Bidirectional Generative Adversarial Networks for Neural Machine Translation

Zhirui Zhang[†]*, **Shujie Liu**[§], **Mu Li**[¶], **Ming Zhou**[§], **Enhong Chen**[†]
[†]University of Science and Technology of China, Hefei, China
[§]Microsoft Research Asia
[†]zrustc11@gmail.com [†]cheneh@ustc.edu.cn
[§]{shujliu,mingzhou}@microsoft.com [¶]limugx@outlook.com

Abstract

Generative Adversarial Network (GAN) has been proposed to tackle the exposure bias problem of Neural Machine Translation (NMT). However, the discriminator typically results in the instability of the GAN training due to the inadequate training problem: the search space is so huge that sampled translations are not sufficient for discriminator training. To address this issue and stabilize the GAN training, in this paper, we propose a novel Bidirectional Generative Adversarial Network for Neural Machine Translation (BGAN-NMT), which aims to introduce a generator model to act as the discriminator, whereby the discriminator naturally considers the entire translation space so that the inadequate training problem can be alleviated. To satisfy this property, generator and discriminator are both designed to model the joint probability of sentence pairs, with the difference that, the generator decomposes the joint probability with a source language model and a source-to-target translation model, while the discriminator is formulated as a target language model and a target-to-source translation model. To further leverage the symmetry of them, an auxiliary GAN is introduced and adopts generator and discriminator models of original one as its own discriminator and generator respectively. Two GANs are alternately trained to update the parameters. Experiment results on German-English and Chinese-English translation tasks demonstrate that our method not only stabilizes GAN training but also achieves significant improvements over baseline systems.

1 Introduction

The past several years have witnessed the rapid development of Neural Machine Translation (NMT) (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2014), from catching up with Statistical Machine Translation (SMT) (Koehn et al., 2003; Chiang, 2007) to outperforming it by significant margins on many languages (Sennrich et al., 2016; Gehring et al., 2017; Vaswani et al., 2017; Hassan et al., 2018). The most common approach to training NMT is to maximize the conditional log-probability of the correct translation given the source sentence. However, as argued in Bengio et al. (2015), the Maximum Likelihood Estimation (MLE) principle suffers from so-called exposure bias in the inference stage: the model predicts next token conditional on its previously predicted ones that may be never observed in the training data. To address this problem, much recent work attempts to reduce the inconsistency between training and inference, such as adopting sequence-level objectives and directly maximizing BLEU scores (Bengio et al., 2015; Ranzato et al., 2015; Shen et al., 2016; Wiseman and Rush, 2016).

Generative Adversarial Network (GAN) (Goodfellow et al., 2014) is another promising framework for alleviating exposure bias problem and recently shows remarkable promise in NMT (Yang et al., 2017; Wu et al., 2017). Formally, GAN consists of two "adversarial" models: a generator and a discriminator. In machine translation, NMT model is used as the generator that produces translation candidates given a source sentence, and another neural network is introduced to serve as the discriminator, which takes sentence pairs as input and distinguishes whether a given sentence pair is real or generated. Adversarial training between the two models involves optimizing a min-max objective, in which, the discriminator learns to distinguish whether a given data instance is real or fake, and the generator learns to confuse the discriminator by generating high-quality translation candidates. Since the generated data is based on

*This work was done when the first author was the intern at Microsoft Research Asia.

Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018), pages 190–199
Brussels, Belgium, October 31 - November 1, 2018. ©2018 Association for Computational Linguistics

discrete symbols (words), we usually adopt policy gradient method (Yu et al., 2017) to update model parameters of the generator. Specifically, given a bunch of translation candidates sampled from the generator, confidence scores calculated by the discriminator are employed as rewards to update the generator.

However, in this training process, the discriminator typically suffers from inadequate training problem, leading to the instability of GAN training. In practice, sampling large translation candidates is time-consuming for NMT system, so we only use a few samples to train the discriminator. For a given source sentence, there is usually only one positive example (real target sentence). If the sampled negative examples are also few, the discriminator will easily overfit to the data. This is the inadequate training problem for the discriminator. In such a case, rewards calculated by the discriminator could be biased, especially for unobserved samples. These biased rewards will provide a wrong signal to the generator and make it update incorrectly, resulting in performance degradation of the generator. Since such issue can occur repeatedly throughout the entire training process, GAN training becomes unstable and the performance of generator will drop drastically.

On the other hand, the generator has well-designed properties that benefit the discriminator, since it models the probability distribution over the entire translation space so that the generator does not overfit to observed samples, while prior knowledge for unobserved samples is naturally considered. At the same time, the generator also exhibits a certain ability to identify whether a given data instance is good enough. For example, target-to-source translation model serves as the discriminator to improve source-to-target translation model (He et al., 2016; Tu et al., 2017). Inspired by this, we propose a novel Bidirectional Generative Adversarial Network for Neural Machine Translation (aka BGAN-NMT), which employs a generator model to perform the role of the discriminator so as to handle inadequate training problem and stabilize GAN training. To satisfy this property, both generator and discriminator of original GAN are designed to model the joint probability of sentence pairs, with the difference that, the generator model A is decomposed into a source language model and a source-to-target translation model, while the discriminator model B is formulated as a target language model and a target-to-source translation model. Intuitively, we can also leverage A to act as the discriminator to improve B, and then improved B reversely serves as a better discriminator to guide the training of A. To make use of this symmetry, we bring in an auxiliary GAN that adopts generator and discriminator models of original one as its own discriminator and generator respectively. Then we design a joint training algorithm to alternately utilize these two GANs to update the source-to-target and target-to-source translation models.

Our experiments are conducted on German-English and Chinese-English translation data sets. Experimental results demonstrate that our BGAN-NMT not only achieves the stability of GAN training but also significantly improves translation performance over baseline systems.

2 Background

2.1 Neural Machine Translation

Attention-based NMT model (Bahdanau et al., 2014) is adopted as the source-to-target and target-to-source translation models used in our BGAN-NMT. The attention-based NMT system is implemented as an encoder-decoder framework with recurrent neural networks (RNN), which can be Gated Recurrent Unit (GRU) (Cho et al., 2014) or Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) networks in practice.

2.1.1 Encoder-Decoder Framework

The encoder reads the source sentence $x = (x_1, x_2, ..., x_T)$ and transforms it into a sequence of hidden states $h = (h_1, h_2, ..., h_T)$ using a bi-directional RNN. The decoder uses another RNN to generate the translation $y = (y_1, y_2, ..., y_{T'})$ based on the hidden states h. At each time stamp i, the conditional probability of each word y_i from a target vocabulary V_y is computed with

$$p(y_i|y_{<i}, h) = g(y_{i-1}, z_i, c_i), \quad (1)$$

where z_i is the i_{th} hidden state of the decoder, which is calculated conditioned on the previous hidden state z_{i-1}, previous word y_{i-1} and the source context vector c_i:

$$z_i = \text{RNN}(z_{i-1}, y_{i-1}, c_i), \quad (2)$$

The source context vector c_i is a weighted sum of the hidden states $(h_1, h_2, ..., h_T)$ with the coeffi-

cients $\alpha_1, \alpha_2, \ldots, \alpha_T$ calculated with

$$\alpha_t = \frac{\exp\left(a(h_t, z_{i-1})\right)}{\sum_k \exp\left(a(h_k, z_{i-1})\right)} \quad (3)$$

where a is a feed-forward neural network with a single hidden layer.

2.1.2 MLE Training

NMT systems are usually trained to maximize the conditional log-probability of the correct translation given a source sentence with respect to the parameters θ of the model:

$$\theta^* = \arg\max_{\theta} \sum_{n=1}^{N} \sum_{i=1}^{|y^n|} \log p(y_i^n | y_{<i}^n, x^n) \quad (4)$$

where N is size of the training corpus, and $|y^n|$ is the length of the target sentence y^n. However, MLE training suffers from exposure bias problem: in training stage, the history of any target word is correct and has been observed in the training data, while during testing, the model predicts next token conditioned on its previously predicted ones that may be never observed in the training data. To solve this problem, reinforcement learning methods are used to sample translation candidates, based on which, rewards are calculated and utilized to update the parameters. GAN follows the same way to solve exposure bias problem and rewards are computed by the discriminator.

2.2 Generative Adversarial Network

As a new paradigm of training generative models, GAN (Goodfellow et al., 2014) has been successfully applied in computer vision tasks (Radford et al., 2015; Arjovsky et al., 2017). Conceptually, GAN consists of two "adversarial" models: a generator G that captures the data distribution, and a discriminator D that estimates the probability that a sample is sampled from the training data rather than from G. When GAN is used for NMT, NMT model is employed as G, and CNN-based or RNN-based neural networks serve as D (Yang et al., 2017; Wu et al., 2017). During adversarial training, G and D play a two-player minmax game with the following value function $V(D, G)$:

$$\min_G \max_D V(D, G) = \mathrm{E}_{(x,y)\sim P_d(x,y)} \left[\log D(x, y)\right] + \mathrm{E}_{(x,y')\sim P_G(x,y)} \left[\log(1 - D(x, y'))\right] \quad (5)$$

where (x, y) is a sentence pair, P_d represents the data distribution and P_G denotes the generator distribution. With this objective function, the discriminator learns to distinguish whether sentence pair is real (sampled from bilingual corpus) or fake (generated by G), and the generator tries to confuse the discriminator by generating high-quality translation samples.

In practice, policy gradient method (Yu et al., 2017) is usually used to calculate gradients for the generator due to discrete symbols (words). To update the generator model, translation candidates are firstly sampled, for which rewards are calculated using the discriminator. With these rewards, we can compute gradients and run back-propagation to update the generator. In such a training process, real target sentence and sampled translation candidates are used as positive and negative examples of discriminator training respectively. Due to the computation cost, we cannot generate many negative examples, so that the discriminator is easy to overfit. The overfitted discriminator will give biased signals to the generator and make it update incorrectly, leading to the instability of the generator training. Wu et al. (2017) found that combining adversarial training objective with MLE can significantly improve the stability of generator training, which is also reported in language model and neural dialogue generation (Lamb et al., 2016; Li et al., 2017). Actually, although this method leverages real translation signal to guide the generator and alleviate the effect of overfitted discriminator, it cannot deal with the inadequate training problem of the discriminator, which essentially plays a more important role in GAN training.

3 Bidirectional Generative Adversarial Network

In GAN for NMT, the generator does not suffer from the inadequate training problem, because the generator is proposed to model probability distribution over the entire translation space (maximizing probability of one translation candidate means minimizing probabilities of the others). At the same time, the generator exhibits a certain ability to discriminate good sentence pairs, for example, target-to-source translation model is used to score samples generated from source-to-target translation model. Thus, introducing a generator model to perform the role of the discriminator is expected

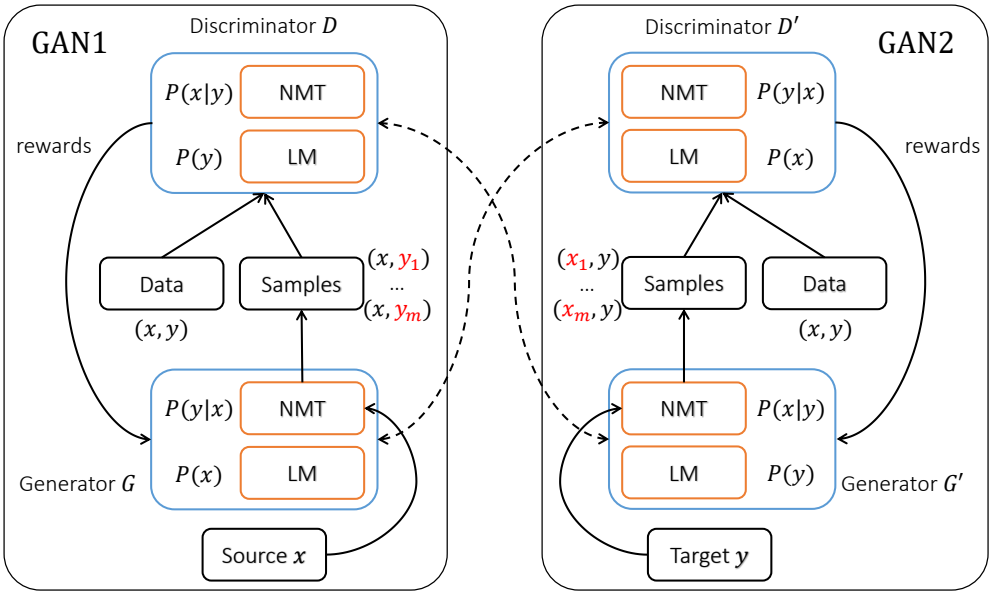


Figure 1: The architecture of BGAN-NMT consisting of two GANs. The dotted line represents that $GAN2$ adopts both generator and discriminator models of $GAN1$ but interchanges their roles.

to address the inadequate training problem and stabilize GAN training. Based on these observations, we design a Bidirectional Generative Adversarial Network for Neural Machine Translation, named as BGAN-NMT.

As illustrated in Figure 1, the overall architecture of BGAN-NMT consists of an original GAN ($GAN1$) and an auxiliary GAN ($GAN2$). Both generator and discriminator of original GAN are defined to model the joint probability of sentence pairs $P(x, y)$. Formally, $P(x, y)$ can be decomposed in two equivalent ways: $P(x, y) = P(x)P(y|x)$ and $P(x, y) = P(y)P(x|y)$, and they are used as generator G and discriminator D for $GAN1$ respectively. Further, the generator model can be decomposed into a source language model and a source-to-target translation model, while the discriminator can be formulated as a target language model and a target-to-source translation model. Auxiliary GAN ($GAN2$) employs G and D of $GAN1$ as its own discriminator D' and generator G' to better exploit the symmetry between G and D. The following of this section details the objective function and joint training algorithm for BGAN-NMT.

3.1 Training Objective

As G and D are defined as $P(x)P(y|x)$ and $P(y)P(x|y)$ respectively, the adversarial training objective $V(D, G)$ of $GAN1$ in Equation 5 can be rewritten as

$$\min_G \max_D V(D, G) = \mathrm{E}_{(x,y)\sim P_d(x,y)} [\log P(x|y)P(y)] + \mathrm{E}_{x\sim P_d(x), y'\sim P(y|x)} [\log(1 - P(x|y')P(y'))] \quad (6)$$

which means, given a source sentence x, source-to-target translation model $P(y|x)$ tries to generate high quality translation y' to fool the discriminator $P(x|y)P(y)$, while target-to-source translation model $P(x|y)$ and language model $P(y)$ learn to distinguish translation candidates from real sentence pairs. In our implementations, two language models $P(x)$ and $P(y)$ are fixed to reduce training complexity.

For discriminator D, D is trained with the ground-truth sentence pair (x, y) and the generated sample (x, y') from G, respectively as positive and negative examples. Formally, the objective function of D is to maximize $V(D, G)$:

$$L_D = \mathrm{E}_{(x,y)\sim P_d(x,y)} [\log P(x|y)P(y)] + \mathrm{E}_{x\sim P_d(x), y'\sim P(y|x)} [\log (1 - P(x|y')P(y'))] \quad (7)$$

Since $P(y)$ is fixed, the gradient of parameter θ_D for the target-to-source translation model $P(x|y)$ is calculated as:

$$\frac{\partial L_D}{\partial \theta_D} = \mathrm{E}_{(x,y)\sim P_d(x,y)} \left[\frac{\partial \log P(x|y)}{\partial \theta_D}\right] + \mathrm{E}_{x\sim P_d(x), y'\sim P(y|x)} \left[\left(1 - \frac{1}{1 - P(x|y')P(y')}\right) \frac{\partial \log P(x|y')}{\partial \theta_D}\right] \quad (8)$$

in which $\frac{\partial \log P(x|y)}{\partial \theta_D}$ is the gradient specified with standard sequence-to-sequence NMT network.

For generator G, following Goodfellow (2016), the objective of training G is to maximize the expected rewards (probability of D) instead of directly minimizing $V(D,G)$:

$$L_G = \mathrm{E}_{x\sim P_d(x),y'\sim P(y|x)}\left[P(x|y')P(y')\right] \quad (9)$$

Since the output of the generator G is a sequence of discrete symbols (words), policy gradient is used to update the parameters, and then the gradient of parameter θ_G for source-to-target translation model $P(y|x)$ can be calculated as:

$$\frac{\partial L_G}{\partial \theta_G} = \mathrm{E}_{x\sim P_d(x),y'\sim P(y|x)}\left[P(x|y')P(y')\frac{\partial \log P(y'|x)}{\partial \theta_G}\right] \quad (10)$$

By exchanging generator and discriminator models of $GAN1$, we introduce $GAN2$, in which the original G is used as the discriminator D' and original D serves as the generator G'. Similarly, the adversarial training objective $V(D',G')$ of $GAN2$ is defined as:

$$\begin{aligned}\min_{G'}\max_{D'} V(D',G') = {} & \mathrm{E}_{(x,y)\sim P_d(x,y)}\left[\log P(y|x)P(x)\right] \\ & + \mathrm{E}_{y\sim P_d(y),x'\sim P(x|y)}\left[\log(1-P(y|x')P(x'))\right]\end{aligned} \quad (11)$$

According to this adversarial training objective, the objective functions of D' and G' are defined as following:

$$\begin{aligned}L_{D'} = {} & \mathrm{E}_{(x,y)\sim P_d(x,y)}\left[\log P(y|x)P(x)\right] \\ & + \mathrm{E}_{y\sim P_d(y),x'\sim P(x|y)}\left[\log(1-P(y|x')P(x'))\right]\end{aligned} \quad (12)$$

$$L_{G'} = \mathrm{E}_{y\sim P_d(y),x'\sim P(x|y)}\left[P(y|x')P(x')\right] \quad (13)$$

where the gradients of parameters $\theta_{D'} = \theta_G$ for $P(y|x)$ and $\theta_{G'} = \theta_D$ for $P(x|y)$ can be respectively calculated as:

$$\begin{aligned}\frac{\partial L_{D'}}{\partial \theta_G} = {} & \mathrm{E}_{(x,y)\sim P_d(x,y)}\left[\frac{\partial \log P(y|x)}{\partial \theta_G}\right] \\ & + \mathrm{E}_{y\sim P_d(y),x'\sim P(x|y)} \\ & \left[(1-\frac{1}{1-P(y|x')P(x')})\frac{\partial \log P(y|x')}{\partial \theta_G}\right]\end{aligned} \quad (14)$$

$$\frac{\partial L_{G'}}{\partial \theta_D} = \mathrm{E}_{y\sim P_d(y),x'\sim P(x|y)}[P(y|x')P(x')\frac{\partial \log P(x'|y)}{\partial \theta_D}] \quad (15)$$

3.2 Joint Training Algorithm

In our approach, G and D actually act as discriminator systems for each other in a joint training process: the generator G can be improved with the discriminator D in $GAN1$, and then the enhanced G serves as a better discriminator to guide the training of D in $GAN2$. This training process can be iteratively carried out to obtain further improvements because after each iteration both G and D are expected to be improved with adversarial training. To simultaneously optimize these two models, we design a joint training algorithm to learn the parameters (θ_G and θ_D) shared in two GANs of BGAN-NMT ($GAN1$ and $GAN2$).

Algorithm 1: Joint Training Algorithm for BGAN-NMT

Input : Bilingual corpus $T = \{(x,y)\}_{n=1}^{N}$;
Pre-trained source-side language model $P(x)$;
Pre-trained target-side language model $P(y)$;
Output: Well-trained models $P(y|x)$ and $P(x|y)$

```
Pre-train P(y|x) and P(x|y) on T with MLE principle ;
for number of training iterations do
    for k steps do
        Get m samples {(x,y)}_{i=1}^m from T ;
        Generate m samples {(x,y')}_{i=1}^m using
          P(y|x) given source sentences of
          {(x,y)}_{i=1}^m;
        Update the parameter θ_D with Equation 8 ;
        Generate m samples {(x',y)}_{i=1}^m using
          P(x|y) given target sentences of
          {(x,y)}_{i=1}^m;
        Update the parameter θ_G with Equation 14 ;
    end
    Get m samples {(x,y)}_{i=1}^m from T ;
    Generate m samples {(x,y')}_{i=1}^m using P(y|x)
      given source sentences of {(x,y)}_{i=1}^m;
    Update the parameter θ_G with Equation 10 ;
    Generate m samples {(x',y)}_{i=1}^m using P(x|y)
      given target sentences of {(x,y)}_{i=1}^m;
    Update the parameter θ_D with Equation 15 ;
end
```

As shown in Algorithm 1, the whole algorithm is mainly divided into two steps. Firstly, given parallel corpora $T = \{(x,y)\}_{n=1}^{N}$, we pre-train $P(y|x)$ and $P(x|y)$ with MLE principle, while source and target language models $P(x)$ and $P(y)$ are pre-trained with corresponding sentences of bilingual data. Next, based on these pre-trained models, we implement the two player minmax game using an iterative approach, in which, we alternate between k (equals to 5 in our experiments) steps of optimizing all discriminators (D and D') and one step of optimizing all generators (G and G'). The iterative training continues until the performance of a development data set is not increased.

4 Experiments

4.1 Setup

To examine the effectiveness of our proposed approach, we conduct experiments on translation

tasks with two language pairs: German-English (De-En for in short) and Chinese-English (Zh-En for in short). In all experiments, BLEU (Papineni et al., 2002) is adopted as the automatic metric for translation quality evaluation and computed using Moses *multi-bleu.perl* script.

4.1.1 Dataset

For German-English translation task, following previous work (Ranzato et al., 2015; Bahdanau et al., 2016), we select data from German-English machine translation track of IWSLT2014 evaluation tasks, which consists of sentence-aligned subtitles of TED and TEDx talks. We closely follow the pre-processing as described in Ranzato et al. (2015). The training corpus contains 153k sentence pairs with 2.83M English words and 2.68M German words. The validation set comprises of 6,969 sentence pairs taken from the training data, and the test set is a combination of dev2010, dev2012, tst2010, tst2011 and tst2012 with total number of 6,750 sentence pairs.

For Chinese-English translation task, training data consists of a set of LDC datasets[1], which has around 2.6M sentence pairs with 65.1M Chinese words and 67.1M English words respectively. Any sentence longer than 80 words is removed from training data. NIST OpenMT 2006 evaluation set is used as the validation set, and NIST 2005, 2008, 2012 datasets as test sets. We limit the vocabulary to contain up to 50K most frequent words on both source and target sides, and convert remaining words into the `<unk>` token.

4.1.2 Model Architecture

RNNSearch model proposed by Bahdanau et al. (2014) is leveraged to be the translation model, but it should be noted that our BGAN-NMT is independent of the NMT network structure. We use a single layer GRU for encoder and decoder. For Zh-En, the size of word embedding (for both source and target words) is 256 and the size of hidden layer is set to 1024. For De-En, in order to compare with previous work (Ranzato et al., 2015; Bahdanau et al., 2016), the size of word embedding and GRU hidden state are both set to 256. In addition, $P(x)$ and $P(y)$ are designed as a single-layer GRU language model, which is pre-trained to compute the marginal probability of a sentence, and the size of word embedding and GRU hidden state are the same as RNNSearch model.

[1] LDC2002E17, LDC2002E18, LDC2003E07, LDC2003E14, LDC2005E83, LDC2005T06, LDC2005T10, LDC2006E17, LDC2006E26, LDC2006E34, LDC2006E85, LDC2006E92, LDC2006T06, LDC2004T08, LDC2005T10

Methods	Baseline	Model
MIXER (Ranzato et al., 2015)	20.10	21.81
MRT (Shen et al., 2016)	-	25.84
BSO (Wiseman and Rush, 2016)	24.03	26.36
Adversarial-NMT (Wu et al., 2017)	-	27.94
A-C (Bahdanau et al., 2016)	27.56	28.53
Softmax-Q (Ma et al., 2017)	27.66	28.77
Adversarial-NMT*	27.63	28.03
BGAN-NMT	27.63	**29.17**

Table 1: Comparison with previous work on IWSLT2014 German-English translation task. The "Baseline" means the performance of pre-trained model used to warmly start training.

4.1.3 Training Details

For the training of BGAN-NMT, parameters are firstly initialized using a normal distribution with a mean of 0 and a variance of $\sqrt{6/(d_{row}+d_{col})}$, where d_{row} and d_{col} are the number of rows and columns in the structure (Glorot and Bengio, 2010). Then we pre-train NMT and language models with MLE principle to convergence, and select the best model according to the performances on the validation set, where BLEU scores and the perplexity are adopted as evaluation metrics for NMT and language models respectively. Both generator and discriminator models in BGAN-NMT are warmly started with those pre-trained models, and optimized using the vanilla SGD algorithm with mini-batch 32 for De-En and 128 for Zh-En. We re-normalize gradients if their norm exceeds 2.0. The initial learning rate is set as 0.2 for De-En and 0.02 for Zh-En, and it is halved when BLEU scores on the validation set do not increase for 20,000 batches. To generate the synthetic bilingual data, beam search strategy with beam size 4 is adopted for both De-En and Zh-En. At test time, beam search is employed to find the best translation with beam size 8 and translation probabilities normalized by the length of the candidate translations. Follow Luong et al. (2015), `<unk>` is replaced with the corresponding target word in a post processing step.

4.2 Results on German-English Translation

For German-English translation task, in addition to the baseline system which is used to warmly start our BGAN-NMT training, we also include

System	NIST2006	NIST2005	NIST2008	NIST2012	Average
HPSMT	32.46	32.42	25.23	26.20	29.08
RNNSearch	38.61	38.31	30.04	28.48	33.86
Adversarial-NMT*	39.79	38.81	31.86	30.19	35.16
BGAN-NMT	**40.74**	**39.20**	**33.55**	**31.30**	**36.19**

Table 2: Case-insensitive BLEU scores (%) on Chinese-English translation. The "Average" denotes the average results of all datasets.

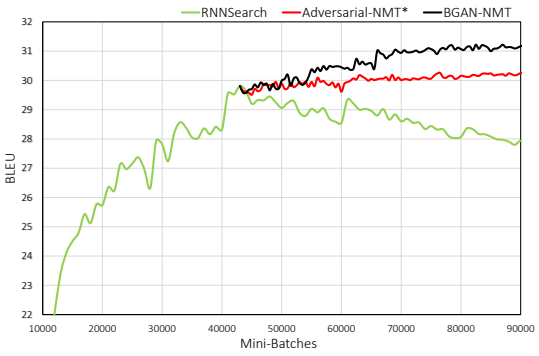


Figure 2: The BLEU score changes on IWSLT2014 German-English validation set for RNNSearch, Adversarial-NMT* and BGAN-NMT as training progresses.

results of other existing NMT systems, including MIXER (Ranzato et al., 2015), MRT (Shen et al., 2016)[2], BSO (Wiseman and Rush, 2016), Adversarial-NMT (Wu et al., 2017), A-C (Bahdanau et al., 2016) and Softmax-Q (Ma et al., 2017). Besides, following Wu et al. (2017), we also implement Adversarial-NMT* system which combines adversarial training objective with MLE. All the results are reported based on case-sensitive BLEU.

From Table 1, we can see that our BGAN-NMT achieves significant improvements over the baseline RNNSearch system. It demonstrates that GAN framework can alleviate exposure bias problem and improve the robustness of NMT systems. Our BGAN-NMT also obtains satisfactory translation quality against other existing NMT systems. In particular, our BGAN-NMT outperforms Adversarial-NMT* by 1.14 BLEU points. Adversarial-NMT* adopts MLE to stabilize the training of generator but gains limited improvement due to the inadequate training problem of the discriminator, while our BGAN-NMT can effectively handle this issue and obtain significant improvement.

[2]The result of MRT method is taken from Wu et al. (2017)

To better analyze training process of the different methods, we compare the BLEU score changes on IWSLT2014 German-English validation set for RNNSearch, Adversarial-NMT* and BGAN-NMT during the entire training. As illustrated in Figure 2, initialized with the best RNNSearch model, Adversarial-NMT* and BGAN-NMT can continually improve the translation performance. In addition, our BGAN-NMT steadily performs much better than Adversarial-NMT* in the whole training process. It confirms that our proposed approach not only stabilizes GAN training but also achieves better results.

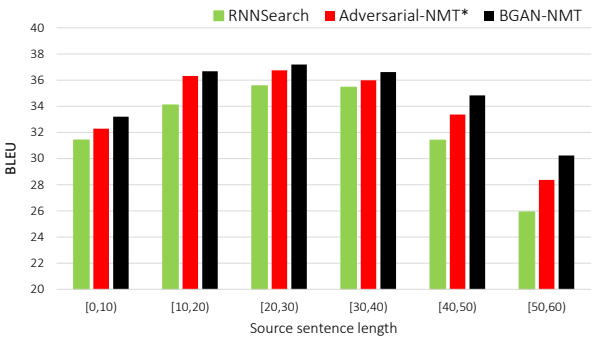
4.3 Results on Chinese-English Translation

We also conduct experiments on Chinese-English translation task with strong SMT and NMT baselines: HPSMT, RNNSearch and Adversarial-NMT*. HPSMT is an in-house implementation of the hierarchical phrase-based MT system (Chiang, 2007), where a 4-gram language model is trained using the modified Kneser-Ney smoothing algorithm over the target data from bilingual data.

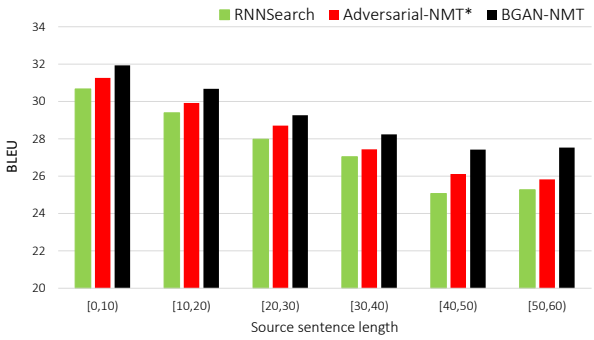
Table 2 shows the evaluation results of different models on NIST datasets. All the results are reported based on case-insensitive BLEU. We can observe that RNNSearch significantly outperforms HPSMT by 4.78 BLEU points on average, and BGAN-NMT can further improve the performances, with 2.33 BLEU points on average. Additionally, our BGAN-NMT gains better performances than Adversarial-NMT* with 1.03 BLEU points on average. These experimental results confirm the effectiveness of our proposed approach, similar as shown in the German-English translation task.

4.4 Effect on Long Sentences

Longer source sentence implies longer translation that more easily suffers from exposure bias problem. In this subsection, we group source sentence of similar length together and calculate the



(a) German-English Translation



(b) Chinese-English Translation

Figure 3: Performance of the generated translations with respect to the length of source sentences on different datasets. For Chinese-English, we merge all NIST datasets in this experiment. For German-English, we only use test datasets.

BLEU score for each group. As shown in Figure 3, we can view that our approach outperforms RNNSearch and Adversarial-NMT* in all length segments, especially achieving notable improvements on long sentences. These results further demonstrate that our approach can better handle this problem and yield higher quality translations.

4.5 Effect of Discriminative Loss

We also perform an ablation experiment in order to quantify the effect of the discriminative loss on our models. As shown in Table 3, the discriminative loss can bring 0.58 and 0.73 BLEU score improvements on English-German and Chinese-English dataset respectively. This result proves that the discriminative loss can improve the discriminative ability of bidirectional NMT models, which can give more accurate rewards for the generator training in GAN framework.

Model	DE-EN	ZH-EN
BGAN-NMT	29.17	36.19
-Discriminative Loss	28.59	35.46

Table 3: Translation performance of BGAN-NMT without discriminative loss on German-English (DE-EN) and Chinese-English (ZH-EN) translations. The BLEU score for Chinese-English translation is the average results of all datasets we used in the experiment.

5 Related Work

As a new paradigm of machine translation, NMT typically suffers from the exposure bias problem due to MLE training. To handle this issue, many methods have been proposed, including designing new training objectives (Shen et al., 2016; Wiseman and Rush, 2016) and adopting reinforcement learning approaches (Ranzato et al., 2015; Bahdanau et al., 2016). Shen et al. (2016) proposed to directly minimize expected loss (maximize the expected BLEU) with Minimum Risk Training (MRT). Wiseman and Rush (2016) adopted a beam-search optimization algorithm to reduce inconsistency between training and inference. Besides, Ranzato et al. (2015) proposed a mixture training method to perform a gradual transition from maximum likelihood learning into optimizing BLEU scores using reinforcement algorithm. Bahdanau et al. (2016) designed an actor-critic algorithm for sequence prediction, in which the NMT system is the actor, and a critic network is proposed to predict the value of output tokens. Recently, Yang et al. (2017) and Wu et al. (2017) proposed to leverage GAN framework to deal with the exposure bias problem, in which NMT model is employed as the generator, and CNN-based or RNN-based model is used as the discriminator. Different from their work, both generator and discriminator in our approach are designed to model the joint probability of sentence pairs and then we design an auxiliary GAN to take advantage of the symmetry of them.

Another similar research in NMT is to leverage bidirectional dependency to improve translation quality. Tu et al. (2017) designed a re-constructor module for NMT in order to make the target representation contain the complete source information which can reconstruct back to the source sentence. Cheng et al. (2016) and He et al. (2016) proposed to reconstruct monolingual data by auto-encoder,

in which bidirectional translation models form a closed loop and are jointly updated. Recently, this similar idea is used in unsupervised machine translation tasks(Artetxe et al., 2017; Lample et al., 2018).

6 Conclusion

In this paper, we have presented a Bidirectional Generative Adversarial Network for Neural Machine Translation, consisting of an original GAN and an auxiliary GAN. Both generator and discriminator in original GAN are designed to model the joint probability of sentence pairs. Auxiliary GAN adopts generator and discriminator models of original one but exchanges their roles to full utilize the symmetry of them. Then these two GANs are alternately updated using joint training algorithm. Experimental results on German-English and Chinese-English translation tasks demonstrate that our proposed approach not only stabilizes GAN training but also leads to significant improvements. In the future, we plan to extend this method to other sequence-to-sequence NLP tasks.

Acknowledgments

We appreciate Dongdong Zhang, Shuangzhi Wu, Wenhu Chen and Guanlin Li for the fruitful discussions. We also thank the anonymous reviewers for their careful reading of our paper and insightful comments.

References

Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *CoRR*, abs/1701.07875.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *CoRR*, abs/1710.11041.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *CoRR*, abs/1607.07086.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*.

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

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33:201–228.

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

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

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*.

Ian J. Goodfellow. 2016. Nips 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160.

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

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

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

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

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

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

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *CoRR*, abs/1804.07755.

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

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

Xuezhe Ma, Pengcheng Yin, Jingzhou Liu, Graham Neubig, and Eduard H. Hovy. 2017. Softmax q-distribution estimation for structured prediction: A theoretical interpretation for raml. *CoRR*, abs/1705.07136.

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

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

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *CoRR*, abs/1511.06732.

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

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

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction. In *AAAI*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

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

Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jian-Huang Lai, and Tie-Yan Liu. 2017. Adversarial neural machine translation. *CoRR*, abs/1704.06933.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2017. Improving neural machine translation with conditional sequence generative adversarial nets. *CoRR*, abs/1703.04887.

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