. Let $(x = \{x_1, x_2, ..., x_{T_x}\}, y = \{y_1, y_2, ..., y_{T_y}\})$ be a bilingual aligned sentence pair for training, where x_i is the i-th word in the source sentence and y_j is the j-th word in the target sentence, T_x and T_y are the number of words in x and y respectively. Let y' denote the translation sentence of the source sentence x generated by G. Intuitively, the more similar y' is to y, the better translation quality y' has. Therefore, inspired by the success of GAN, we explicitly force y' to be similar to y in an adversarial manner. We introduce an adversary network D to differentiate human translation from machine translation. The objective of the NMT model G is to produce a target sentence as similar as the human translation so as to fool the adversary.

3.1. NMT Model

We adopt the recurrent neural network (RNN) based encoder-decoder as the NMT model to seek a target language translation y' given source sentence x. In particular, a probabilistic mapping $G(y|x)$ is firstly learnt and the translation result $y' \sim G(\cdot|x)$ is sampled from it. To be specific, given source sentence x and previously generated words $y_{<t}$, the probability

of generating word y_t is:

$$G(y_t|y_{<t}, x) = \rho(y_{t-1}, r_t, c_t), \tag{1}$$

$$r_t = g(r_{t-1}, y_{t-1}, c_t), \tag{2}$$

where ρ is the non-linear function like softmax, r_t is the decoding state from decoder at time t. Here g is the recurrent unit such as the long short term memory (LSTM) unit (Hochreiter and Schmidhuber, 1997) or gated recurrent unit (GRU) (Cho et al., 2014), and c_t is a distinct source representation at time t calculated by an attention mechanism (Bahdanau et al., 2015):

$$c_t = \sum_{i=1}^{T_x} \alpha_{it} h_i, \tag{3}$$

$$\alpha_{it} = \frac{\exp\{a(h_i, r_{t-1})\}}{\sum_k \exp\{a(h_k, r_{t-1})\}}, \tag{4}$$

where T_x is the source sentence length, $a(\cdot, \cdot)$ is a feed-forward neural network, and h_i is the hidden state from RNN encoder computed by h_{i-1} and x_i:

$$h_i = f(h_{i-1}, x_i). \tag{5}$$

The translation result y' can be sampled from $G(\cdot|x)$ either in a greedy way or using beam search (Sutskever et al., 2014) at each timestep.

3.2. Adversary Model

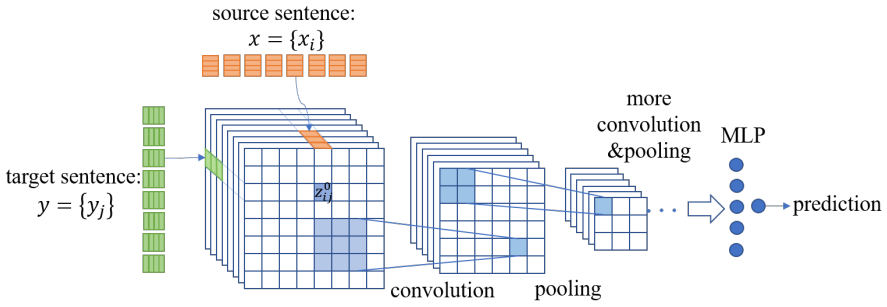


Figure 2: The CNN adversary framework.

The adversary is used to differentiate the model translation result y' from the ground-truth translation y, given the source language sentence x. To achieve that, one needs to measure the translation matching degree of (source, target) sentence pair (x, y). We turn to convolution neural network (CNN) for this task (Hu et al., 2014; Yin et al., 2016). With the layer-by-layer convolution and pooling strategies, CNN is able to accurately capture the hierarchical correspondence of (x, y) at different abstraction levels.

The general structure is shown in Figure 2. Specifically, given a sentence pair (x, y), we first construct an image-like representation $z^{(0)}$ by simply concatenating the embedding

vectors of words in x and y. That is, for i-th word x_i in the source sentence x and j-th word y_j in the target sentence y, we have the following feature map:

$$z_{i,j}^{(0)} = [x_i^T, y_j^T]^T.$$

Based on such an image-like representation, we perform convolution on every 3×3 window, with the purpose to capture the correspondence between segments in x and segments in y by the following feature map of type f:

$$z_{i,j}^{(1,f)} = \sigma(W^{(1,f)} \hat{z}_{i,j}^{(0)} + b^{(1,f)}),$$

where $\hat{z}_{i,j}^{(0)} = [z_{i-1:i+1,j-1:j+1}^{(0)}]$ represents the 3×3 window, and $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function.

After that we perform a max-pooling in non-overlapping 2×2 window:

$$z_{i,j}^{(2,f)} = \max(\{z_{2i-1,2j-1}^{(1,f)}, z_{2i-1,2j}^{(1,f)}, z_{2i,2j-1}^{(1,f)}, z_{2i,2j}^{(1,f)}\}).$$

Such a $2D$ architecture can make the source and target sentences meet before their own high-level representations, and still retain the space for the individual development of abstraction of each sentence. Therefore, the semantic relationship between the two sentences can be better modeled (Hu et al., 2014).

We could go on for more layers of convolution and max-pooling, aiming at capturing the correspondence at different levels of abstraction. The extracted features are then fed into a multi-layer perceptron, with sigmoid activation at the last layer to give the probability that (x, y) is from ground-truth data, i.e., $D(x, y)$. The optimization target of such CNN adversary is to minimize the cross-entropy loss for binary classification, with ground-truth data (x, y) as the positive instance while sampled data from G as the negative one.

3.3. Policy Gradient Algorithm to Train Adversarial-NMT

Following (Goodfellow et al., 2014a), we formulate the training of Adversarial-NMT as a two-player minimax game:

$$\min_G \max_D V(G, D), \tag{6}$$

where $V(G, D)$ is the value function of the two participants: NMT model G and adversary D. For Adversarial-NMT, $V(G, D)$ is specified as

$$\begin{aligned} V(G, D) = &\mathbb{E}_{(x,y)\sim P_{\text{data}}(x,y)}[\log D(x, y)] + \\ &\mathbb{E}_{x\sim P_{\text{data}}(x), y'\sim G(\cdot|x)}[\log(1 - D(x, y'))], \end{aligned} \tag{7}$$

where $D(x, y)$ is the probability that (x, y) is from the ground-truth data as described above, P_{data} is the unknown underlining distribution of the data. The parameters of G and D are denoted as Θ_G and Θ_D respectively.

Intuitively, as shown in Eqn.(7), the translation model G tries to produce high quality translation y' to fool the adversary D (the outer-operator min), while the adversary D tries to classify whether a translation result is from real data (i.e., ground-truth) or from translation model G (the inner-operator max).

We can use gradient based algorithm to optimize G and D. The most important thing is to calculate the gradients of $V(G,D)$ w.r.t Θ_G and Θ_D respectively. For ease of reference, denote $L(x,y;G,D)$ as

$$L(x,y;G,D) = \log D(x,y) + \sum_{y' \in \mathcal{Y}} G(y'|x) \log(1 - D(x,y')), \tag{8}$$

where $\mathcal{Y}$ is the collection of all possible translations. When the context is clear, we denote $L(x,y;G,D)$ as $L(x,y)$. Note to mention that $\mathbb{E}_{x,y\sim P_{\text{data}}(x,y)} L(x,y)$ is the value function $V(G,D)$.

The gradients $\nabla_{\Theta_D} L(x,y)$ and $\nabla_{\Theta_G} L(x,y)$ can be calculated as follows:

$$\nabla_{\Theta_D} L(x,y) = \nabla_{\Theta_D} \log D(x,y) + \sum_{y' \in \mathcal{Y}} G(y'|x) \nabla_{\Theta_D} \log(1 - D(x,y')), \tag{9}$$

$$\nabla_{\Theta_G} L(x,y) = \sum_{y' \in \mathcal{Y}} \nabla_{\Theta_G} G(y'|x) \log(1 - D(x,y')). \tag{10}$$

Since $\mathcal{Y}$ is exponentially large, $\nabla_{\Theta_D} L(x,y)$ and $\nabla_{\Theta_G} L(x,y)$ are intractable to calculate, and so are $\nabla_{\Theta_D} V(G,D)$ and $\nabla_{\Theta_G} V(G,D)$.

To tackle the above challenge, we leverage a Monte-Carlo based method to optimize D and G. Note that Eqn.(10) can be equivalently written as

$$\nabla_{\Theta_G} L(x,y) = \sum_{y' \in \mathcal{Y}} G(y'|x) \nabla_{\Theta_G} \log G(y'|x) \log(1 - D(x,y')). \tag{11}$$

Thus, we can estimate $\nabla_{\Theta_D} L(x,y)$ and $\nabla_{\Theta_G} L(x,y)$ by the following two steps: 1) for any x, sample a translation $y' \sim G(\cdot|x)$; 2) calculate $\nabla_{\Theta_D} \tilde{L}(x,y,y')$ and $\nabla_{\Theta_G} \tilde{L}(x,y,y')$, which are defined as

$$\nabla_{\Theta_D} \tilde{L}(x,y,y') = \nabla_{\Theta_D} \log D(x,y) + \nabla_{\Theta_D} \log(1 - D(x,y')), \tag{12}$$

$$\nabla_{\Theta_G} \tilde{L}(x,y,y') = \nabla_{\Theta_G} \log G(y'|x) \log(1 - D(x,y')). \tag{13}$$

One can easily verify that $\nabla_{\Theta_D} \tilde{L}(x,y,y')$ and $\nabla_{\Theta_G} \tilde{L}(x,y,y')$ are unbiased estimators of $\nabla_{\Theta_D} L(x,y)$ and $\nabla_{\Theta_G} L(x,y)$. As a result, taking the SGD algorithm as an example, we can update Θ_D and Θ_G as follows.

$$\Theta_D \leftarrow \Theta_D + \alpha_D \nabla_{\Theta_D} \tilde{L}(x,y,y');\ \Theta_G \leftarrow \Theta_G - \alpha_G \nabla_{\Theta_G} \tilde{L}(x,y,y'), \tag{14}$$

where α_D and α_G are the learning rates of D and G respectively, and $y' \sim G(\cdot|x)$.

Eqn.(13) is exactly the REINFORCE algorithm (Williams, 1992) in reinforcement learning literature. Using the language of reinforcement learning, in Adversarial-NMT: the NMT model acts as the *agent* with policy function $G(\cdot|x)$, and the translation sentence y' is the *action*. The *environment* is characterized via the source sequence x and the adversary model D, which provides the *reward* $-\log(1-D(x,y'))$ based on the classification accuracy based on y'.

The variance of such a Monte-Carlo estimation is high, leading to the instability issue in training as observed in previous works (Bahdanau et al., 2017; Ranzato et al., 2016).

To reduce the variance, a moving average of the historical reward values is set as a reward baseline (Weaver and Tao, 2001). Another way to reduce high variance is sampling multiple trajectories y' in each decoding step, by regarding G as the roll-out policy (Silver et al., 2016; Yu et al., 2017). However, empirically we find such approach is intolerably time-consuming, given that the decoding space in NMT is extremely large (the same as vocabulary size).

It is worth comparing our adversarial training with existing methods that directly maximize sequence level measure BLEU in training NMT models, such as minimal risk training (MRT) (Shen et al., 2016) and related approaches based on reinforcement learning (Bahdanau et al., 2017; Ranzato et al., 2016). We argue that Adversarial-NMT makes the optimization easier and has several advantages compared with these methods. First, the reward learned by our adversary D provides *rich and global information* to evaluate the translation quality, which goes beyond the BLEU's simple low-level n-gram matching criteria. Acting in this way provides much smoother objective compared with BLEU since the latter is highly sensitive (i.e., slight translation difference at word or phrase level is probably to induce significant BLEU variation). Second, the NMT model G and the adversary D in Adversarial-NMT co-evolve. The dynamics of adversary D makes NMT model G grows in an *adaptive* way rather than controlled by a fixed evaluation metric such as BLEU. Given the above two reasons, Adversarial-NMT makes the optimization process towards sequence level objectives more robust and better controlled, which is further verified by its superior performances to the aforementioned methods as reported in Section 4.

4. Experiments

4.1. Settings

We evaluate our model on two translation tasks: English→French translation (En→Fr for short) and German→English translation (De→En for short).

Dataset: For En→Fr translation, for the sake of fair comparison with previous works, we use the same dataset as (Bahdanau et al., 2015). The dataset is composed of a subset of WMT 2014 training corpus as training set, the combination of news-test 2012 and news-test 2013 as dev set and news-test 2014 as test set, which respectively contains roughly $12M$, $6k$ and $3k$ sentence pairs. The maximal sentence length is 50. We use top $30k$ most frequent English and French words as vocabulary and replace the other words as 'UNK' token.

For De→En translation, following previous works (Bahdanau et al., 2017; Ranzato et al., 2016), the dataset is from IWSLT 2014 evaluation campaign (Cettolo et al., 2014), consisting of training/dev/test corpus with approximately $153k$, $7k$ and $6.5k$ bilingual sentence pairs respectively. The maximal sentence length is also set as 50. The vocabulary for English and German corpus respectively include $22,822$ and $32,009$ most frequent words (Bahdanau et al., 2017), with other words replaced as a special token 'UNK'.

Models: In Adversarial-NMT, for the NMT model G, we use the same model structure as RNNSearch model (Bahdanau et al., 2015), an RNN based encoder-decoder framework with attention mechanism. A bidirectional GRU layer acts as the encoder and a unidirectional GRU layer acts as the decoder. Following the common practice in NMT (Jean et al., 2015), for En→Fr translation, the dimensions of word embedding and GRU hidden state are 620 and 1000 respectively, and for De→En translation they are both 256.

The adversary D is a CNN architecture starting with two convolution layers, where each one has 20 filters of size 3×3 and each one is followed by a max-pooling layer with filter size 2×2. After the convolution and pooling layers, one MLP layer is added with 20 hidden nodes, and eventually sigmoid activation is used to give the probability. The dimension of word embedding used in adversary D is the same as that for the corresponding G.

Training Procedure: To stabilize the training process, following (Shen et al., 2016; Tu et al., 2016), we first pre-train the NMT model G and obtain a warm-start model for Adversarial-NMT[1]. Then, the CNN adversary model D is also pre-trained using the data (x, y') where y' is sampled from this well-trained NMT model G, as well as the the ground-truth data (x, y).

Next, we use the warm-start NMT model G and adversary D to initialize our Adversarial-NMT and jointly train the two models. We optimize the NMT model G using vanilla SGD with mini-batch size 80 for En→Fr and 32 for De→En translation. Gradient clipping is used with clipping value 1 for En→Fr and 10 for De→En. The initial learning rate is chosen from cross-validation on dev set (0.02 for En→Fr, 0.001 for De→En) and we halve it every $80k$ iterations. For the adversary D, it is optimized using Nesterov SGD (Nesterov, 1983) with batch size 80 for En→Fr and 32 for De→En. The initial learning rate is 0.002 for En→Fr, 0.001 for De→En, both chosen according to the accuracy on the dev set. We fix the word embeddings of D during training. Batch normalization (Ioffe and Szegedy, 2015) is observed to significantly improve D's performance. Considering efficiency, the negative training data (x, y') used in D's training are generated using beam search with width 4. We terminate the training when the performance of NMT model G is not improved on dev set.

A key factor we find in successfully training G is that the combination of adversarial objective and MLE objective. That is, for any mini-batch, with equal probability, G is optimized by adversarial objective or MLE objective. Acting in this way significantly improves the stability in model training, which is also reported in other tasks such as language model (Lamb et al., 2016) and dialogue generation (Li et al., 2017). We conjecture the reason is that MLE acts as a regularizer to guarantee smooth model update, alleviating the negative effects brought by high gradient estimation variance of the one-step Monte-Carlo sample in REINFORCE.

All our models are implemented with Theano and trained on NVIDIA K40 GPU. For En→Fr translation tasks, it takes about one week; for De→En, it roughly takes 20 hours.

In generating translation results for evaluation, we set beam width as 4 for En→Fr, and 12 for De→En. The translation quality is measured by tokenized case-sensitive BLEU (Papineni et al., 2002) score[2].

4.2. Result on En→Fr translation

In Table 1 we provide the En→Fr translation result of Adversarial-NMT, together with several strong NMT baselines, such as the well representative attention-based NMT model RNNSearch (Bahdanau et al., 2015). In addition, to make our comparison comprehensive, we would like to cover several well acknowledged techniques whose effectiveness has been

1. The well-trained RNNSearch models serve as baselines and the BLEU scores are 29.92 for En→Fr and 23.70 for De→En. Those numbers are close to the ones reported in (Jean et al., 2015) and (Wiseman and Rush, 2016), as shown in Table 1 and 3 with symbol $*$.
2. https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

Table 1: Different NMT systems' performances on En→Fr translation. The default setting is single layer GRU with MLE training, and 30k word vocabulary, trained with no monolingual data, i.e., the RNNSearch model proposed by (Bahdanau et al., 2015). ∗: our warm-start RNNSearch baseline model. †: significantly better than (Shen et al., 2016) ($\rho < 0.05$).

System	System Configurations	BLEU
Representative end-to-end NMT systems		
Sutskever et al. (2014)	LSTM with 4 layers + 80k word vocabulary	30.59
Bahdanau et al. (2015)	RNNSearch∗	29.97[a]
Jean et al. (2015)	RNNSearch + UNK Replace	33.08
Jean et al. (2015)	RNNSearch + 500k word vocabulary + UNK Replace	34.11
Luong et al. (2015)	LSTM with 4 layers + 40k word vocabulary	29.50
Luong et al. (2015)	LSTM with 4 layers + 40k word vocabulary + PosUnk	31.80
Shen et al. (2016)	RNNSearch +Minimum Risk Training	31.30
Sennrich et al. (2016)	RNNSearch +Monolingual Data	30.40[b]
He et al. (2016)	RNNSearch+ Monolingual Data + Dual Learning	32.06
Adversarial-NMT		
this work	RNNSearch + Adversarial Training	31.91†
	RNNSearch + Adversarial Training + UNK Replace	34.78

a. Reported in Jean et al. (2015).
b. Reported in He et al. (2016).

verified to improve En→Fr translation by previously published works, including the leverage of 1) Using large vocabulary to handle rare words (Jean et al., 2015; Luong et al., 2015); 2) Different training objectives (Bahdanau et al., 2017; Ranzato et al., 2016; Shen et al., 2016) such as minimum risk training (MRT) to directly optimize evaluation measure (Shen et al., 2016), and dual learning to enhance both primal and dual tasks (e.g., En→Fr and Fr→En) (He et al., 2016); 3) Improved inference process such as beam search optimization (Wiseman and Rush, 2016) and postprocessing UNK (Jean et al., 2015; Luong et al., 2015); 4) Leveraging additional monolingual data (He et al., 2016; Sennrich et al., 2016).

From the table, we can clearly observe that Adversarial-NMT obtains satisfactory translation quality against baseline systems. In particular, it even surpasses the performances of other models with much larger vocabularies (Jean et al., 2015), deeper layers (Luong et al., 2015), much larger monolingual training corpus (Sennrich et al., 2016), and the goal of directly maximizing BLEU (Shen et al., 2016). In fact, as far as we know, Adversarial-NMT achieves state-of-the-art result (34.78) on En→Fr translation for single-layer GRU sequence-to-sequence models trained with only supervised bilingual corpus on news-test 2014 test set.

Human Evaluation: Apart from the comparison based on the objective BLEU score, to better appraise the performance of our model, we also involve human judgments as a subjective measure. Specifically, we generate the translation results for 500 randomly selected English sentences from En→Fr news-test 2014 set using both MRT and our Adversarial-NMT. Here MRT is chosen since it is the well representative of previous NMT methods which maximize sequence level objectives, achieving satisfactory results among all single

layer models (i.e., 31.30 in Table 1). Afterwards, we ask three human evaluators to choose the better one from the two versions of translated sentences. The evaluation process is conducted on Amazon mechanical turk[3] with all the evaluators to be native French speakers and familiar with English.

Table 2: Human evaluations for Adversarial-NMT and MRT on En→Fr translation. "286 (57.2%)" means that evaluator 1 made a decision that 286 (57.2%) out of 500 translations generated by Adversarial-NMT were better than MRT.

	Adversarial-NMT	MRT
evaluator 1	286 (57.2%)	214 (42.8%)
evaluator 2	310 (62.0%)	190 (38.0%)
evaluator 3	295 (59.0%)	205 (41.0%)
Overall	**891 (59.4%)**	**609 (40.6%)**

Result in Table 2 shows that 59.4% sentences are better translated by our Adversarial-NMT, compared with MRT. Such human evaluation further demonstrates the effectiveness of our model and matches the expectation that Adversarial-NMT provides more human desired translation.

Adversarial Training: Slow or Fast: We further analyze how to set the pace for training the NMT model G and adversary D, to make them combatting effectively. Specifically, for En→Fr translation, we inspect how dev set BLEU varies along adversarial training process with different initial learning rates for G (shown in 3(a)) and for D (shown in 3(b)), conditioned on the other one fixed.

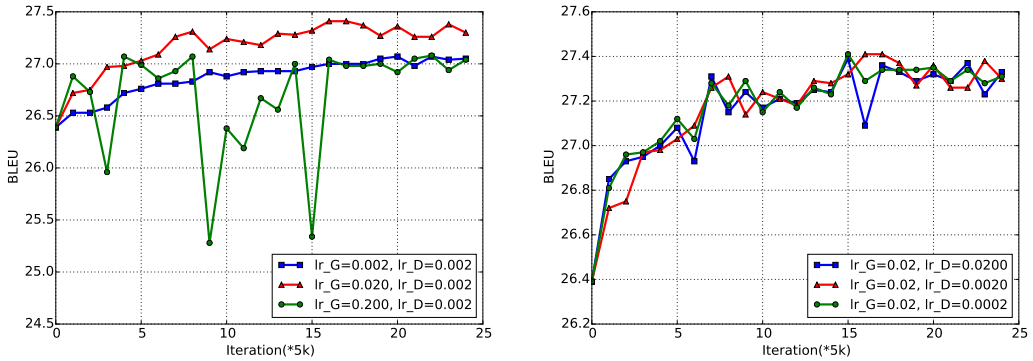
Overall speaking, these two figures show that Adversarial-NMT is much more robust with regard to the pace of D making progress than that of G, since the three curves in 3(b) grow in a similar pattern while curves in 3(a) drastically differ with each other. We conjecture the reason is that in Adversarial-NMT, CNN based D is powerful in classification tasks, especially when it is warm started with sampled data from RNNSearch. As a comparison, the translation model G is relatively weak in providing qualified translations. Therefore, training G needs careful configurations of learning rate: small value (e.g., 0.002) leads to slower convergence (blue line in 3(a)), while large value (e.g., 0.2) brings un-stability (green line in 3(a)). The proper learning rate (e.g. 0.02) induces G to make fast, meanwhile stable progress along training.

4.3. Result on De→En translation

In Table 3 we provide the De→En translation result of Adversarial-NMT, compared with some strong baselines such as RNNSearch (Bahdanau et al., 2015) and MRT (Shen et al., 2016).

Again, we can see that Adversarial-NMT performs best against other models from Table 3, achieves 27.94 BLEU score.

3. https://www.mturk.com



(*a*) D: Same learning rates; G: different learning rates.

(*b*) G: Same learning rates; D: different learning rates.

Figure 3: Dev set BLEUs during En→Fr Adversarial-NMT training process, with same learning rates for D, different learning rates for G in left 3(*a*), and same learning rates for G and different learning rates for D in right 3(*b*).

Table 3: Different NMT systems' performances on De→En translation. The default setting is single layer GRU encoder-decoder model with MLE training, i.e., the RNNSearch model proposed by (Bahdanau et al., 2015). ∗: our warm-start RNNSearch baseline model. †: significantly better than (Shen et al., 2016) ($\rho < 0.05$).

System	System Configurations	BLEU
Representative end-to-end NMT systems		
Bahdanau et al. (2015)	RNNSearch∗	23.87[a]
Ranzato et al. (2016)	CNN encoder + Sequence level REINFORCE	21.83
Bahdanau et al. (2017)	CNN encoder + Sequence level Actor-Critic	22.45
Wiseman and Rush (2016)	RNNSearch + Beam search optimization	25.48
Shen et al. (2016)	RNNSearch + Minimum Risk Training	25.84[b]
Adversarial-NMT		
this work	RNNSearch + Adversarial Training	26.98†
	RNNSearch + Adversarial Training + UNK Replace	27.94

a. Reported in (Wiseman and Rush, 2016).
b. Result from our implementation, and we reproduced their reported En→Fr result.

Effect of Adversarial Training: To better visualize and understand the advantages of adversarial training brought by Adversarial-NMT, we show several translation cases in Table 4. Concretely speaking, we give two De→En translation examples, including the source language sentence x, the ground-truth translation sentence y, and two NMT model translation sentences, respectively out from RNNSearch and Adversarial-NMT (trained after 20 epochs) and emphasize on their different parts by bold fonts which lead to different

Table 4: Cases-studies to demonstrate the translation quality improvement brought by Adversarial-NMT. We provide two De→En translation examples, with the source German sentence, ground-truth (reference) English sentence, and two translation results respectively provided by RNNSearch and Adversarial-NMT (A-NMT). $D(x, y')$ is the probability of model translation y' being ground-truth translation of x, calculated from the adversary D. Here BLEU is the sentence level translation bleu score. *: BLEU score is based on 1-gram, 2-gram, 3-gram, 4-gram match together, for this specific sentence, BLEU score is 0 since there is no 4-gram match.

Source x	ich weiß, dass wir es können , und soweit es mich betrifft ist das etwas ,was die welt jetzt braucht .	$D(x, y')$	BLEU
Reference y	i know that we can , and as far as i 'm concerned , that 's something the world needs right now .		
RNNSearch y'	i know we can do it , and as far as **it 's in time , what** the world needs now .	0.14	27.26
A-NMT y'	i know that we can , and as far as **it is to be something that** the world needs now .	0.67	50.28
Source x	wir müssen verhindern , dass die menschen kenntnis erlangen von dingen , vor allem dann , wenn sie wahr sind .	$D(x, y')$	BLEU
Reference y	we have to prevent people from finding about things , especially when they are true .		
RNNSearch y'	we need to prevent people who are able to know **that people have to do** , especially if they are true .	0.15	0.00*
A-NMT y'	we need to prevent people who are able to know **about things** , especially if they are true .	0.93	25.45

translation quality. For each model translation y', we also list $D(x, y')$ in the third column, i.e., the probability that the adversary D regards y' as ground-truth, and the sentence level BLEU score of y' in the last column.

Since RNNSearch model acts as the warm start for training Adversarial-NMT, its translation could be viewed as the result of Adversarial-NMT at its initial phase. Therefore, from Table 4, we can observe:

- With adversarial training goes on, the quality of translation sentence output by G gets improved, both in terms of subjective feelings and BLEU score as a quantitative measure.
- Correspondingly, the translation quality growth makes the adversary D deteriorated, as shown by D's successful recognition of y' by RNNSearch as translated from model, whereas D makes mistakes in classifying y' out from Adversarial-NMT as ground-truth (by human).

Compare to other works: In Table 3, we compare our work with existing highly acknowledged works. We can see that our proposed method significantly outperforms the above baselines, which demonstrates the effectiveness of our Adversarial-NMT. Besides, we also compare our approach with Yang et al. (2018) in De→En translation task, which is in parallel to our work. As discussed before, they adopt two separate and independent

CNNs as discriminator to model the source sentence and target sentence representations. Different from their ways, we directly apply a $2D$ CNN to model the interaction between source and target sentences. In this way, the two sentences meet before their own high-level representations, while still retaining the individual abstract space of each sentence. Therefore the relationship between the source and target sentences is better modeled, which is more helpful for our adversary to classify whether a sentence is a nature one. The eventual translation quality benefits from the $2D$ CNN: Yang et al. (2018) achieves 26.57 [4] BLEU score while our approach achieves 26.98 BLEU score. On the other hand, in terms of the time efficiency, their method adopts the Monte-Carlo search in each step to get the intermediate score, and the experiment roughly takes about 26 hours. We only use one sample from a trajectory to train the model, together with using a moving average of the historical reward values to set as a reward baseline. Our experiment only takes about 20 hours. By showing the better accuracy and less training cost, we successfully demonstrate the effectiveness and efficiency of our Adversarial-NMT structure.

5. Conclusion

We in this paper propose a novel and intuitive training objective for NMT, that is to force the translation results to be as similar as ground-truth translations generated by human. Such an objective is achieved via an adversarial training framework called Adversarial-NMT which complements the original NMT model with a CNN based adversary. Adversarial-NMT adopts both new network architectures to reflect the mapping within (source, target) sentence, and an efficient policy gradient algorithm to tackle the optimization difficulty brought by the discrete nature of machine translation. The experiments on English→French andp German→English translation tasks clearly demonstrate the effectiveness of such adversarial training method for NMT.

As to future works, with the hope of achieving new state-of-the-art performance for NMT system, we plan to fully exploit the potential of Adversarial-NMT by combining it with other powerful methods listed in Subsection 4.2, such as training with a larger vocabulary, minimum risk training principle, and deep structures. We additionally would like to emphasize and explore the feasibility of adversarial training to other text processing tasks, such as image caption, dependency parsing, and sentiment classification.

References

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

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. In *ICLR*, 2017.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. Report on the 11th iwslt evaluation campaign, iwslt 2014. 2014.

4. We provide the result by running the experiments based on their public code: https://github.com/ZhenYangIACAS/NMT_GAN/tree/master/RNNsearch_GAN.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

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

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

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

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *NIPS*, 2014.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML-15*, 2015.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *ACL*, 2015.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *NAACL*, 2003.

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

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

Minh-Thang Luong and Christopher D Manning. Achieving open vocabulary neural machine translation with hybrid word-character models. In *ACL*, 2016.

Minh-thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. In *ACL*, 2015.

Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/k2). In *Doklady an SSSR*, 1983.

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

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *ICLR*, 2016.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and X-i Chen. Improved techniques for training gans. In *NIPS*, 2016.

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

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

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, , et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016.

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

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. Neural machine translation with reconstruction. In *AAAI*, 2016.

Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning. In *UAI*, 2001.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 1992.

Sam Wiseman and Alexander M. Rush. Sequence-to-sequence learning as beam-search optimization. In *EMNLP*, 2016.

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

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. Improving neural machine translation with conditional sequence generative adversarial nets. In *NAACL*, 2018.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. 2016.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: sequence generative adversarial nets with policy gradient. In *AAAI*, 2017.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. Adversarial feature matching for text generation. In *ICML*, 2017.