Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 4680–4688
Marseille, 11–16 May 2020

Best Student Forcing: A Simple Training Mechanism in Adversarial Language Generation

Jonathan Sauder*, Ting Hu*, Xiaoyin Che, Gonçalo Mordido, Haojin Yang, Christoph Meinel
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany
jonathan.sauder@student.hpi.de
{ting.hu, xiaoyin.che, goncalo.mordido, haojin.yang, meinel.christoph}@hpi.de

Abstract

Language models trained with Maximum Likelihood Estimation (MLE) have been considered as a mainstream solution in Natural Language Generation (NLG) for years. Recently, various approaches with Generative Adversarial Nets (GANs) have also been proposed. While offering exciting new prospects, GANs in NLG by far are nevertheless reportedly suffering from training instability and mode collapse, and therefore outperformed by conventional MLE models. In this work, we propose techniques for improving GANs in NLG, namely Best Student Forcing (BSF), a novel yet simple adversarial training mechanism in which generated sequences of high quality are selected as temporary ground-truth to further train the generator. We also use an ensemble of discriminators to increase training stability and sample diversity. Evaluation shows that the combination of BSF and multiple discriminators consistently performs better than previous GAN approaches over various metrics, and outperforms a baseline MLE in terms of Fréchét Distance, a recently proposed metric capturing both sample quality and diversity.

Keywords: natural language generation, generative adversarial nets, reinforcement learning

1. Introduction

***N**atural **L**anguage **G**eneration* (NLG) is a critical sub-task in many natural language processing tasks including machine translation, image captioning, and conversational systems. However, the generation of sequences that are semantically coherent and grammatically correct is difficult. Neural networks trained with ***M**aximum **L**ikelihood **E**stimation* (MLE) over Ground-Truth (GT) sentences have shown impressive results on many of these tasks (Karpathy and Fei-Fei, 2015; Wu et al., 2016; Hu et al., 2017), and been considered as the mainstream solution in NLG.

Recently, generative models of ***V**ariational **A**uto**E**ncoder* (VAE) are reportedly achieving inspiring results over MLE (Schmidt, 2019). In VAE, a pair of encoder-decoder are trained simultaneously, while applies a latent loss typically based on KL-divergence of the encoded latent vector from prior Gaussian distribution, in addition to reconstruction loss calculated by MLE. After training, the decoder will do the generation job alone. Apparently, VAE models should also be trained only with GT sentences.

Meanwhile, ***G**enerative **A**dversarial **N**ets* (GANs) (Goodfellow et al., 2014) have been introduced into NLG (Yu et al., 2017; Che et al., 2017; Guo et al., 2017; Lin et al., 2017), bringing in new possibility to train the model further with guiding signals not directly from GT. However, various reports indicate that language GANs have shown shortcomings in terms of training stability and sample diversity, which make them less competitive in performance with conventional language models trained with MLE (Caccia et al., 2018; Semenuita et al., 2018; Tevet et al., 2018; Zhu et al., 2018), not to mention VAE. This fact motivates us to work for improvement.

We propose ***B**est **S**tudent **F**orcing* (BSF) for adversarial training for NLG, which uses generated sequences that the discriminator perceives as being of high quality to further train the generator in an MLE-like manner. This can be interpreted as "*forcing*" the generative model to learn from the "*best student*". Theoretically, BSF could take any type of non-adversarial NLG model as the generator, including the decoder of VAE. We also introduce a dynamic ensemble of multiple discriminators to alleviate mode collapse, a common problem encountered by GANs, and increase sample diversity. Experiments that compare various GAN approaches for language generation and a baseline MLE model shows that BSF leads to significant improvements over previous GAN approaches. In particular, our approach in a multi-discriminator setting outperforms the baseline language model trained with MLE over a recently proposed metric, *Fréchét Distance*, that captures both sample quality and diversity (Semenuita et al., 2018).

The rest of this paper is organized as follows: Section 2 discusses related work, Section 3 and Section 4 introduce two technical proposals, Best Student Forcing and dynamic ensembles of discriminators. This is followed by the experiment setup, results and their analysis, and conclusion. We highlight our main contributions as:

- A novel, simple, versatile and efficient adversarial training method, Best Student Forcing, for discrete sequence generation.
- The introduction of a dynamic ensemble of discriminators in GANs for language generation, reducing mode collapse and increasing training stability.
- A detailed evaluation with both traditional and recently proposed metrics which proves the capability of above two and their combination.

*Denotes equal contribution

2. Related Work

The goal of NLG is to produce sequences of tokens $x_0, x_1, \dots x_t$ which form syntactically correct and semantically coherent sentences. Currently, many top-performing models are RNN language models (Mikolov et al., 2010), which are typically trained in a supervised fashion using MLE (also known as *teacher forcing*) (Williams and Zipser, 1989). During MLE training, a θ-parametrized RNN is trained to approximate $P(x_t|x_0, x_1 \dots x_{t-1})$ by $\hat{P}(x_t|x_0, x_1 \dots x_{t-1}, \theta)$, by minimizing the multi-label cross-entropy via the objective function:

$$J_\theta(x) = -\sum_{t=1}^{T} \log \hat{P}(x_t|x_0, x_1 \dots x_{t-1}, \theta) \qquad (1)$$

However, MLE training is reported to be flawed due to *exposure bias*, which arises from the model only seeing ground-truth data during the training phase (*teacher-forcing* mode) and therefore potentially misbehaving when being fed sequence prefixes sampled from its own distribution during the inference phase (*free-running* mode) (Lamb et al., 2016; Ranzato et al., 2015). In order to mitigate exposure bias, a method called Professor Forcing (Lamb et al., 2016) proposes regularizing the difference between hidden states after encoding real and generated samples during training, while Scheduled Sampling (Bengio et al., 2015) applies a mixture of teacher-forcing and free-running mode with a partially random scheme. However, Scheduled Sampling has been shown to be inconsistent (Huszár, 2015).

Variational Auto Encoder (VAE) is one form of generative model, proposed by Kingma and Welling (2013). The VAE model consists of a ϕ-parametrized encoder and a θ-parametrized decoder. The whole model works by maximizing the marginal log-probability $\log p_\theta(x)$, which can be achieved by maximizing its lower bound:

$$L = E_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p_\theta(z)) \qquad (2)$$

where the first term is the reconstruction loss, computed by MLE, and the KL divergence term works as a regularizer.

Recently, another class of generative models, GAN approaches, has been introduced into NLG. GANs (Goodfellow et al., 2014) typically consist of a θ-parametrized generator network G_θ and a ϕ-parametrized discriminator network D_ϕ, where D_ϕ is trained to distinguish whether a sample comes from G_θ or from the ground-truth, while G_θ is trained to maximize the discriminator's perceived realness, thus "fooling" D_ϕ. Together, their interaction can be expressed as a minimax game: $\min_\theta \max_\phi \mathbb{E}_{x\sim p_{\text{data}}}[\log(D_\phi(x))] + \mathbb{E}_{x\sim G_\theta}[\log(1 - D_\phi(x))]$.

In NLG tasks, GANs are particularly difficult to train since the output of G_θ is discrete and non-differentiable. Among many approaches to overcome this, SeqGAN (Yu et al., 2017) has drawn a lot of attention due to successfully applying the REINFORCE (Policy Gradient) algorithm (Sutton et al., 2000). From this perspective, NLG is interpreted as a sequential decision-making process, where sequence prefix $x_0, x_1 \dots x_{t-1}$ is the state at time step t, and the next token x_t is the action to be selected from the action space of the whole vocabulary, and the reward is based on the discriminator's perceived realness of the generated sequence − full sequence D's score is directly taken as reward for the last time step, while average D's score over sequences generated with certain prefix in a Monte Carlo roll-outs operation is taken as the reward for relevant intermediate time steps.

Despite the promising result achieved by SeqGAN on traditional metrics, the above reward estimation method has drawbacks. Firstly, sequences that are clearly recognized as fake by the discriminator still receive non-negligible positive rewards, pushing the generator to learn from noise. On the other hand, as the discriminator learns to fit the training data very strongly throughout adversarial training, even sequences of relatively high quality receive small rewards, making the generator unable to learn effectively from such "vanishing" signals (Che et al., 2017). Attempts to alleviate the vanishing rewards include RankGAN (Lin et al., 2017), which changes the discriminator's objective into a ranking loss, and MaliGAN (Che et al., 2017), which changes the objective of the generator to a normalized maximum likelihood optimization target. LeakGAN (Guo et al., 2017) attempts to further improve results by using a hierarchical RL architecture and "leaking" features from the discriminator to the generator during generation. In recently proposed ARAML (Ke et al., 2019), the generator is updated by samples acquired from a stationary distribution in a weighted MLE manner. Whereas the way to construct stationary distribution is very complicated.

Besides, Zhang et al. (2019) propose to select oracle sentences in high BLEU scores to train Neural Machine Translation(NMT) system, and report encouraging results (mainly evaluated by BLEU, too). Their methodology is somehow similar to ours, however, using BLEU score to measure the quality of generated sentences is perhaps less suitable in scenario of unconditional NLG than in conditional NMT.

Another focus of previous research is how sequence generation should be properly evaluated. As human evaluation is unfeasible for large amounts of data, the most popular automatic metric used in recent years is n-gram based BLEU (Papineni et al., 2002) or ROUGE (Lin, 2004). However, as these metrics do not capture sample diversity, self-BLEU is introduced (Zhu et al., 2018), and later a *Boltzmann temperature sweep* (Caccia et al., 2018) is further proposed to observe the dynamic balance between BLEU and self-BLEU. Meanwhile, n-gram free metric *Fréchét Distance* (Semenuita et al., 2018) is also proposed. On these new metrics, previous GAN approaches are widely reported as outperformed by a benchmark RNN trained by MLE (Caccia et al., 2018; Semenuita et al., 2018; Tevet et al., 2018; Zhu et al., 2018), although most of them build on MLE pretraining. This fact stirs us to seek improvement.

On the other hand, almost all GAN approaches, not only in NLG, suffer from mode collapse (Goodfellow, 2017), in which the generator learns to cover just a small subset of the original distribution, effectively only producing samples with very low diversity. Many efforts have been made to tackle this phenomenon, but generally require significant modification to the model architecture or training objective (Che et al., 2016; Arjovsky et al., 2017; Arjovsky and Bottou, 2017). However, a recently proposed simple ap-

Algorithm 1: Best Student Forcing (with a single discriminator)

Initialize G_θ, D_ϕ
Pre-train G_θ on real samples
Generate negative samples using G_θ for training D_ϕ
Pre-train D_ϕ via minimizing binary cross entropy
for *adversarial epochs* **do**
  for *generator iterations* **do**
    Generate M full sequences $x^{(1)} \dots x^{(M)} \sim G_\theta$
    Select "Best Student": $x^* := \text{argmax}\{D_\phi(x^{(m)}) \mid 1 \leq m \leq M\}$
    Train G_θ by minimizing $J_\theta(x^*) = -\sum_{t=1}^{T} \log \hat{P}(x_t^* | x_0^*, x_1^* \dots x_{t-1}^*, \theta)$
  end
  for *discriminator iterations* **do**
    Use current G_θ to generate negative samples and combine with real samples
    Train D_ϕ by minimizing binary cross entropy
  end
end

proach, Dropout-GAN (Mordido et al., 2018), suggests using multiple discriminators in GAN training and randomly hiding some of them, as in the well-known dropout operation for neural network weights (Srivastava et al., 2014). Many GAN approaches, including GANs for language generation, can easily be extended with Dropout-GAN.

3. Best Student Forcing

In this paper, we propose a novel training method for GANs for discrete sequence generation, namely ***B**est **S**tudent **F**orcing* (BSF). The basic idea of BSF can be described as follows: we use the discriminator's perceived realness to identify sequences with particularly high quality, using these as temporary "pseudo" ground-truth. The generator is then trained with an MLE-like mechanism on these selected sequences.

Specifically, once a batch of complete sequences is drawn from the generator, the one which best "fools" the discriminator by achieving the highest D's score will be selected. We then use this "best" sequence just as a ground-truth sample and minimize the multi-label cross entropy between the generator's distribution and this sample. The generator is thus updating in a teacher-forcing manner but against "pseudo" ground-truth sequences generated in free-running mode instead of real data. All remaining sequences, which are not of the highest D's score, are simply ignored.

By only updating with the "best student", BSF ensures that sequences of lower quality receive no reward, preventing the generator from essentially learning from noise. At the same time, a strong training signal remains even as the discriminator learns to distinguish samples more clearly, avoiding the vanishing rewards problem. However, we are fully aware that it is still possible that all sequences in a batch are of comparatively low quality, and then BSF would have to learn from the "least bad" option, which is not optimal. Therefore, We recommend pre-training G_θ with MLE as in previous GANs (Yu et al., 2017; Lin et al., 2017; Guo et al., 2017) to allow the generator to produce decent results when starting adversarial training. The full procedure of BSF training is described in Algorithm 1.

From another perspective, BSF can be considered as extending the training set by adding "pseudo" ground-truth picked by discriminator, which would be compatible with any structural update of the generator, such as potential replacement of RNN with *Transformer* (Vaswani et al., 2017), or the decoder in VAE. Compared with previous GAN approaches, we also consider BSF as a light-weight approach. BSF does not add any extra training component and works with a typical GAN structure, while requiring significantly fewer discriminator evaluations during training, as complete sequences are evaluated only once, instead of needing to evaluate many roll-outs. This makes BSF computationally efficient and easy to implement.

4. Dynamic Ensemble of Discriminators

In adversarial training, the quality of the feedback provided by the discriminator is a requirement for successful learning. In our use case, this implies the scalar reward attributed by D_ϕ to a fake sample must be a good indicator of sample quality for G_θ to be able to produce realistic-looking samples. With a single discriminator, the discriminator may "bias" on a certain pattern of sequences generated. Thus, we propose to use an ensemble of different discriminators and guide G_θ by averaging the scores of multiple discriminators at the end of each batch, alleviating such kind of "bias", just as a paper would be better reviewed by multiple reviewers rather than one.

In discriminator training iterations, different discriminators in the ensemble are fed with different batches of both real and fake samples for updating. It is expected that learning from different samples would avoid homogeneous behaviours among discriminators. In this work, the batches are drawn randomly to make things simple, however, a more strategical selection scheme might be introduced in future, such as distributing sequences with different length into different discriminator, to compensate so-called "long sentence punishment" − since a shorter sentence is naturally less error-prone.

Moreover, to further tackle the well-known mode collapse problem in language GANs (Semenuita et al., 2018; Zhu

$$\min_{\theta} \max_{\phi} \frac{\sum_{i=k}^{K} \delta_k (\mathbb{E}_{x \sim p_{\text{data}}}[\log(D_{\phi_k}(x))] + \mathbb{E}_{x \sim G_\theta(\cdot)}[\log(1 - D_{\phi_k}(x))])}{\sum_{i=k}^{K} \delta_k} \tag{3}$$

et al., 2018), we adopt the methodologies introduced in Dropout-GAN (Mordido et al., 2018) and discard the D's score of a given discriminator with a probability d, or *dropout rate*, leading to minimax game of adversarial training as shown in Eq. 3. This makes the ensemble of discriminators "dynamic" at the end of every batch, that G_θ has to please different discriminator sub-groups and minimize its loss, ultimately making G_θ more general and less prone to mode collapse. To the best of our knowledge, we are the first to apply such techniques to the natural language generation setting.

5. Experimental Setup

5.1. Dataset

In this work, we evaluate the ability of various models to match the distribution of a text corpus. We perform all our experiments on the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015), consisting of pairs of sequences with a label representing certain semantic attributes. We ignore these labels and keep all distinct sequences with the 5000 most common words in the dataset, resulting in 500k sequences.

5.2. Models

We systematically compare BSF to SeqGAN (Yu et al., 2017), RankGAN (Lin et al., 2017), and a conventional language model trained with MLE. For fairness, all generator models consist of a single LSTM layer (Hochreiter and Schmidhuber, 1997) with hidden and encoding/decoding units of size 256. Following SeqGAN (Yu et al., 2017), we use a convolutional neural network (CNN) as described by Zhang and LeCun (2015) with an added highway architecture (Srivastava et al., 2015) as a discriminator. We compare the performance of a single-discriminator approach to using an ensemble of discriminators. We also evaluate LeakGAN (Guo et al., 2017), but only evaluate its final output, as the training process is quite different from others due to LeakGAN's special model structure. We also only test LeakGAN in a single discriminator setting, as current implementations (Guo et al., 2017; Zhu et al., 2018) are too memory-intensive to run multiple discriminators.

To verify the universality of BSF, we also apply BSF to update the generator pre-trained by VAE loss. The VAE model consists of one encoder and one decoder, both of which are single LSTM layer, with hidden states of size 256 and latent vectors of size 64. We use the VAE text generation tool provided by Hu et al. (2019), in which KL annealing and word dropout techniques have been applied. After pre-training, we use the decoder of VAE as the generator and an ensemble of discriminators to discriminate sentences.

5.3. Metrics

We intend to keep completeness and consistency with previous works by calculating BLEU scores between 10,000 generated samples and ground-truth for quality evaluation, along with self-BLEU scores, which was introduced to measure model collapse in terms of repeated n-grams within generated samples themselves. As another measure for sample diversity, we also show the absolute number of unique 4-grams. Furthermore, we also evaluate BLEU and self-BLEU scores under a recently proposed *Boltzmann temperature sweep* (Caccia et al., 2018). Please note that all above-mentioned metrics are actually based on n-grams. However, only using n-gram based metrics is challenged by Semenuita et al. (2018) as "insufficient". Alternately, Semenuita et al. (2018) proposed ***Fréchét Distance*** (FD) and claimed that FD is very well-correlated with human judgment of sample quality while also capturing mode collapse for language GANs. FD is actually a generalization of the Fréchét Inception Distance (FID) (Heusel et al., 2017), a widely accepted metric for GAN performance in computer vision research. By using an independent model for extracting features, FD measures the distance between the distributions of features extracted from real and generated data. Following Semenuita et al. (2018), we use the publicly available pre-trained InferSent model v2 (Conneau et al., 2017) as the feature extractor. The feature distribution distance is calculated by:

$$FD(r, g) = \|\mu_r - \mu_g\|_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{0.5}) \tag{4}$$

where μ_r and μ_g denote the mean features of real and generated samples respectively, while Σ_r and Σ_g denote the corresponding covariance matrices of the features. We calculate FD on 10,000 generated samples and 10,000 real samples. In this work, we include FD as one of the major evaluation metrics.

5.4. Training Details

All considered GAN approaches to sequence generation rely on pre-training the generator with MLE such that it is possible to draw reasonable sequences from the generator's distribution. Otherwise, it would be a daunting task to produce a sequence that can not immediately be clearly distinguished as fake by a discriminator. In our experiment, the generator is pre-trained using MLE until convergence. We then switch to adversarial training for the GAN approaches and train for an additional 400 epochs. When employing BSF to the decoder of VAE, the VAE model is pre-trained by its loss function until convergence. Then the decoder is used as the generator, to which BSF adversarial training is applied for 40 epochs.

Besides model architecture, size, and learning rate, which are kept constant across all evaluated models, the remaining hyperparameters are the number of roll-outs, the initial discriminator strength (the number of discriminator pre-training epochs), and the number of discriminator iterations per adversarial epoch. We performed a grid search with 100 trials per model over these hyperparameters, and then ran the best configuration per model seven times to obtain the results presented in this work.

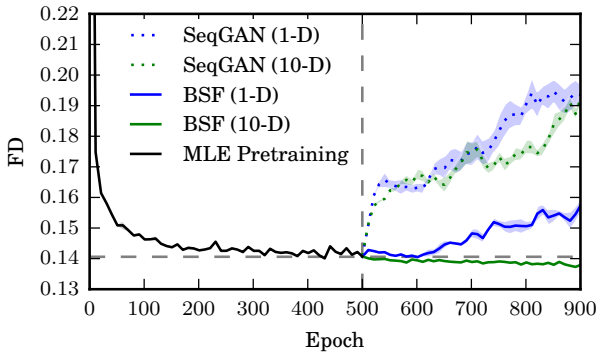

Figure 1: FD comparison throughout training, showing mean and standard deviation over 7 runs. The dashed lines represent the end of pre-training and corresponding mean FD. BSF with multiple discriminators shows the best stability and keeps improving after pre-training.

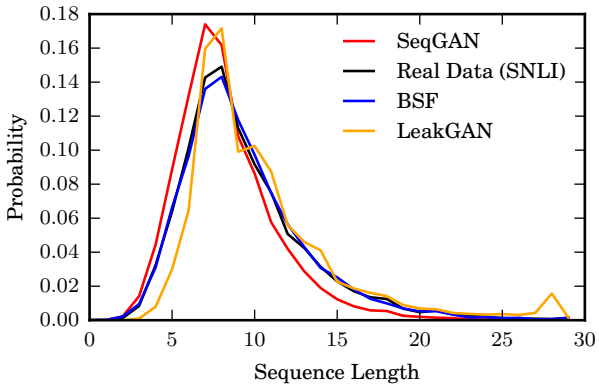

Figure 3: Figure showing probabilities of sentence lengths occurring, as approximated by 10,000 samples.

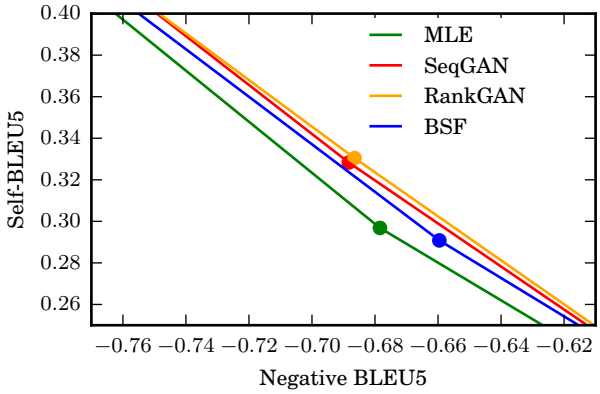

Figure 4: Boltzmann temperature sweeping of different models with temperature parameters $\alpha \in [0.7, 1.3]$. Lower is better for both axis.

6. Experimental Result and Analysis

In this section, we would like to present the experimental result and then make some analysis for discussion. Fig. 1 displays FD values throughout the training process of SeqGAN and our proposed BSF on SNLI dataset, with MLE pre-training as a baseline. The curves in Fig. 1 clearly show that SeqGAN immediately performs worse according to FD after switching to adversarial training, while BSF shows stability and can further improve FD over MLE in a multi-discriminator setting.

Highlighting the general effect of using multiple discriminators in adversarial NLG, Fig. 2 presents the standard deviations and the average FDs over 7 runs of BSF, SeqGAN and RankGAN with varying numbers of discriminators. For all models, adding more discriminators (from 1 to 15) shows positive effects on training stability (lower standard deviation) and sample diversity (generally lower FD), while BSF benefits the most.

Table 1 illustrates performances over more metrics based on 10,000 samples generated at the end of each model's training. BSF (with 10 discriminators) exhibits the best FD, slightly but statistically significantly ($p < 0.0005$) outperforming baseline MLE and way better than other GAN approaches. While SeqGAN and LeakGAN show higher BLEU scores, just same as what others reported (Semenuita et al., 2018; Zhu et al., 2018), but their smaller numbers of unique 4-grams suggest that the high BLEU could probably attribute to a small variety of samples generated, so do their higher Self-BLEU scores. Meanwhile, proposed BSF achieves lowest Self-BLEU and the highest number of unique 4-grams, which would also suggest better sample diversity.

Previous work has observed that SeqGAN does not match the target distribution in terms of sequence length, collapsing onto short and simple sentences (Zhu et al., 2018; Semenuita et al., 2018). Matching the distribution of sequence length is another potential indicator of how capable a model is to fit the training data. Fig. 3 shows the estimated distribution of sequence lengths from different models. It clearly shows that BSF matches the original distribution (SNLI) closely than other GAN approaches.

Meanwhile, a *Boltzmann temperature sweep* is proposed to evaluate a language model over whole quality-diversity space. According to Fig. 4, MLE performs the best over a temperature sweep, while BSF is better than other GAN approaches. The circles in Fig. 4 indicate where temperature parameter $\alpha = 1$, where the samples are actually generated.

For the generator pre-trained by VAE loss, its generation result comparison before and after adversarial training is listed in Table 2. We can see that both BLEU and Self-BLEU score increase after BSF training, indicating higher quality but less diversity in generated sentences. The larger FD also shows the same tendency. This is explainable, since the latent space after training is close to standard Gaussian distribution, and BSF training intensifies latent vectors corresponding to higher-quality sentence, resulting in less diversity. It will be interesting for future work to figure out how to use BSF to push the whole latent space closer to standard Gaussian, rather than some local areas.

In terms of human evaluation, only a small subset of actual samples can be presented here. Table 3 shows generated samples containing the verb "throw(s)". BSF seems to achieve better generation performance than others, taking into consideration grammar, semantics and diversity. For

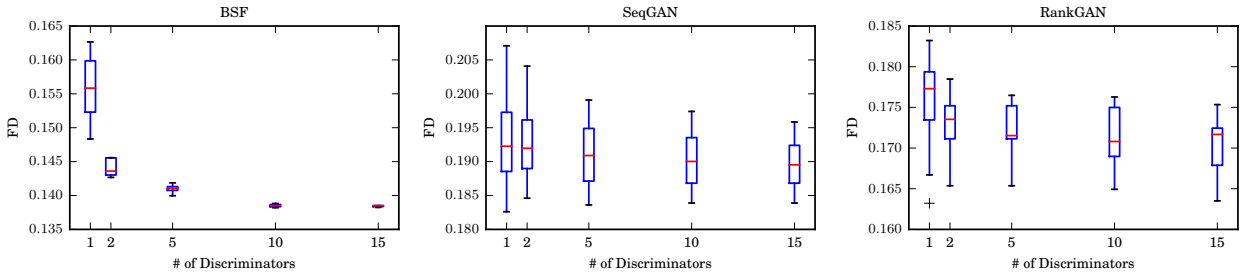

Figure 2: The effects of using multiple discriminators when training with BSF (left), SeqGAN (middle) and RankGAN (right). Using multiple discriminators leads to better performance (lower FD) and more training stability (lower variance).

	MLE	SeqGAN-10D	RankGAN-10D	LeakGAN-1D	BSF-10D
FD ↓	0.141 ± 0.001	0.192 ± 0.007	0.180 ± 0.008	0.572 ± 0.112	**0.138 ± 0.001**
4-Grams ↑	53.6k ± 0.95k	44.2k ± 1.49k	48.2k ± 1.42k	24.9k ± 2.21k	**53.9k ± 0.38k**
BLEU3 ↑	0.922 ± 0.002	0.925 ± 0.004	0.919 ± 0.004	**0.932 ± 0.034**	0.914 ± 0.001
BLEU4 ↑	0.819 ± 0.003	0.826 ± 0.006	0.811 ± 0.007	**0.832 ± 0.051**	0.805 ± 0.002
BLEU5 ↑	0.687 ± 0.004	0.699 ± 0.007	0.675 ± 0.009	**0.741 ± 0.071**	0.671 ± 0.002
Self-BLEU3 ↓	0.663 ± 0.005	0.690 ± 0.007	0.678 ± 0.009	0.699 ± 0.030	**0.654 ± 0.005**
Self-BLEU4 ↓	0.472 ± 0.004	0.502 ± 0.010	0.483 ± 0.010	0.555 ± 0.023	**0.463 ± 0.007**
Self-BLEU5 ↓	0.305 ± 0.002	0.333 ± 0.009	0.313 ± 0.009	0.447 ± 0.041	**0.299 ± 0.007**

Table 1: Metric comparison for different models, showing mean and standard deviation over seven runs. ↑ means higher is better, ↓ means lower is better.

	VAE	VAE+BSF
FD ↓	**0.144 ± 0.001**	0.150 ± 0.006
4-Grams ↑	**53.2k ± 0.02k**	51.6k ± 0.19k
BLEU3 ↑	0.906 ± 0.001	**0.912 ± 0.006**
BLEU4 ↑	0.775 ± 0.001	**0.788 ± 0.003**
BLEU5 ↑	0.613 ± 0.002	**0.632 ± 0.020**
Self-BLEU3 ↓	**0.657 ± 0.002**	0.660 ± 0.007
Self-BLEU4 ↓	**0.445 ± 0.003**	0.451 ± 0.008
Self-BLEU5 ↓	**0.292 ± 0.003**	0.296 ± 0.006

Table 2: Metric comparison for different models. ↑ means higher is better, ↓ means lower is better.

example, the last sentence generated by MLE and the first sentence generated by VAE are apparently wrong in semantics but correct in grammar. And BSF tends to generate more diverse phrases related to "throw(s) a ball". Nevertheless, we are crystal clear that only a few dozens samples are far from enough to sufficiently represent the whole set, so we make all generated samples from our experiments available for public evaluation [1]

Besides the generally positive outcomes, we also encountered some problems during BSF set up and want to present them here for discussion. For example, even with a high number of discriminators, the D's score still cannot reliably indicate the quality of sentence generated. This is reflected by selecting "best student" from a large number of candidates (e.g. 64) resulting in worse performance than using a smaller number. Practically, we found 16 as the optimal option in our case. Also, there is no clear standard that how can we define a "best student", perhaps setting an absolute threshold on D's score might also be applicable if the discriminators are considered as generally trustworthy. Moreover, we also tested saving "best students" as a part of fake samples to train discriminators in next epoch, but without getting an improvement. All these facts suggest that there is still quite a lot to do in future.

[1] 10,000 samples from each model involved in evaluation are provided in the following link: `https://drive.google.com/drive/folders/1bVuerqXi69o8UGX1BV0AtnFlB3CdVz3Z`, will be online together with source code upon publication.

7. Conclusion

In this work, we focus on improving GANs for language generation. We tackle the problems of training instability, mode collapse, and sample quality exhibited in previous related work by proposing Best Student Forcing and using multiple discriminators. Evaluation shows that (1) BSF consistently outperforms existing GAN approaches; (2) implementing multiple discriminators generally improves the performances of all language GANs; (3) BSF with a multi-discriminator setting performs better than baseline MLE over recently proposed Fréchét Distance, but still needs to improve over a Boltzmann temperature sweep.

Our future work will first focus on getting a more profound understanding of how the signal from an ensemble of discriminators can be an even more accurate estimation of true sequence quality. We would also attempt with more variants of BSF, especially the token-wise architecture, in order to further improve adversarial training effects on language generation task. On the other hand, we plan to implement human evaluation for samples generated from different models by using some public crowd-sourcing platforms.

MLE	a woman and child **throw** a large box . children **throw** a red ball into a pond under an outside market . a sport player makes a up **throw** as a crowd watches . the woman **throws** the ball . someone is going to the best bar to **throw** a ball at a baseball game the man prepares to **throw** a ball before a large group of people . a boy **throws** rocks into a lake . volleyball players **throw** the ball . a boy wearing white shorts and a blue shirt prepares to **throw** the huge grass .
MLE+BSF	the woman is going to **throw** the stick to his dog for the dog . the best player prepares to **throw** the ball in rugby . a young man stands by a girl who is raised about to **throw** a snow . baseball player in a black uniform is about to **throw** the ball . the child **throws** a football in the sports field . the boy is hitting a **throw** the bowling ball . people using the street , they **throw** boxes into a opposite ways . a man **throws** a basketball . one man **throws** a ball into the ground .
SeqGAN	the girl and man **throw** the matching jacket the man . two woman are doing a weekend **throw** . a boy **throws** a football around during the sunny day . the child **throws** the football . two boys **throw** a ball a boy playing basketball is getting ready to **throw** a basketball . the pitcher is going to **throw** a strike . guy getting ready to **throw** a baseball on a field . a girl is about to **throw** a football .
VAE	a football player is about to **throw** his leg off the wall . a baseball player prepares to **throw** the ball . the player **throws** the hockey ball . a guy in a red sweater **throws** an apple at the railing . a person prepares to **throw** . a man watches another guy **throw** a football . the woman is about to **throw** flowers at the snowboarder a hockey player jumps to **throw** the ball to the player . an older man in a yellow shirt **throw** a stick.
VAE+BSF	a man wearing an orange and red uniform is attempting to **throw** the javelin . the brothers **throw** a ball at the park . the girls are about to **throw** the football to the house a girl is playing about to **throw** something to a car a man in a purple shirt is about to **throw** a bowling ball . a boy **throws** a ball . a woman **throws** a tennis ball . the man **throws** a football at the golf course . young boy in a white t-shirt **throws** a snowball in his mouth .

Table 3: Generated samples that contain the word "throw(s)" (in bold font) among different approaches trained with SNLI dataset, a pattern of sport-like "throw a ball" is highlighted by underline. Only the first 9 samples from each approach are presented here due to space limitation, and samples are presented by the original order as they were generated. "MLE+BSF" indicates generator pre-trained by MLE, "VAE+BSF" denotes generator pre-trained by VAE loss, then both of them trained by BSF with 10 discriminators. The full sample packages are available online, along with the ground-truth package.

8. Bibliographical References

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. *International Conference on Learning Representations (ICLR)*.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223.

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, et al., editors, *Advances in Neural Information Processing Systems 28*, pages 1171–1179. Curran Associates, Inc.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D.

(2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Caccia, M., Caccia, L., Fedus, W., Larochelle, H., Pineau, J., and Charlin, L. (2018). Language gans falling short. *arXiv preprint arXiv:1811.02549*.

Che, T., Li, Y., Jacob, A. P., Bengio, Y., and Li, W. (2016). Mode regularized generative adversarial networks. *CoRR*, abs/1612.02136.

Che, T., Li, Y., Zhang, R., Hjelm, R. D., Li, W., Song, Y., and Bengio, Y. (2017). Maximum-likelihood augmented discrete generative adversarial networks. *CoRR*, abs/1702.07983.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September. Association for Computational Linguistics.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, et al., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Goodfellow, I. J. (2017). NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160.

Guo, J., Lu, S., Cai, H., Zhang, W., Yu, Y., and Wang, J. (2017). Long text generation via adversarial training with leaked information. *CoRR*, abs/1709.08624.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. (2017). Controllable text generation. *CoRR*, abs/1703.00955.

Hu, Z., Shi, H., Tan, B., Wang, W., Yang, Z., Zhao, T., He, J., Qin, L., Wang, D., et al. (2019). Texar: A modularized, versatile, and extensible toolkit for text generation. In *ACL 2019, System Demonstrations*.

Huszár, F. (2015). How (not) to Train your Generative Model: Scheduled Sampling, Likelihood, Adversary? *ArXiv e-prints*, November.

Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Ke, P., Huang, F., Huang, M., and Zhu, X. (2019). Araml: A stable adversarial training framework for text generation. *arXiv preprint arXiv:1908.07195*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Lamb, A. M., GOYAL, A. G. A. P., Zhang, Y., Zhang, S., Courville, A. C., and Bengio, Y. (2016). Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609.

Lin, K., Li, D., He, X., Zhang, Z., and Sun, M.-T. (2017). Adversarial ranking for language generation. In *Advances in Neural Information Processing Systems*, pages 3155–3165.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Mordido, G., Yang, H., and Meinel, C. (2018). Dropout-gan: Learning from a dynamic ensemble of discriminators. *arXiv preprint arXiv:1807.11346*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. (2015). Sequence level training with recurrent neural networks. *CoRR*, abs/1511.06732.

Schmidt, F. (2019). Generalization in generation: A closer look at exposure bias. *arXiv preprint arXiv:1910.00292*.

Semenuita, S., Severyn, A., and Gelly, S. (2018). On accurate evaluation of gans for language generation. *arXiv preprint arXiv:1806.04936*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015). Highway networks. *CoRR*, abs/1505.00387.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.

Tevet, G., Habib, G., Shwartz, V., and Berant, J. (2018). Evaluating text gans as language models. *arXiv preprint arXiv:1810.12686*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858.

Zhang, X. and LeCun, Y. (2015). Text understanding from scratch. *CoRR*, abs/1502.01710.

Zhang, W., Feng, Y., Meng, F., You, D., and Liu, Q. (2019). Bridging the gap between training and inference for neural machine translation. *arXiv preprint arXiv:1906.02448*.

Zhu, Y., Lu, S., Zheng, L., Guo, J., Zhang, W., Wang, J., and Yu, Y. (2018). Texygen: A benchmarking platform for text generation models. *CoRR*, abs/1802.01886.