ADVERSARIAL GRAMMATICAL ERROR GENERATION: APPLICATION TO PERSIAN LANGUAGE

Nassibeh Golizadeh[1], Mahdi Golizadeh[1] and Mohamad Forouzanfar[2]

[1]Faculty of Electrical & Computer Engineering, University of Tabriz, Tabriz
[2]Faculty of Electrical Engineering, K.N. Toosi University of Technology, Tehran

ABSTRACT

Grammatical error correction (GEC) greatly benefits from large quantities of high-quality training data. However, the preparation of a large amount of labelled training data is time-consuming and prone to human errors. These issues have become major obstacles in training GEC systems. Recently, the performance of English GEC systems has drastically been enhanced by the application of deep neural networks that generate a large amount of synthetic data from limited samples. While GEC has extensively been studied in languages such as English and Chinese, no attempts have been made to generate synthetic data for improving Persian GEC systems. Given the substantial grammatical and semantic differences of the Persian language, in this paper, we propose a new deep learning framework to create large enough synthetic sentences that are grammatically incorrect for training Persian GEC systems. A modified version of sequence generative adversarial net with policy gradient is developed, in which the size of the model is scaled down and the hyperparameters are tuned. The generator is trained in an adversarial framework on a limited dataset of 8000 samples. Our proposed adversarial framework achieved bilingual evaluation understudy (BLEU) scores of 64.5% on BLEU-2, 44.2% on BLEU-3, and 21.4% on BLEU-4, and outperformed the conventional supervised-trained long short-term memory using maximum likelihood estimation and recently proposed sequence labeler using neural machine translation augmentation. This shows promise toward improving the performance of GEC systems by generating a large amount of training data.

KEYWORDS

natural language processing; grammatical error correction; grammatical and semantic errors; natural language generation; generative adversarial network

1. INTRODUCTION

Natural language processing (NLP) relies on the design of powerful and intelligent algorithms that can automatically process and understand complex patterns within the text [1-3]. Automatic NLP systems that can detect and correct grammar and word usage errors require a large amount of correct and incorrect labeled data. As a result, the performance of automated error correction systems is arguably too low in practical applications mainly due to the limited amount of training data [4]. While correct sentences can be found easily in many resources such as CLC (http://www.cambridge.org/elt/corpus/learner_corpus2.htm), labeled incorrect sentences are very limited, especially in languages other than English. To solve this problem, one can manually gather and produce labeled incorrect sentences which are time-consuming and prone to human errors. The other solution is to supplement existing manual annotation with synthetic instances. Such artificial samples can be produced using deep learning techniques such as deep generative models [5].

DOI: 10.5121/ijnlc.2022.11402

The main idea of generative models is to approximate the distribution of basic data by training a model that best fits them. By learning the distribution of data, generative models can generate observable data values. Therefore, a large amount of data can be generated by training a generative model with a small amount of raw data. Various generative models have been proposed in the literature, such as latent Dirichlet distribution [6], restricted Boltzmann machines [7], and generative adversarial networks (GANs) [8], which use the adversarial training idea for generating more realistic data samples. Among the existing generative models, GANs have attracted more attention. The main idea of GAN is to play a min–max game between a discriminator and a generator, i.e., adversarial training. The discriminator tries to differentiate between real data and data generated by the generator (fake data), while the generator tries to generate data that is recognizable as real data by the discriminator. GANs are extremely powerful in generating artificial images and have facilitated many applications. For example, a GAN-based image generator can produce super-resolution images from their low-resolution versions[9], create realistic images from some sketches [10], or perform auto painting [11].

GAN’s remarkable performance in generating realistic images has led to its application in NLP tasks for sentence generation. For example, Zhang et al. [12] and Semeniuta et al. [13] used GANs for text generation and achieved state-of-the-art results. A modified GAN (dialogue-GAN) was proposed in [14], demonstrating the ability of GAN in generating realistic dialogues. In [15], a deep variational GAN was developed to generate text by modifying the standard GAN objective function to a variational lower-bound of the log-likelihood. In [16], adversarial learning was applied to subsequences, rather than only to the entire sequences, in a GAN framework to improve unconditional text generation. However, the existing work in the field of text generation is mainly focused on the generation of correct sentences in languages such as English and Chinese. More research is required for the adoption of GAN for the processing of other languages.

In this paper, we propose a modified GAN for the generation of grammatical and semantic erroneous sentences that are required to train text error detection systems. The focus is on the development of an error-generating system for the Persian language, which is still lacking. Because of the Persian language’s different structure, compared to English and Chinese languages, it constitutes different grammatical and semantic errors. As such, the current training error detection systems do not apply to the Persian language. Here, the sequence generative adversarial nets (Seq-GAN) [17], is adopted to learn Persian sentences with common grammatical and semantic errors and to regenerate them to be used to train error detection systems. The performance is compared with the state-of-the-art methods in terms of bilingual evaluation understudy score, or BLEU.

2. BACKGROUND

In computer vision, new training instances can be created by simply blurring, rotating, or deforming the existing images with a small amount of processing [18], but a similar procedure cannot be performed on language data because mutating even a single letter or a word can change the whole sentence meaning or render it nonsensical. As there are vast differences between languages, transfer learning [19] cannot be applied in language models as it has been applied in computer vision. In other words, for every individual language, one must create their own dataset and models. As creating such datasets is challenging, many automatic generative methods have been proposed in the literature.

In [20], Brockett et al. developed a phrasal statistical machine translation technique to identify and correct writing errors made by English learners as a second language. They used examples of mass noun errors found in the Chinese Learner Error Corpus (CLEC) to create an engineered training set. In [21], Rozovskayaand Roth incorporated error generation methods to introduce mistakes in the training data. In [22], Foster and Andersen leveraged linguistic information to identify error patterns and transfer them onto the grammatically correct text. In [23], Imamura et al. presented an error correction method of Japanese particles that uses pseudo error generation. To achieve this goal, they used a domain adaptation technique. In [24], Rei et al. proposed treating error generation as a machine translation task. They used their model to translate correct text to the ones that contained errors. They also proposed a system capable of extracting textual patterns from annotated corpus which could be used togenerate incorrect sentences by inserting errors into grammatically correct sentences. In [25], Yannakoudakis et al. proposed an approach to N-best list ranking using a neural sequence-labeling model that calculates the probability of each token in a sentence being correct or incorrect in context. In [26], Kasewa et al. used neural machine translation to learn the distribution of language learner errors and utilized it as a data augmentation technique to induce similar errors into the grammatically-correct text. In [27], Lee et al. introduced a Korean GC model based on transformers with a coping mechanism. They also used a systematic process for generating grammatical noise which reflects general linguistic mistakes.

While in previous studies almost all the attention has been placed on creating incorrect sentences from correct ones, in this paper, we propose a new method that learns to generate new incorrect sentences from limited available incorrect samples. This can provide a large amount of training data to be used by grammatical error correction (GEC) systems. The focus is on the Persian language that has not been studied so far in this application.

3. MATERIALS AND METHODS

3.1. Persian Sentence Generation with Adversarial Networks

In our approach, unlike previous studies that convert a correct sentence to a wrong sentence, we prepared a dataset of sentences with common grammatical and semantic errors. Our goal was to learn and model the distribution of the errors and use them to expand our already existing data.
To expand our dataset, we used deep generative models [28]. Among generative models, GANs have shown excellent performance in different applications such as image processing [11, 29].
GAN contains two networks, the discriminator and the generator, that contest with each other in a game where one network's gain is another network's loss. The generator tries to produce fake data that are as close as possible to real data. The discriminator differentiates between real data and the fake data generated by the generator. However, GANs are initially defined to process real-valued continuous dataand therefore they may not work well with discrete data such as text.
There exist two approaches to solve the aforementioned problem. One approach is based on the approximation of the discrete input to have a differentiable loss function. GAN can then be trained using gradient descent algorithms. For example, Softmax has been used to approximate the one-hot encoded data [12, 30]. The second approach is to use reinforcement learning to train the discriminator in discrete scenes. Seq-GAN lays out the discriminator as a policy on the pretrained maximum likelihood estimation (MLE) model which is trained using policy gradient [17]. As the quality evaluation of the generated text is difficult to specify, GAN is suitable for this problem.

In this paper, we adopted Seq-GAN [17], which has the best performance in generating text among other proposed GAN models. Seq-GAN addresses the problem of discrete tokens by directly performing gradient policy updates. Complete sequences are judged by the reward signal which comes from the GAN's discriminator (Figure 1).

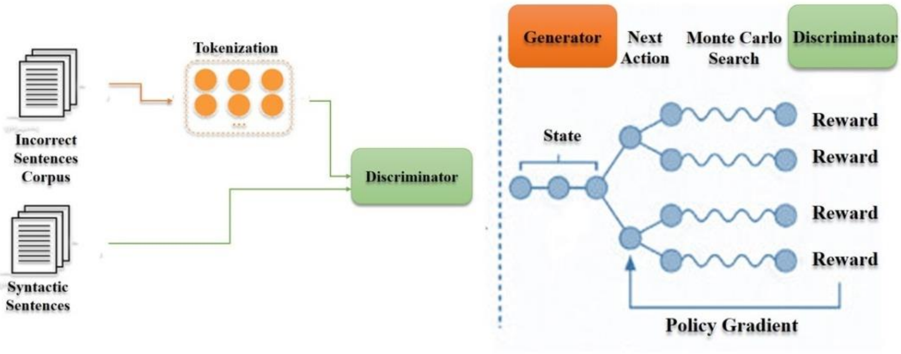


Figure 1. An illustration of our proposed method to generate Persian sentences with common grammatical and semantic errors.

As this model yields better results when trained on a big dataset, the size of the model and its hyperparameters were tuned to make it appropriate for training on a relatively small dataset (8000 samples).

The generator part of our proposed model was the same as the original Seq-GAN. To match the network to our data, only the Monte Carlo search window size was changed to 8. A CNN-based classifier was chosen as the discriminator. As the model is directly trained on true data and an oversized discriminator can be overfitted easily on a small dataset, the kernel size in convolutional layers was halved but the size of windows and the number of convolutional layers were not altered.

Because of the lack of a robust model in the same field for Persian language to compare with our proposed model, we trained our dataset on a long short-term memory (LSTM) with supervised learning as a baseline for comparison. In addition, we compared the performance of our method with the state-of-the-art sequence labeler using neural machine translation augmentation (SL-NMT) [26].

3.2. Dataset

The data used in this article were sentences with grammatical and semantic errors in thesis drafts. To decrease the number of words used in network training, drafts of theses with similar topics in the field of computer science were used. Our dataset consisted of more than 8000 sentences with the mentioned features. It should be noted that sentences with the highest number of common errors were selected.

Word-level tokenization was used to break the sentences into words. After tokenizing the sentences, 4144 tokens of words were obtained. Because the goal was to generate sentences with semantic and grammatical errors, all punctuation marks were also considered in tokenization (such as semicolons, commas, question marks, exclamation marks, and so forth).

3.3. Model Training

The training process of our proposed model consists of two main parts: pretraining and adversarial training. These two steps are as follows [17]:

Pretraining: To increase the convergence speed of the model and to achieve the desired outcome with a smaller number of steps, we pretrained the generator on our main data using the MLE method. The advantage of this type of training is that the weights of the network will not be random during the main training. After pretraining of the generator, a random batch of data was generated using the generator and used as a negative class along with the main data as a positive class for the discriminator pretraining.

Adversarial training: This consisted of two parts that were repeated alternately with a specified number of epochs (here, set to 60). In the first step, the generator generated a sentence as large as the maximum size of the real data. Then, the value of a cost function $Q_{D_\Phi}^{G_\theta}$ was calculated for the entire length of the sentence. The parameters of the generating network were updated using the policy gradient as follows [17]:

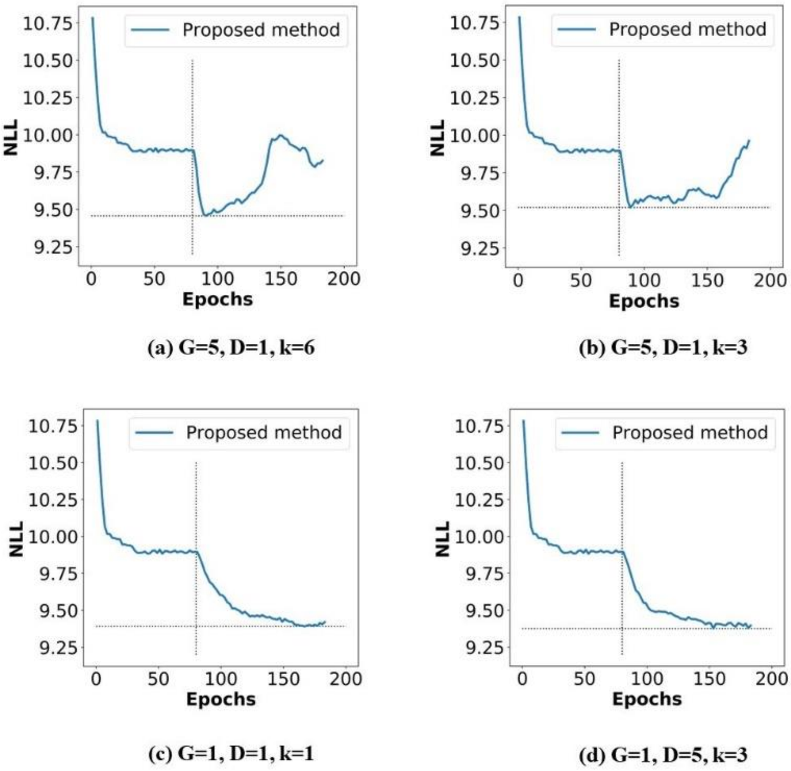
$$\boldsymbol{Q_{D_\Phi}^{G_\theta}(s = Y_{1:t-1}\text{‘}a = y_t)} = \begin{cases} \boldsymbol{\frac{1}{N}\sum_{n=1}^{N} D_\Phi(Y_{1:T}^{n})\text{‘}\, Y_{1:T}^{n} \in MC^{G_\beta}(Y_{1:t}; N) for t < T} \\ \boldsymbol{D_\Phi for t = T} \end{cases} \quad (1)$$

where t is the time step, N is the Monte Carlo window size, $\boldsymbol{a = y_t}$ is next action, $\boldsymbol{s = Y_{1:t-1}}$ is the current generated tokens, $\boldsymbol{D_\Phi(Y_{1:t})}$ is reward given by the discriminator to the generator, $\boldsymbol{MC^{G_\beta}}$ represents the Monte Carlo N-time search, and T is the maximum length of sentences.
In the second step, a batch of data was generated by the generator as large as the maximum size of the real data, and after preprocessing the generated sentences, it was used as a negative batch with a random batch of the main data to train the discriminator.

The first and second steps were repeated until the model converged.

4. Results and Discussions

The stability of the developed model depends on training strategy, i.e., the number of generator training steps (G), the number of discriminator training steps for discriminator training (D), and the number of discriminator training epochs (K). Figure 2 shows the effect of parameters G, D, and K on the negative log-likelihood convergence performance of the proposed method in generating Persian sentences with common grammatical and semantic errors.



(a) G=5, D=1, k=6

(b) G=5, D=1, k=3

(c) G=1, D=1, k=1

(d) G=1, D=5, k=3

Figure 2. Negative log-likelihood (NLL) convergence performance of the proposed method in generating Persian sentences with common grammatical and semantic errors using different hyperparameters: (a) G = 5, D = 1, k = 6; (b) G = 5, D = 1, k = 3; (c) G = 1, D = 1, k = 1; (d) G = 1, D = 5, k = 3. The vertical dashed lines represent the beginning of adversarial training.

In Figure 2a,b, the number of steps G is more than D (G = 5, D = 1). That is, the generator is five times more trained than the discriminator. This leads to rapid generator convergence, but the discriminator does not have enough data for training. Therefore, it gradually leads to incorrect outputs because the non-well-trained discriminator does not have the ability to accurately classify the sentences. Therefore, in this case, the generated sentences may not be very acceptable to the human observer.

In Figure 2c, the generator and discriminator are trained only one step before generating output (K = 1, G = 1). In this way, the discriminator is trained with the training data and the data generated by the generator for one epoch. In addition, the generator is only allowed one epoch to update its weights through adversarial training. As it can be observed, the results are stable.

In Figure 2d, the number of discriminator training steps is five times more than the generator (D = 5, G = 1). The generator produces five different batches of data as negative samples. The bootstrap method was used to form five different batches of real-world data. This type of training lets the discriminator learn all the negative sentences which are generated by the generator. Therefore, the discriminator is not easily fooled by the generator. It can be observed from Figure 2d that as the number of training epochs increases, the loss decreases.

According to Figure 2, it can be observed that the pretraining stage reaches stability after a certain epoch (~75) and the continuation of the supervised training stages does not improve the performance of the generative model. This indicates that increasing the pretraining steps does not affect the pretraining process. Therefore, it is better to perform pretraining until this epoch (75).

According to the explanations presented in the previous sections and the explanations related to the pretraining step, the proposed approach consists of 80 pretraining epochs in which training is supervised (it was shown that the pretraining error does not improve after 75 epochs, so to be on the safe side, we chose 80 epochs), 60 adversarial training epochs (it was observed that the error does not improve after 60 adversarial training epochs), and effective hyperparameters of G = 1, D = 15, and K = 3. Figure 3 compares the performance of the proposed method versus the supervised MLE LSTM approach. Comparing Figure 3 with Figure 2d, it can be observed that the change of hyperparameters has led to the stability of adversarial training. According to Figure 3, it can be observed that the lowest value of loss function is obtained in epoch 140. The vertical line separates the pretraining steps from the main training.

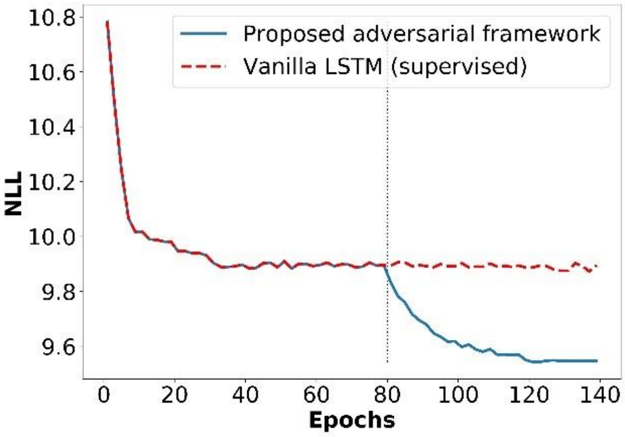


Figure 3. Negative log-likelihood convergence with respect to the training epochs. The vertical dashed line represents the end of pretraining for Persian-GAN and MLE.

BLEU score [31] was used to evaluate the proposed method. BLEU is an algorithm developed for assessing the quality of machine-translated text. The main idea behind BLEU is to evaluate the quality based on the correspondence between a machine's output and that of a human's judgment of quality. It is one of the most popular automated and inexpensive metrics for the assessment of language models' quality. Table 1 summarizes the BLEU scores obtained by our proposed method, a baseline conventional technique (LSTM), and a state-of-the-art approach (SL-NMT). It was observed that our proposed method achieved an improvement of 31.36% in the BLEU-2 score, 15.40% in the BLEU-3 score, and 8.62% in the BLEU-4 score, compared to the supervised LSTM algorithm. Our proposed method also outperformed the advanced SL-NMT approach [26].

Table 1. BLEU score on test data.

Algorithm	BLEU-2	BLEU-3	BLEU-4
MLE-LSTM	0.491	0.383	0.197
SL-NMT [26]	0.603	0.394	0.205
Our proposed modified Seq-GAN	0.645	0.442	0.214

Table 2 illustrates some examples of the sentences generated by the proposed model. In our approach, the discriminator gives high scores to the sentences that have a large number of grammatical errors or to the semantically incomprehensive sentences. Though sentence number 1 has a meaning, it has quite a few grammatical errors. Therefore, the discriminator gave it a high score (almost equal to 1). The sentence has errors such as the nonconformity of space, half-space, and extra space before the dot in “دراین”, “درزمینه”, “سیستمهای”, “توزیع شده”, and “ میشود .”.Sentence number 2 not only has semantic errors, but also has many grammatical errors, therefore the discriminator gave it a high score. This sentence has errors such as nonconformity of space, half-space, and extra space before the dot in “نقشها”, “به منظور”, “امکانپذیر”, “خود تطبیق”, “سامانههای”, and “میباشد”. Sentence number 3, similar to sentence number 1, has a large number of grammatical errors; thus, it received a high score from the discriminator. This sentence has errors such as the nonconformity of extra space before the dot and half-space in “استدلال های”, “پاسخهای”, and “.می باشد”. Sentence number 4 (with no semantic meaning and a few grammatical errors) received a low score from the discriminator and was mistakenly classified as a sentence with no errors. This sentence has errors such as the nonconformity of extra space before the comma, half-space, and using “:” instead of “.” in “، منبع”, “یادگیریهای”, and “دارند:”. Note that as our generator is a reinforcement learning agent which updates its weights based on the score received after finishing an episode, the scores given by the discriminator to episodes, whether low or high, will force the generator to correct its weights. Therefore, the output scores are more important than the assigned classes.

Table 2. Examples of generated sentences by our proposed model with the discriminator scores and the assigned classes (0: no error, 1: with error).

	Class	Score	Sentence	Correct English Translation
1	1	0.99999976	دراین فصل به برخی از کارهای مرتبط انجام شده درزمینهسیستمهایتوزیع شده هوشمندپرداختهمیشود .	This chapter discusses some of the related work done in the field of intelligent distributed systems.
2	1	0.9999995	اعمال بخش نیازمندبرایمدیریت نقشها به منظور امکانپذیرکردن خود تطبیق در این نوع سامانههاینوآوریدیگراینتحقیقمیباشد .	The application of the required section to manage roles in order to enable self-adaptation in these types of systems is another innovation of this research.
3	1	0.9999999	هدف ما افزایش تنوع دانش و استدلال های مختلف برایتولیدپاسخهای درست می باشد .	Our goal is to increase the diversity of knowledge and different arguments to produce correct answers.
4	0	2.7418137×10-6	دراینمنبع ،خودسازماندهی در بسیاری از یکدیگر و یادگیریهاییکتصویر وجود دارند:	This sentence does not have any meaning.

5. CONCLUSIONS

In this paper, we developed a modified Seq-GAN to generate sentences with common semantic and grammatical errors in Persian text, in which policy gradient and direct model pretraining were used. Using this approach, we were able to generate the desired sentences with relatively good quality. The pretraining step was very important because, without this step (i.e., assigning random weights to the generator and its adversarial training), it is not possible to generate sentences as real as possible that are indistinguishable by a human observer. The performance of the proposed method was evaluated using the BLEU score. It was found that the proposed method achieves substantial improvements compared to supervised-trained LSTM using MLE. A future direction could be the adoption of newly proposed deep learning models such as transformers [32] in adversarial training to achieve more accurate results.

REFERENCES

[1] V. R. Mirzaeian, H. Kohzadi, and F. Azizmohammadi, "Learning Persian grammar with the aid of an intelligent feedback generator," *Engineering Applications of Artificial Intelligence,* vol. 49, pp. 167-175, 2016.

[2] V. Cherkassky, N. Vassilas, G. L. Brodt, and H. Wechsler, "Conventional and associative memory approaches to automatic spelling correction," *Engineering Applications of Artificial Intelligence,* vol. 5, no. 3, pp. 223-237, 1992.

[3] V. Makarenkov, L. Rokach, and B. Shapira, "Choosing the right word: Using bidirectional LSTM tagger for writing support systems," *Engineering Applications of Artificial Intelligence,* vol. 84, pp. 1-10, 2019.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473,* 2014.

[5] K. Farhadyar, F. Bonofiglio, D. Zoeller, and H. Binder, "Adapting deep generative approaches for getting synthetic data with realistic marginal distributions," *arXiv preprint arXiv:2105.06907,* 2021.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research,* vol. 3, pp. 993-1022, 2003.

[7] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural networks: Tricks of the trade*: Springer, 2012, pp. 599-619.

[8] I. Goodfellow *et al.*, "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.

[9] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681-4690.

[10] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *European conference on computer vision,* 2016: Springer, pp. 597-613.

[11] Y. Liu, Z. Qin, T. Wan, and Z. Luo, "Auto-painter: Cartoon image generation from sketch by using conditional Wasserstein generative adversarial networks," *Neurocomputing,* vol. 311, pp. 78-87, 2018.

[12] Y. Zhang, Z. Gan, and L. Carin, "Generating text via adversarial training," in *NIPS workshop on Adversarial Training,* 2016, vol. 21: academia. edu, pp. 21-32.

[13] S. Semeniuta, A. Severyn, and E. Barth, "A hybrid convolutional variational autoencoder for text generation," *arXiv preprint arXiv:1702.02390,* 2017.

[14] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," *arXiv preprint arXiv:1701.06547,* 2017.

[15] M. Hossam, T. Le, M. Papasimeon, V. Huynh, and D. Phung, "Text Generation with Deep Variational GAN," *arXiv preprint arXiv:2104.13488,* 2021.

[16] X. Chen, P. Jin, Y. Li, J. Zhang, X. Dai, and J. Chen, "Adversarial subsequences for unconditional text generation," *Computer Speech & Language,* vol. 70, p. 101242, 2021.

[17] L. Yu, W. Zhang, J. Wang, and Y. Y. SeqGAN, "Sequence generative adversarial nets with policy gradient. arxiv e-prints, page," *arXiv preprint arXiv:1609.05473,* 2016.

[18] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621,* 2017.

[19] Z. Alyafeai, M. S. AlShaibani, and I. Ahmad, "A survey on transfer learning in natural language processing," *arXiv preprint arXiv:2007.04239,* 2020.

[20] C. Brockett, B. Dolan, and M. Gamon, "Correcting ESL errors using phrasal SMT techniques," in *21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, Sydney, Australia,* 2006.

[21] A. Rozovskaya and D. Roth, "Training paradigms for correcting errors in grammar and usage," in *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics*, 2010, pp. 154-162.

[22] J. Foster and O. Andersen, "Generrate: Generating errors for use in grammatical error detection," in *Proceedings of the fourth workshop on innovative use of nlp for building educational applications,* 2009, pp. 82-90.

[23] K. Imamura, K. Saito, K. Sadamitsu, and H. Nishikawa, "Grammar error correction using pseudo-error sentences and domain adaptation," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers),* 2012, pp. 388-392.

[24] M. Rei, M. Felice, Z. Yuan, and T. Briscoe, "Artificial error generation with machine translation and syntactic patterns," *arXiv preprint arXiv:1707.05236,* 2017.

[25] H. Yannakoudakis, M. Rei, Ø. E. Andersen, and Z. Yuan, "Neural sequence-labelling models for grammatical error correction," in *Proceedings of the 2017 conference on empirical methods in natural language processing,* 2017: Association for Computational Linguistics (ACL), pp. 2795-2806.

[26] S. Kasewa, P. Stenetorp, and S. Riedel, "Wronging a right: Generating better errors to improve grammatical error detection," *arXiv preprint arXiv:1810.00668,* 2018.

[27] M. Lee, H. Shin, D. Lee, and S.-P. Choi, "Korean Grammatical Error Correction Based on Transformer with Copying Mechanisms and Grammatical Noise Implantation Methods," *Sensors,* vol. 21, no. 8, p. 2658, 2021.

[28] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine,* vol. 35, no. 1, pp. 53-65, 2018.

[29] Z. Xu, M. Wilber, C. Fang, A. Hertzmann, and H. Jin, "Learning from multi-domain artistic images for arbitrary style transfer," *arXiv preprint arXiv:1805.09987,* 2018.

[30] M. J. Kusner and J. M. Hernández-Lobato, "Gans for sequences of discrete elements with the gumbel-softmax distribution," *arXiv preprint arXiv:1611.04051,* 2016.

[31] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311-318.

[32] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683,* 2019.