Diversity Regularized Autoencoders for Text Generation

Hyeseon Ko
Sungkyunkwan University
Republic of Korea
hyeseonko@skku.edu

Junhyuk Lee
Sungkyunkwan University
Republic of Korea
ljhjoon00@skku.edu

Jinhong Kim
Sungkyunkwan University
Republic of Korea
legend7811@skku.edu

Jongwuk Lee*
Sungkyunkwan University
Republic of Korea
jongwuklee@skku.edu

Hyunjung Shim
Yonsei University
Republic of Korea
kateshim@yonsei.ac.kr

ABSTRACT

In this paper, we propose a simple yet powerful text generation model, called *diversity regularized autoencoders (DRAE)*. The key novelty of the proposed model lies in its ability to handle various sentence modifications such as insertions, deletions, substitutions, and maskings, and to take them as input. Because the noise-injection strategy enables an encoder to make the latent distribution smooth and continuous, the proposed model can generate more diverse and coherent sentences. Also, we adopt the Wasserstein generative adversarial networks with a gradient penalty to achieve stable adversarial training of the prior distribution. We evaluate the proposed model using quantitative, qualitative, and human evaluations on two public datasets. Experimental results demonstrate that our model using a noise-injection strategy produces more natural and diverse sentences than several baseline models. Furthermore, it is found that our model shows the synergistic effect of grammar correction and paraphrase generation in an unsupervised way.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; *Natural language generation*; Lexical semantics;

KEYWORDS

Variational autoencoder, Adversarial training, Data augmentation

ACM Reference Format:
Hyeseon Ko, Junhyuk Lee, Jinhong Kim, Jongwuk Lee, and Hyunjung Shim. 2020. Diversity Regularized Autoencoders for Text Generation. In *The 35th ACM/SIGAPP Symposium on Applied Computing (SAC '20), March 30-April 3, 2020, Brno, Czech Republic.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3341105.3373998

*Corresponding author

SAC '20, March 30-April 3, 2020, Brno, Czech Republic

ACM ISBN 978-1-4503-6866-7/20/03…$15.00
https://doi.org/10.1145/3341105.3373998

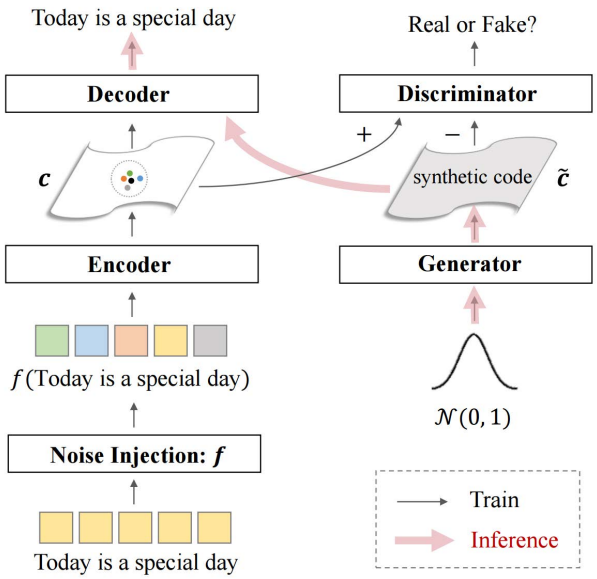

Figure 1: The overall framework of DRAE, where $f(\cdot)$ indicates various noise injection methods for insertions, deletions, substitutions and masking of words.

1 INTRODUCTION

Generative models for text have gained wide attention in various natural language processing (NLP) applications. The most famous examples include machine translation [2, 20], dialogue systems [27, 28], storytelling [18], and text summarization [25]. Typically, text generation models can be categorized into two approaches: (1) maximum likelihood estimation (MLE)-based models [5, 26, 32] that generate a sequence of words in a sentence under an encoder-decoder framework, and (2) generative adversarial network (GAN)-based models [9, 12, 14, 19, 33] that implicitly learn the latent or text distributions, where the generator produces synthetic sentences by adversarial training with the discriminator.

Among the MLE-based models, the variational autoencoder (VAE) [16] represent the promising solutions. It is a tractable generative model that performs variational inference built on the autoencoding architecture. It employs a reparameterization trick of

the variational posterior that guarantees a continuous latent distribution. However, the primary challenge of VAE is the collapse of posterior loss (represented by a Kullback-Leibler (KL) divergence term); this is known as the KL vanishing problem. In other words, because VAE relies on the autoregressive properties of the decoder, it tends to overlook the prior distribution of latent variables computed by the encoder. Another issue of VAE is the lack of a diverse generation of text. Because the parametric formulation of the posterior and prior approximations is typically designed as simple diagonal Gaussian distributions, it is limited to representing complex prior distributions.

Recently, it has been shown that adversarially regularized autoencoder (ARAE) [34] can adequately overcome the limitation of simple prior approximations. Instead of the variational bound or simple model approximation for prior distribution, ARAE learns prior distribution through adversarial training, thereby enabling it to infer more complicated distributions. The adversarial training of ARAE is similar to that of adversarial autoencoder (AAE) [22]. In addition, because ARAE does not enforce a pre-defined prior distribution, it can alleviate the posterior collapse problem by considering an arbitrary prior distribution as a regularizer. However, it still cannot generate diverse sentences; it also experiences difficulty in achieving stable training for the prior distribution.

To overcome these limitations, we propose various *noise-injection strategies*. Figure 1 depicts an overview of the proposed model, called *diversity regularized autoencoders (DRAE)*, which is built on ARAE. The key idea of the proposed model is to adopt various modifications of the input data to make a robust encoder. Given a training sentence, we perform insertions, deletions, substitutions, and masking of words; then, we use them as augmented inputs. Using this simple modification as a data-augmentation rule, we can make the proposed model more robust and generate more diverse sentences. That is, adding some noise to input sentences introduces perturbations in the latent space, thereby making it compact and smooth. Therefore, the target distribution for adversarial training is more advantageous for stable GAN training.

In addition to developing the data augmentation rule for text data, we utilize the Wasserstein distance with gradient penalty (GP) for adversarial training because of the success of Wasserstein GAN-GP (WGAN-GP) [1] in image generation. Although WGAN theoretically converges to an optimal solution for the Dirac distribution, it can exhibit pathological behavior owing to the implementation of weight clipping for approximating the 1-Lipschitz constraint. Therefore, our model utilizes a better training method such as WGAN-GP [11] to achieve stable training of GAN.

The main contributions of this paper are summarized as follows.

- We propose a novel text generation model that produces high-quality and diverse sentences using various data augmentations.
- We demonstrate that our model outperforms several baseline models in terms of both the quality and diversity of text on several benchmark datasets.
- We observe that our model also performs grammar correction and paraphrase generation in an unsupervised manner as our model leads a dense and smooth latent distribution with semantic interpretability.

2 RELATED WORK

The generative model for text has two objectives: the generation of realistic and diverse texts. If the model consistently generates high-quality, but the same sentences repeatedly, it cannot be considered as a good generative model. Toward these goals, text generation techniques can be classified into two pillars: MLE- and GAN-based approaches.

MLE-based Approaches. One of the most successful approaches in text generation tasks is using VAE. Inspired by [16] in the field of computer vision, Bowman et al. [5] successfully adopted VAE in language generation tasks under a recurrent neural network (RNN) encoder-decoder framework. However, the main issue in training VAE in texts is that the KL divergence term becomes small, which is known as *KL vanishing* or *posterior collapse*. It occurs because an autoregressive decoder leaks ground-truth information, enabling it to choose an easier path (*i.e.*, relying only on the inputs of the decoder) to generate sequences. To mitigate this problem, [5] introduced two heuristic techniques such as KL annealing and word dropout. Recently, several studies [26, 32] have suggested employing convolutional decoders instead of RNN decoders. However, the careful tuning of the decoder is also required. Also, some recent studies [8, 30, 31] have exploited different prior distributions instead of Gaussian distribution. However, determining an appropriate prior distribution is nontrivial.

To address this problem, adversarial autoencoders (AAE) [22] adopted an adversarial training for the latent code space to distinguish real from fake code by using a discriminator. Although VAE utilizes KL divergence to match the latent code and prior distribution analytically, AAE regularizes the latent code via adversarial training. Through the adoption of adversarial training instead of a KL term, bypassing the KL vanishing problem is possible. However, the issue of generating diverse texts remains unresolved. Recently, Zhao et al. [34] proposed adversarially regularized autoencoders (ARAE), which learn an arbitrary latent code instead of assuming a fixed prior. It is suitable for representing an arbitrary prior distribution. Inspired by the success of WGAN [1], ARAE adopt the Wasserstein distance for adversarial training. However, recent studies [7, 10] have shown that ARAE still suffers from mode collapse, *i.e.*, it tends to generate some sentences repeatedly.

GAN-based Approaches. Generating texts using text-based GANs has been established as one of the mainstream techniques in text generation. However, a major hurdle for text generation lies in the discrete nature of texts. Since the argmax operator represents the vector as one-hot, it is non-differentiable. Therefore, operating backpropagation makes it difficult to train the generator.

To address the discreteness of texts, considerable research has been conducted in reinforcement learning (RL)-based and continuous approximation methods. RL-based methods require pre-training, which is time-consuming. One example of the RL-based method is SeqGAN [33], which utilizes a rollout policy to receive a reward even when the sentence is incomplete. MaliGAN [6] uses co-training to reduce gradient variance, and RankGAN [19] leverages ranking models instead of the discriminator. LeakGAN [12] provides the generators with intermediate information. MaskGAN [9] employs an actor-critic training procedure that encourages the model to fill in the missing blanks through the contexts of adjacent

words. For continuous approximation methods, [21] approximated discrete sampling to continuous space for training. [15] proposed a reparameterization trick for Gumbel-softmax, which approximates the discrete distribution to a continuous one by replacing a non-differentiable with a differentiable sample. [17] leveraged the Gumbel-softmax idea into generating texts using GANs, and [23] incorporated Gumbel-Softmax and multiple embedded representations into training GANs.

3 BACKGROUND

In this section, we briefly review several text generation models using autoencoders.

Autoencoder (AE). This is a representative unsupervised model to learn the hidden representation for a given dataset. Specifically, it consists of two components: an encoder $q_\phi(c|x)$, which maps input x to a latent code c, and a decoder $p_\theta(x|c)$, which forces reconstruction of input x from the latent code c. The training of an AE aims to minimize a reconstruction loss.

$$\min_{\phi,\theta} \mathcal{L}_{recon}(x) = -\log p_\theta(x|q_\phi(c|x)), \tag{1}$$

where $\mathcal{L}_{recon}(\cdot)$ denotes the reconstruction loss function.

Since the underlying code of AE is sparse and discrete, a plain AE cannot be used as a generative model. To generate samples from a plain AE, Cífka et al. [7] assumed that a prior distribution on the latent code is either Gaussian or spherical. Although no explicit regularization for the latent code exists, the output of the encoder is restricted to a pre-defined distribution for the latent code. (In experiments, we assume that the latent code of the AE is based on a spherical distribution.)

Variational Autoencoders (VAE) [5]. It is a popular generative model built on the autoencoding architecture. Similar to the conventional AE, VAE [16] uses a reconstruction loss term that forces the model to reconstruct the input. It also imposes a prior distribution close to the standard Gaussian using KL divergence, thus enabling VAE to be a generative model.

Instead of directly computing the marginal probability of $p(x)$, VAE formally optimizes a lower bound on the marginal log-likelihood.

$$\begin{aligned}\min_{\phi,\theta} \mathcal{L}_{recon}(x) + \mathcal{L}_{KL}(x) = \quad & -\mathbb{E}_{q_\phi(c|x)}[\log p_\theta(x|c)] \\ & +KL(q_\phi(c|x) \,||\, p(c)),\end{aligned} \tag{2}$$

where $\mathcal{L}_{recon}(x)$ is the expected reconstruction loss term and $\mathcal{L}_{KL}(x)$ is the KL divergence term between the posterior and prior distributions, where the latter term acts as a regularization for the latent code.

Inspired by [16], Bowman et al. [5] proposed a text generation model by using a VAE. Unlike the conventional VAE, it utilized an RNN encoder and decoder. Also, [5] adopted KL cost annealing to increase the weight of the KL term gradually from 0 to 1 and the word dropout by randomly masking words from the input of the decoder. The optimization methods enable the decoder to utilize the smooth latent codes produced by the encoder.

Adversarial Autoencoder (AAE) [22]. It regularizes the latent code space through adversarial training under the architecture of a GAN. AAE employs a discriminator D to predict whether a given latent code z is a sample drawn from the prior distribution $p(z)$ (*real*) or is produced by the encoder (*fake*). Therefore, it is trained with two loss terms: the reconstruction loss and the adversarial loss term. This means that the discriminator D optimizes the ability to distinguish between fake and real samples.

$$\min_{\phi,\theta} \mathcal{L}_{recon}(x) = -\mathbb{E}_{q_\phi(c|x)}[\log p_\theta(x|c)] \tag{3}$$

$$\min_{\omega} \mathcal{L}_{dis}(x,z) = \mathbb{E}_{q_\phi(c|x)}[\log p_\omega(c)] - \mathbb{E}_{z\sim p(z)}[\log p_\omega(z)] \tag{4}$$

where ω is the parameter of the discriminator D, $p_\omega(\cdot)$ is the probability inferred by the discriminator D, and $z \sim p(z)$ is a genuine sample from the prior distribution $p(z)$. Although $p(z)$ can be an arbitrary prior distribution, in practice, the choice of the prior distribution is usually constrained.

Adversarially Regularized Autoencoder (ARAE) [34]. It regularizes a flexible and complex prior by using adversarial training to match the prior and posterior. Using the latent code c obtained from the encoder as a real latent distribution, it forces the generator G to reproduce the synthetic latent code $\tilde{c}$. In this way, it can overcome the limitation of a simple fixed prior approximation.

The training process for ARAE consists of three loss terms. First, the reconstruction loss is used to train the encoder ϕ and decoder θ. Second, the discriminator loss is used to train the discriminator D to maximize the ability to discern between real and synthetic latent codes. Finally, the generator loss is used to train the generator and encoder to fool the discriminator.

$$\min_{\phi,\theta} \mathcal{L}_{recon}(x) = -\mathbb{E}_{q_\phi(c|x)}[\log p_\theta(x|c)] \tag{5}$$

$$\min_{\omega} \mathcal{L}_{dis}(x,z) = -\mathbb{E}_{q_\phi(c|x)}[\log p_\omega(c)] + \mathbb{E}_{p_\psi(\tilde{c}|z)}[\log p_\omega(\tilde{c})] \tag{6}$$

$$\min_{\phi,\psi} \mathcal{L}_{gen}(x,z) = \mathbb{E}_{q_\phi(c|x)}[\log p_\omega(c)] - \mathbb{E}_{p_\psi(\tilde{c}|z)}[\log p_\omega(\tilde{c})] \tag{7}$$

where ω and ψ are the parameters of the discriminator D and the generator G, respectively. Through this process, it is found that ARAE tends to generate better sentences. However, ARAE easily falls into a mode collapse with a small perturbation in hyperparameters. In other words, ARAE is vulnerable to hyperparameter settings.

Limitations of Existing Models. Despite their success in text generation, previous models suffer from the mode collapse problem, (*i.e.*, most sentences repeat the same words or phrases repeatedly, as reported in Table 3). For instance, existing models are likely to repeat the most frequent words (*e.g.*, "a man"). This is because (1) the prior distribution of existing models is too restricted, and (2) the training process for existing models is too biased for the memorization of the training samples. Although the memorization of high-frequency words is helpful in producing realistic sentences, generating diverse sentences is limited.

4 PROPOSED MODEL

In this section, we propose a new text generation model, namely *diversity regularized autoencoders (DRAE)*. The key novelty of the proposed model is two-fold: (1) We introduce a novel noisy injection method for diverse and robust text generation. Since the proposed model utilizes various input noises, it can synthesize both high-quality and diverse sentences. (2) The proposed noisy injection strategies are model-agnostic. Because ARAE has shown better performance than other models, we focus on incorporating the

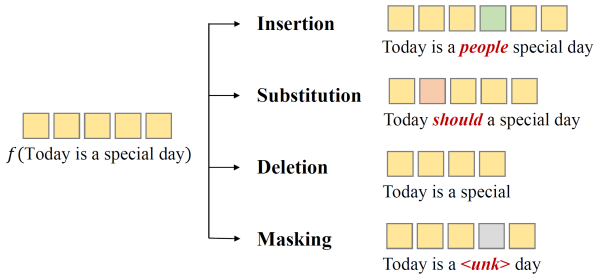

Figure 2: Noise injection strategies used in DRAE.

noisy injection strategy into an ARAE. It is also possible to apply the noisy injection strategy to the other models.

Overview. Figure 1 depicts the overall architecture of DRAE. Specifically, the proposed model improves on ARAE [34] in two respects. (1) It makes use of noise injections into the encoder such that negative noise allows fresh and random words to be intentionally added to the training dataset. Noise injections can be regarded as a data-augmentation rule. The key advantage of noise injections is to avoid memorizing existing data. This is critical because our model aims to generate diverse sentences. In this manner, we can improve the diversity of text generation by smoothing and spreading the latent distribution of the autoencoder. (2) We adopt improved training methods such as WGAN-GP [11] to enhance the training stability of the GAN. Since the training dynamics of GAN models are notoriously unstable, leveraging the improved training methods can directly influence the generation quality. As a result, our model utilizes dual regularization effects by using both the noise injection scheme and gradient penalty in the WGAN-GP.

Encoding. We employ the noise-injection strategy for diverse and robust text representation in the encoder. To implement the encoder, it is possible for using various encoders such as a bidirectional LSTM [13] or transformer [29]. In this paper, we used LSTM to validate the effects of noise injections.

Our motivation for injecting noise is to develop a more robust encoder, which results in compact latent codes. Although the conventional encoder is trained only with real data, we denoised some input tokens to acquire smoother latent representations for text. Specifically, we examine four cases of synthetic noise for each word (*i.e.*, insertion, substitution, masking, and deletion). Interestingly, it is empirically observed that the noise-injection strategy forces the model to remove unnecessary words or add an appropriate article to a proper position. This is because training with noisy inputs is useful for understanding the context of an incorrect sentence, thus helping map it to the correct sentence. We stress that the text generation model armed with such grammar correction functionality has not been investigated in the previous studies.

Noise Injection Strategies. As depicted in Figure 2, we explain various noisy generation methods for a given sample.

- **Insertion**: We randomly insert $k \in \mathbb{N}$ words in each sentence. In this manner, the length of noisy inputs becomes longer than that of original inputs (*e.g.*, I am a student $\rightarrow$ I university am a student.)
- **Deletion**: Deleting randomly selected $k \in \mathbb{N}$ words in each sentence is slightly different from applying a masked language model because the length of noisy inputs decreases and differs from the original inputs, (*e.g.*, I am a student $\rightarrow$ I a student).
- **Substitution**: With randomly selected $k \in \mathbb{N}$ words in each sentence, we replace the selected words with random words in the vocabulary. Following the substitution scheme, the length of noisy inputs is the same as that of the original, (*e.g.*, I am a student $\rightarrow$ I am cat student).
- **Masking**: In a similar vein, we replace randomly selected $k \in \mathbb{N}$ words into <UNK> token. The combination of substitution and masking is on par with applying a masked language model to the encoder as in BERT. Regarding masking strategy, the length of noisy inputs is the same as that of the original, (*e.g.*, I am a student $\rightarrow$ I <UNK> a student.)

We can also unify all four noise injection methods into a model. Let α, β, γ, and δ be the ratios of insertion, substitution, deletion, masking for training, respectively. For example, assume that $\beta = 0.1$, $\delta = 0.8$, $\alpha = \gamma = 0$ and the number of sentences is 10. Of the 10 sentences, one sentence is used as the original, one sentence is subjected to the substitution strategy, and eight sentences are subjected to the masking strategy.

Adversarial Training. ARAE learns a prior distribution using the Wasserstein distance to regularize a discrete autoencoder. As a result, ARAE improves the variability and quality of generating discrete structures. Despite the good properties of Wasserstein distance, the ideal implementation is computationally intractable. Thus, WGAN [1] adopts weight clipping to approximate the 1-Lipschitz constraint, which is identically applied to ARAE. However, as analyzed by WGAN-GP [11], the implementation of WGAN using weight clipping may exhibit pathological behavior, which is well demonstrated in several toy samples. To mitigate training instability, we alternatively adopt an improved training method such as WGAN-GP [11]. WGAN-GP has been applied to many benchmark image datasets and has shown effectiveness in achieving training stability and quality improvement.

Decoding. Given synthetic/real sentences in their fixed-length latent space, we train a model that maps these vectors to their respective sentences. As done with ARAE in [34], we train the LSTM as the decoder. For using other decoders such as a dilated convolutional neural network (CNN) [32] and transformer [29], we leave it for our future work.

5 EXPERIMENTAL SETUP

In this section, we set up experiments to compare the proposed and baseline models. We first describe the two datasets that we used and the training details. We then explain the evaluation metrics to measure the fluency and diversity of the synthesized sentences from each model.

5.1 Datasets

We used two benchmark datasets: the large-scale Stanford Natural Language Inference (SNLI) [4] and the medium-scale BookCorpus (BC) [35]. As our experiments focused on sentence generation, we

Table 1: Statistics of two benchmark datasets

Dataset	Train	Valid	Test	Vocab
SNLI	660K	12K	12K	20K
BC	180K	9K	9K	20K

used only sentences for (sentence, label) pairs. All datasets were filtered with sentences, where the number of words in a sentence was between 5 and 30. For the SNLI and BC datasets, the vocabulary size was limited to 20K for the most frequently seen words in the training dataset. Table 1 provides detailed statistics of the two benchmark datasets.

5.2 Training Details

We used a one-layer LSTM-RNN with a hidden dimension of 300 for both the encoder and decoder as with ARAE in [34]. The generator was a three-layer MLP with a rectified linear unit (ReLU) activation function, and the discriminator was a three-layer MLP with a Leaky ReLU activation function. Whereas the autoencoder was optimized with a stochastic gradient descent at a learning rate of 1, the generator and discriminator were optimized by Adam with a learning rate of 10^{-4} and $\beta_1 = 0.5$ and $\beta_2 = 0.999$. As in [4], we adopted KL cost annealing using a logistic function and applied word dropout with a probability of 0.5 for the decoder when training VAE. Since AE converges much faster than GANs, we trained the GAN more often than AE, where the iteration ratio of AE over GAN increased by 100 percent after every two epochs. The batch size was set to 64. The discriminator was trained for five iterations in every loop.

5.3 Evaluation Metrics

We evaluated text generation models on three tasks: (1) sampling, (2) reconstruction, and (3) latent space walking. To assess the fluency and diversity of our model on the sampling task, we used forward perplexity (FPPL), reverse perplexity (RPPL), self-BLEU, and distinct. For the reconstruction task, we assessed how well the model could reconstruct a given sentence, and we used the BLEU score.

Forward Perplexity (FPPL). FPPL is used to measure the quality of each generated sentence. FPPL is the expected negative log-probability of samples from the model with respect to the true data distribution. Since $p_{\text{data}}(x)$ is unknown, it can be approximated by using a language model. Here, we used a pre-trained KN 5-gram language model on real data.

$$\mathbb{E}_{p_G(x)}\left[-\log p_{\text{data}}(x)\right] \tag{8}$$

Reverse Perplexity (RPPL). RPPL is used to measure the variety of the generated sentences. Because $p_G(x)$ is intractable for any given x, we approximated $p_G(x)$ using an RNN language model trained on 100K generated samples. We then evaluated the LM on a test set. If the generated sentences were diverse, which means they covered the training dataset to a good extent, it would have better (lower) reverse perplexity measures where these results from the trained LSTM network (language model).

$$\mathbb{E}_{p_{\text{data}}(x)}\left[-\log p_G(\text{x})\right] \tag{9}$$

Bilingual Evaluation Understudy (BLEU) [24]. BLEU is a popular metric that measures the geometric mean of the modified n-gram precision with a brevity penalty (BP). The BLEU score quantifies the similarity degree between generated sentences and the ground-truth. For the BLEU score evaluation, we followed the strategy given in [33] of using the entire test set as the reference set. We computed the BP and BLEU as follows:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{10}$$

$$\text{BP} = \begin{cases} 1 & \text{if } l' > l \\ e^{(1-l/l')} & \text{if } l' \leq l \end{cases} \tag{11}$$

where l and l' denote the lengths of real and synthetic sentences, respectively.

Self-BLEU. Self-BLEU is a measure of diversity for generated texts [36]. For each generated sentence, we computed the BLEU using the sentence as a hypothesis and the remaining generated sentences as the reference. When averaged over all references, BLEU gives us a measure of the extent to which sentences are diversified. Lower Self-BLEU scores are better, as high BLEU scores indicate a higher degree of similarity.

Distinct. A Distinct measure is calculated by counting the number of unique k-grams denoted as Dist-k. This metric has been widely used in existing studies [3, 9].

Human Evaluation. We conducted a user study, in which six native speakers evaluated the fluency of 20 randomly generated sentences from real data and each model based on a 5-point Likert scale.

6 EXPERIMENTAL RESULTS

In this section, we conducted three experiments for *sampling*, *reconstruction*, and *latent space walking*. First, a sampling task was conducted to measure how well the model approximates the underlying data distribution using RPPL, FPPL, Self-BLEU, Distinct, and human judgments. Next, a reconstruction task was conducted to evaluate the quality of a reconstructed sentence as compared to a given input sentence using BLEU. Finally, latent space walking showed the interpolated sentence to the given start and end sentences.

6.1 Sampling

We first generated 10K sentences by drawing random samples from the prior $\mathcal{N}(0, 1)$ and then gauged the fluency and diversity of the generated samples from each model. To measure the fluency of the sentences, we used both FPPL and human evaluation scores. As the key objective of our approach is to achieve diversity, we reported the diversity using various evaluation metrics, including RPPL, Self-BLEU, and distinct. Each diversity metric quantifies the diversity degree at a different angle. For example, a lower RPPL signifies that the generated sentences are sufficiently diverse to approximate the coverage of the dataset. Self-BLEU is an indicator of how well the sentences differ from each other within the generated samples. Distinct is used to measures the score for the uniqueness of k-grams. Overall, RPPL, Self-BLEU, and Distinct are the diversity measures

Table 2: Quantitative evaluation results for sampling task in SNLI (left) and BC (right) dataset. RPPL and FPPL stand for Reverse Perplexity and Forward Perplexity, respectively. SB and Dist indicate Self-BLEU and Distinct, respectively. Also, ins, sub, del and unk is the abbreviation for insertion, substitution, deletion and masking, respectively. Note that DRAE-all indicates the combination of substitution and masking techniques, with the ratio of 0.1 and 0.8.

Model	SNLI						BC					
	RPPL	FPPL	SB-4	SB-5	Dist-1	Dist-2	RPPL	FPPL	SB-4	SB-5	Dist-1	Dist-2
AE	270.56	906.21	**39.01**	29.97	0.01	0.14	1067.35	1732.58	30.26	23.88	0.01	0.17
VAE	1624.40	1075.20	83.52	79.68	0.03	0.09	1123.56	819.23	87.27	84.47	0.02	0.07
AAE	277.56	958.28	39.99	27.73	0.01	0.15	1027.52	1892.99	29.29	23.37	0.01	0.18
ARAE	100.86	129.17	40.37	25.98	0.03	0.34	357.45	441.39	20.16	13.96	0.06	0.56
DRAE-ins	98.53	136.05	39.81	**25.84**	**0.04**	**0.36**	327.48	451.91	**19.55**	**13.59**	**0.07**	**0.57**
DRAE-sub	96.25	88.98	48.34	32.51	0.04	0.30	**314.22**	259.86	26.34	17.83	0.06	0.45
DRAE-del	97.67	79.90	50.88	34.54	0.03	0.28	315.19	**221.12**	27.97	18.92	0.06	0.45
DRAE-unk	96.75	102.41	46.12	30.69	0.04	0.32	322.78	337.21	22.93	15.80	0.06	0.51
DRAE-all	**96.03**	**77.56**	51.94	35.10	0.03	0.26	323.42	239.46	27.08	18.46	0.05	0.44

for the synthesized sentences in terms of *dataset coverage*, *sentence similarity*, and *word similarity*, respectively.

Table 2 reports the quantitative sampling results for each model. For the SNLI dataset, our model outperformed the baselines with respect to RPPL, FPPL, SB-5, Dist-1, and Dist-2. Although the AE showed better performance on SB-4 for the SNLI dataset, the Distinct metric was considerably low. This suggests that each sentence was different, but the same words were consistently repeated within a sentence. For the BC dataset, our model excelled in all the evaluation metrics for sampling. Interestingly, insertion demonstrated the best performance with Self-BLEU and Distinct for both datasets. In addition, substitution and deletion outperformed the baselines with RPPL and FPPL, respectively.

For the qualitative evaluation, Table 3 shows five generated sentences for each model. The generated samples from AE and AAE were not realistic. In addition, although VAE showed more realistic sentences than AE and AAE, they all merely started with the phrase *"A man"*. Compared to all of the baselines, ARAE showed fairly fluent sentences. Finally, the proposed model outperformed ARAE in all evaluation metrics. Albeit simple, our model using noise injection strategies showed the best performance because it generated more diversified sentences without compromising on their fluency.

In addition to the objective evaluation metrics, we also carried out a user study to assess the subjective quality of generated sentences from each model. The six native English speakers evaluated 20 randomly selected sentences from each model as well as real sentences. The participants scored each sentence on a 5-point Likert scale, with 240 sentences in total. During the evaluations, we provided instructions for scores as follows: "1" for sentences that were both ungrammatical and incomprehensible, "3" for those that were either ungrammatical or incomprehensible, "5" for those that were grammatically correct and understandable. We averaged the scores from four variants of the proposed model and reported it as the subjective quality of DRAE. As Table 4 shows, the proposed model scored significantly better than other approaches. In particular, the gap between our DRAE and the second-best algorithm was approximately 0.8/0.6, which was comparable to the gap between the real sentence and DRAE (0.76/0.98). In other words, the proposed model could generate much more diverse and high-quality sentences than the baseline models.

6.2 Reconstruction

As Table 5 shows, the proposed models tended to have worse BLEU scores on reconstruction. This means that our models did not reconstruct the same sentence to the given input. Yet, our reconstruction made sense because the reconstructed output was semantically similar to the input in terms of either structure or contextual sense.

Tables 6 and 7 show reconstructed sentences for the given input from each model. Interestingly, DRAE-ins tended to delete a word while keeping its grammar intact. Similarly, DRAE-del tended to add a word without losing the semantics and syntax of the given sentence. This is a particularly attractive property, as the model eventually learns to retain the grammar of a sentence, regardless of whether a word is removed or added. In addition to grammar correction, our model successfully paraphrased the sentence as suitable for the given input. However, the baselines were likely to reconstruct the same sentence or generate a dissimilar sentence. For example, as shown in Table 6, the meaning of *"caught stealing"* was changed a lot with the AE, AAE, and ARAE, but the meaning of *"practicing fast"* and *"caught stealing"* have nothing in common, so this change was improper. From this observation, we conjectured that the previous models learned to copy all inconsistent sentences to the given source sentence or generate new ones. Even when different sentences are generated, losing the semantics should be avoided. In addition, our model using deletions showed some idiosyncratic results. For example, DRAE-del occasionally repeated a word (*e.g.*, "love love"), which produced improper sentences, as presented in Table 7.

That is, our models must correctly understand the meaning of the input and decode it with flexibility. Since the objective functions of our models are combined with the reconstruction and generation terms, an accurate reconstruction can lead to memorization, which is rather undesirable for improving the quality and diversity of text generation.

Table 3: Synthesized sentences for each model trained using SNLI dataset with a maximum word length of 30.

Model	Generated samples
AE	Five are have them are setting . The He in a a little a puppy . Some woman are two Three men are outside . The his takes the bar . two friends for all stand .
VAE	A man is playing a bike . A man is in his bike . A man is on a field . A man is playing a bike . A man is playing a guitar .
AAE	Some girls are dance . Two hockey on on on with on on . a girl wearing a hat . Two soccer play soccer run soccer . a person eating a statue .
ARAE	The man is wearing pajamas . The woman is in a square building . A man fell over a concrete ramp . A person is getting with a ball . The man is sweeping .
DRAE-ins	The woman is playing Monopoly . Some children sitting on a bench . A boy is playing on the bed . A woman standing among the young man . The dog is indoors and a white dog .
DRAE-sub	Two men are getting married . A person is happily sitting down . Someone in shorts in a field of grass . A couple is watching a rodeo . People having a blue conversation .
DRAE-del	A person is riding a sidewalk . He is getting ready for by clothing . A girl is in a red vehicle , towards the ocean . A girl is doing various pedestrians . The dogs are late .
DRAE-unk	Soccer in a track in silence . Two girls are spinning together . People play for a routine . A girl is jumping ahead of the man . A man in a camouflage operates ladder .
DRAE-all	A blonde child is brushing it in swim . A dog jumps on a beach with a rabbit . There is a clean car in the water . A boy shows his favorite shop outside . Two kids race at the beach .

Table 4: Human evaluation results of the various models and real data. Each score is 5-scale point. Higher value indicates better result in terms of fluency.

Dataset	AE	VAE	AAE	ARAE	DRAE	Real
SNLI	1.49	3.26	1.83	3.18	**4.08**	4.84
BC	1.53	2.71	1.45	2.53	**3.30**	4.28

Table 5: Reconstruction results of various models and ours w.r.t. BLEU-n scores. B-n indicates BLEU-n.

Model	SNLI			BC		
	B-3	B-4	B-5	B-3	B-4	B-5
AE	78.42	74.61	71.19	77.82	73.11	68.83
VAE	82.58	79.25	75.48	84.57	81.67	78.92
AAE	77.55	73.58	70.00	85.59	81.44	77.34
ARAE	84.63	81.81	79.21	87.52	84.62	81.81
DRAE-ins	76.46	70.64	65.04	66.36	56.85	48.59
DRAE-sub	66.65	59.34	53.04	76.43	70.20	64.53
DRAE-del	67.64	60.09	53.08	59.88	50.02	41.43
DRAE-unk	81.41	77.73	74.39	87.41	84.34	81.38
DRAE-all	80.26	76.15	72.45	80.81	76.16	71.86

6.3 Latent Space Walking

We performed latent space walking between two points z_1 and z_2 from $\mathcal{N}(0, 1)$ to achieve the following objectives: 1) to better understand the representation power of the learned latent space, and 2) to confirm that our generation model did not memorize the training dataset. We randomly sampled two points z_1 and z_2 from $\mathcal{N}(0, 1)$ and then produced intermediate points z_{walking} by smoothly walking between z_1 and z_2. We passed this z_{walking} to the generator and then constructed a sentence using greedy decoding or sampling from the latent space. In this experiment, we generated eight intermediate sentences by linear interpolations (Let $\delta = [0, 1/7, ..., 1]$).

$$z_{\text{walking}} = (1 - \delta)z_1 + \delta z_2 \quad \text{where } \delta \in [0, 1] \tag{12}$$

Note that the successful transitions in the latent space walking was often a strong indicator that (1) the generative model did not memorize the training dataset, and (2) its latent space was dense and compact. This allowed us to deform a sentence continuously by manipulating the volume-filling latent code space, thus enabling us to understand how neighboring sentences would appear.

Table 8 shows some examples of latent space walking with ARAE and DRAE. (Note that because of page limitations, we show only the results for DRAE-all and DRAE-sub). These results are impressive, as we observed smooth and coherent translations by latent space walking in terms of structure or meaning of words, while each generated sentence was natural and different from others. Especially, DRAE-all showed a transition from the phrase of *"playing Monopoly"* to *"engaged about activity"*, then to the verb of *"compete"*, and finally to the verb of *"study"*. These four expressions were smoothly connected based on their semantics. This was an important milestone, as existing models have difficulties generating

Table 6: Reconstructed outputs to the given input "The boy was caught stealing" from each model trained on SNLI. Whereas the baseline models generate dissimilar sentences to the given input, our models do not lose the semantics.

Input	The boy was caught stealing .
AE	The boy was caught ***groceries*** .
VAE	The boy was caught stealing .
AAE	The boy was caught ***fish***.
ARAE	The boy was ***practicing fast***.
DRAE-ins	The boy was caught .
DRAE-sub	The boy was caught ***out*** .
DRAE-del	The boy was caught ***after*** stealing .
DRAE-unk	The boy was caught stealing .

Table 7: Reconstructed outputs to the given input "you think of everything, my love" from each model trained on BC. Whereas baseline models parrot back the given input, our models tend to paraphrase the given input.

Input	you think of everything , my love .
AE	you think of everything , my love .
VAE	you think of everything , my love .
AAE	you think of everything , my love .
ARAE	you think of everything , my love .
DRAE-ins	you think ***of*** , my love .
DRAE-sub	you think of everything , my ***dear*** .
DRAE-del	you think of everything , my ***love love*** .
DRAE-unk	you think of everything , my love .

semantically meaningful transitions through latent space walking. Specifically, ARAE generated somewhat inconsistent sentences or rarely changed sentences while walking in latent space. This means that the latent space for ARAE was full of holes and disconnected. For instance, the first example of ARAE given in Table 8 shows that the sentence was nearly the same as the previous sentence, and the second example lists inconsistent sentences between start and end points.

7 CONCLUSION

In this paper, we proposed a new text generation model that aims to generate diverse and fluent sentences. By adopting a newly proposed noise injection to an encoder, the proposed model can yield a smooth and well-spread latent distribution. Although our model is simple, it is effective at producing diverse and natural-looking sentences. Experimental results showed that our model achieved better performance than the baseline models in terms of diversity metrics such as RPPL, Self-BLEU, and Distinct. In future work, we will combine the noise-injection strategies with different architectures such as a dilated CNN and transformer. In addition, we will extend our model to conditional supervised tasks such as sentence simplification and paraphrase generation.

Table 8: Linear interpolations between pairs of two random points on SNLI. ARAE (authors) is taken from [34].

Model	Sample interpolations
ARAE (authors)	A man is on a ship path with the woman . A man is on a ship path with the woman . A man is passing on a bridge with the girl . A man is passing on a bridge with the girl . A man is passing on a bridge with the girl . A man is passing on a bridge with the dogs . A man is passing on a bridge with the dogs .
ARAE	A young girl is visiting the girl in a purple blouse . A young soldier is playing in the evening cutting a heart . An African american is pointing to win the bubble heart . The young soldier is observing the toddler free an belly dancer . The young diver is inspecting the bird free blond . The orange sculptor is pulling his finger fruit . The large woman climbs a truck is white ice cream . The large black box with a large bull is picking cats .
DRAE -sub	Men are walking outside . Men are playing together . People are eating sandwiches . People are barbecuing sandwiches . Boys are receiving beers . Boys prepare to christmas tea . Boys are preparing to play a card . The boys want to tag at a picnic .
DRAE -all	A tennis band player is playing Monopoly . Two younger soccer players are playing Monopoly . Two tennis players are engaged about activity . Two tennis players are about to move . Two girls are about to compete . Two students are about to study . Two women are about to study project . Two women are going to their pictures .

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant (No. NRF-2018R1A2B6009135) and by the Korean National Police Agency and the Ministry of Science and ICT for Police field customized research and development project (No. NRF-2018M3E2A1081572). Also, this work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00421, AI Graduate School Support Program and No. 2019-0-01590, High-Potential Individuals Global Training Program).

REFERENCES

[1] Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *ICML*. 214–223.
[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.

[3] Hareesh Bahuleyan, Lili Mou, Olga Vechtomova, and Pascal Poupart. 2018. Variational Attention for Sequence-to-Sequence Models. In *COLING*. 1672–1682.

[4] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*. 632–642.

[5] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *CoNLL*. 10–21.

[6] Tong Che, Yanran Li, Ruixiang Zhang, R. Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. 2017. Maximum-Likelihood Augmented Discrete Generative Adversarial Networks. *CoRR* abs/1702.07983 (2017).

[7] Ondrej Cífka, Aliaksei Severyn, Enrique Alfonseca, and Katja Filippova. 2018. Eval all, trust a few, do wrong to none: Comparing sentence generation models. *CoRR* abs/1804.07972 (2018).

[8] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. 2018. Hyperspherical Variational Auto-Encoders. In *UAI*. 856–865.

[9] William Fedus, Ian J. Goodfellow, and Andrew M. Dai. 2018. MaskGAN: Better Text Generation via Filling in the _____. *CoRR* abs/1801.07736 (2018).

[10] Jules Gagnon-Marchand, Hamed Sadeghi, Md. Akmal Haidar, and Mehdi Rezagholizadeh. 2019. SALSA-TEXT: Self Attentive Latent Space Based Adversarial Text Generation. In *Canadian AI*. 119–131.

[11] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. Improved Training of Wasserstein GANs. In *NeurIPS*. 5767–5777.

[12] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long Text Generation via Adversarial Training with Leaked Information. In *AAAI*. 5141–5148.

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[14] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward Controlled Generation of Text. In *ICML*. 1587–1596.

[15] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *ICLR*.

[16] Diederik P. Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *CoRR* abs/1312.6114 (2013).

[17] Matt J. Kusner and José Miguel Hernández-Lobato. 2016. GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution. *CoRR* abs/1611.04051 (2016).

[18] Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Generating Reasonable and Diversified Story Ending Using Sequence to Sequence Model with Adversarial Training. In *COLING*. 1033–1043.

[19] Kevin Lin, Dianqi Li, Xiaodong He, Ming-Ting Sun, and Zhengyou Zhang. 2017. Adversarial Ranking for Language Generation. In *NeurIPS*. 3155–3165.

[20] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*. 1412–1421.

[21] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *ICLR*.

[22] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian J. Goodfellow. 2015. Adversarial Autoencoders. *CoRR* abs/1511.05644 (2015).

[23] Weili Nie, Nina Narodytska, and Ankit Patel. 2019. RelGAN: Relational Generative Adversarial Networks for Text Generation. In *ICLR*.

[24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL*. 311–318.

[25] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *ACL*. 1073–1083.

[26] Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A Hybrid Convolutional Variational Autoencoder for Text Generation. In *EMNLP*. 627–637.

[27] Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. 2017. A Conditional Variational Framework for Dialog Generation. In *ACL*. 504–509.

[28] Xiaoyu Shen, Hui Su, Shuzi Niu, and Vera Demberg. 2018. Improving Variational Encoder-Decoders in Dialogue Generation. In *AAAI*. 5456–5463.

[29] 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 *NeurIPS*. 5998–6008.

[30] Yijun Xiao, Tiancheng Zhao, and William Yang Wang. 2018. Dirichlet Variational Autoencoder for Text Modeling. *CoRR* abs/1811.00135 (2018).

[31] Jiacheng Xu and Greg Durrett. 2018. Spherical Latent Spaces for Stable Variational Autoencoders. In *EMNLP*. 4503–4513.

[32] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved Variational Autoencoders for Text Modeling using Dilated Convolutions. In *ICML*. 3881–3890.

[33] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *AAAI*. 2852–2858.

[34] Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun. 2018. Adversarially Regularized Autoencoders. In *ICML*. 5897–5906.

[35] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *ICCV*. 19–27.

[36] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A Benchmarking Platform for Text Generation Models. In *SIGIR*. 1097–1100.