SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks

Ke Wang, Xiaojun Wan
Institute of Computer Science and Technology, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{wangke17, wanxiaojun}@pku.edu.cn

Abstract

Generating texts of different sentiment labels is getting more and more attention in the area of natural language generation. Recently, Generative Adversarial Net (GAN) has shown promising results in text generation. However, the texts generated by GAN usually suffer from the problems of poor quality, lack of diversity and mode collapse. In this paper, we propose a novel framework - SentiGAN, which has multiple generators and one multi-class discriminator, to address the above problems. In our framework, multiple generators are trained simultaneously, aiming at generating texts of different sentiment labels without supervision. We propose a penalty based objective in the generators to force each of them to generate diversified examples of a specific sentiment label. Moreover, the use of multiple generators and one multi-class discriminator can make each generator focus on generating its own examples of a specific sentiment label accurately. Experimental results on four datasets demonstrate that our model consistently outperforms several state-of-the-art text generation methods in the sentiment accuracy and quality of generated texts.

1 Introduction

Emotional intelligence is an important part of artificial intelligence. Automatic understanding and generation of sentimental texts not only make machines more friendly to humans, but also make them look more intelligent. Nowadays sentiment classification on short texts has made good progress. For instance, one of the state-of-the-art sentiment classifiers has achieved an accuracy of 90% on the Stanford Sentiment Treebank dataset [Hu *et al.*, 2016]. But compared with the success of sentiment classification, generic sentimental text generation is challenging and very few recent attempts have been made to investigate it. Previous work has been mostly limited to task-specific applications and just use hidden variables to indirectly control the sentiment labels of generated texts, especially in emotional response generation [Zhou *et al.*, 2017; Zhou and Wang, 2017]. It is difficult to design an appropriate and specific training objective in deep generative models for sentimental text generation. Generative Adversarial Net (GAN) [Goodfellow *et al.*, 2014] is a good solution to this problem which uses a discriminator instead of a specific objective to guide the generator. The main intuition is that since text sentiment classification is very strong, we can use the classifier to guide the generation of sentimental texts.

In this study, we aim to generate a variety of high-quality sentimental texts using GAN. That is, in the absence of parallel corpus, we can automatically generate a variety of controllable sentimental texts without supervision. However, there are a few challenges to be addressed when applying GAN to generate sentimental texts. Firstly, the discrete nature of texts leads to a sampling step that is not differentiable, which makes it impossible for the gradient to pass from the discriminator to the generator. Recently, some studies use reinforcement learning which treats the process of discriminator guiding generator as a reinforcement learning policy [Yu *et al.*, 2017; Guo *et al.*, 2017]. Even though, the generated texts face the problem of poor quality. Secondly, one of the major drawbacks of GAN is the problem of "mode collapse", and it has been empirically proven that GAN prefers to generate samples around only a few modes whilst ignoring other modes [Theis *et al.*, 2016]. So there is a lack of diversity in generated texts.

We propose a new text generation framework - SentiGAN to address the above issues and generate texts of different sentiment labels. SentiGAN consists of multiple generators and one multi-class discriminator, which are trained simultaneously. Like [Yu *et al.*, 2017], we consider the sequence generation procedure as a sequential decision making process. We regard each generative model as a stochastic parametrized policy and use Monte Carlo search to approximate the state-action value. Then we use the discriminator to evaluate the sequence and guide the learning of the generative model. But unlike previous works, our model contains multiple generators and one discriminator. Firstly, we propose a novel penalty based objective, which adopts a more reasonable measure and aims to minimize overall penalties instead of maximizing rewards. It is proved both experimentally and theoretically that, our penalty based objective can force each generator to generate diversified texts of a specific sentiment label, rather than generating examples which are repetitive but "safe" and "good". Secondly, the use of our discriminator's multi-class classification objective can makes

generators more focused on generating their own examples of specific sentiment labels, and stay away from other types of sentiments. This improves the sentiment accuracy of the generated texts.

We use a well-performed sentiment classifier as evaluator to verify the sentiment accuracy of the generated texts, as well as several other metrics (i.e., fluency, novelty, diversity, intelligibility) to measure the quality of generated texts from different aspects. We compare our model with several state-of-the-art deep generative models, including RNNLM [Mikolov *et al.*, 2011], GAN and its variants [Yu *et al.*, 2017; Lin *et al.*, 2017; Guo *et al.*, 2017], VAEs [Kingma and Welling, 2014; Kingma *et al.*, 2014]. Experimental results on four datasets (i.e., movie reviews, beer reviews, custom reviews and a synthetic dataset) demonstrate that our model consistently outperforms the existing models in both the sentiment accuracy and quality of generated texts.

The major contributions of this paper are summarized as follows:

1) We propose a novel framework SentiGAN to generate generic, diversified and high-quality sentimental texts of different sentiment labels.
2) We propose a new penalty based objective to make each generator in SentiGAN produce diversified texts of a specific sentiment label.
3) Extensive experiments are performed on four datasets and the results demonstrate the efficacy and superiority of our proposed model.

2 Related Work

Unsupervised text generation is an important research area in natural language processing. A standard recurrent neural network language model [Mikolov *et al.*, 2011] predicts each word of a sentence conditioned on the previous word and an evolving hidden state. However, it suffers from two major drawbacks when used to generate texts. First, RNN based models are always trained through the maximum likelihood approach, which suffers from exposure bias [Bengio *et al.*, 2015]. Second, the loss function used to train the model is at word level but the performance is typically evaluated at sentence level. There are some researches which use generative adversarial network (GAN) to solve the problems.

Generative Adversarial Nets (GANs) [Goodfellow *et al.*, 2014] are a recent novel class of deep generative models. Though GANs achieve great successes on computer vision applications [Denton *et al.*, 2015; Isola *et al.*, 2016; Salimans *et al.*, 2016], there are only a few progresses in natural language generation because the discrete sequences are not differentiable. Some works attempt to solve this problem, including Gumbel-softmax distribution [Kusner and Hernández-Lobato, 2016], Professor Forcing [Lamb *et al.*, 2016] and so on. However, it is more common to tackle the non-differentiable problem with a strategic gradient of reinforcement learning, including SeqGAN [Yu *et al.*, 2017], RankGAN [Lin *et al.*, 2017], LeakGAN [Guo *et al.*, 2017]. The effects of these variants of GANs are not very different, and none of these methods can generate samples with diverse attributes. Without loss of generality, we will focus on comparison with SeqGAN in this study. Conditional GAN [Mirza and Osindero, 2014] is a variant of GAN that produces controlled samples which uses a condition variable to guide the generation. This is also one of our main comparisons. LabelGAN [Salimans *et al.*, 2016] uses a discriminator to identify multiple categories which is similar to us, but it has only one generator and does not solving discrete problems in text generation. Other superior unsupervised deep generative models include Variational Autoencoders (VAE) [Kingma and Welling, 2014], semi-supervised VAE (S-VAE) [Kingma *et al.*, 2014]. VAE consists of encoder and generator networks which encode a data example to a latent representation and generate samples from the latent space, respectively. Although VAE does not have the problem of generating discrete data, it has more constraints and restrictions than GAN, and we will also compare our model with it in the experiments.

Other related work includes product review generation conditioned on specific inputs (e.g., user, product, aspect, ratings) [Dong *et al.*, 2017; Zang and Wan, 2017; Lipton *et al.*, 2015]. These methods usually need a large parallel corpus for learning an encoder-decoder model. Different from these studies, we aim to generate a variety of generic reviews of different sentiment labels.

3 SentiGAN

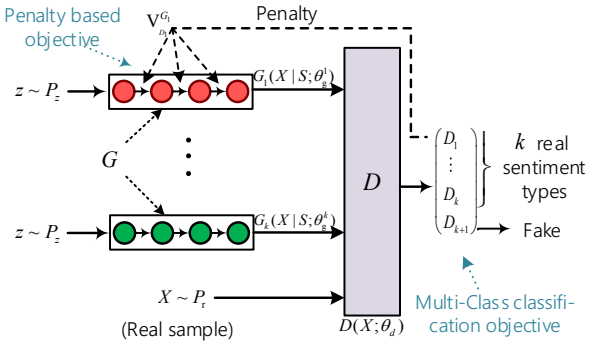

Figure 1: The framework of SentiGAN with k generators and one multi-class discriminator.

3.1 Overall Framework

The framework of our proposed SentiGAN is shown in Figure 1. Supposing we want to generate texts with k types of sentiments(i.e., k sentiment labels), we use k generators $\{G_i(X|S;\theta_g^i)\}_{i=1}^{i=k}$ and one discriminator $D(X;\theta_d)$, where θ_g^i, θ_d are the parameters of the i-th generator and the discriminator respectively, and the prior input noise z sampled from the distribution P_z (e.g., a normal distribution) is used to initialize the generator's input.

The whole framework can be divided into two adversarial learning objectives: generator learning and discriminator learning. The goal of the i-th generator G_i is to generate texts with the i-th sentiment type that can deceive the discriminator. Specifically, it aims to minimize the penalty based objective that we propose. Instead, the goal of the discrimina-

tor is to distinguish between fake texts (texts generated by generators) and real texts with k sentiment types as much as possible, which is the multi-class classification objective we adopt.

Without loss of generality, we set k to 2 in the experiments and let SentiGAN generate two types of sentimental texts (positive and negative).

Generator Learning

To solve the problem that the gradient cannot pass back to the generative model when the output is discrete, we formalize the text generation problem as a sequential decision making process [Bachman and Precup, 2015]. That is, at each timestep t, we train a generator G_i to produce a sequence $X_{1:t} = \{X_1, ..., X_t\}$, where X_t represents a word token in the given vocabulary C. $G_i(X_{t+1}|S_t; \theta_g^i)$ means the probability that selecting the $(t+1)$-th word depends on the previously generated words (its current state), denoted as $S_t = \{X_1, ..., X_t\}$. And we define a new penalty based loss function $L(X)$:

$$L(X) = G_i(X_{t+1}|S_t; \theta_g^i) \cdot V_{D_i}^{G_i}(S_t, X_{t+1}) \tag{1}$$

where $V_{D_i}^{G_i}(S_t, X_{t+1})$ is the penalty for a sequence $X_{1:t+1}$ which is calculated by the discriminator. Finally, the objective of the i-th generator $G_i(X|S; \theta_g^i)$ is to minimize the total penalty based value:

$$\begin{aligned} J_{G_i}(\theta_g^i) &= \mathbb{E}_{X \sim P_{g_i}}[L(X)] \\ &= \sum_{t=0}^{t=|X|-1} G_i(X_{t+1}|S_t; \theta_g^i) \cdot V_{D_i}^{G_i}(S_t, X_{t+1}) \end{aligned} \tag{2}$$

where $X_t \in C$. Because the discriminator can only judge on a complete sequence, we apply Monte Carlo search with roll-out policy G_i to sample the unknown last $|X| - t$ tokens. Thus, Our penalty function for the i-th generator is calculated as:

$$V_{D_i}^{G_i}(S_{t-1}, X_t) = \begin{cases} \frac{1}{N} \sum_{n=1}^{N} (1 - D_i(X_{1:t}^n; \theta_d)) & t < |X| \\ 1 - D_i(X_{1:t}; \theta_d) & t = |X| \end{cases} \tag{3}$$

where $X_{1:t}^n$ is N-time Monte Carlo search sampled based on the roll-out policy G_i and the current state, and $D_i(X_{1:t}^n; \theta_d)$ is the sentence probability given by the discriminator that $X_{1:t}^n$ is the real i-th type sentimental text.

In addition, our generator here is a simple layer of Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] which outputs the t-th word according to the distribution:

$$p(X_t) = softmax(LSTM_{\theta_g}(h_{t-1}, X_{t-1})) \tag{4}$$

where the parameters of the $LSTM_{\theta_g}$ is θ_g, and h_t is the hidden state of timestep t. It is worth noticing that our generator can be easily extended to other types of generators as well.

Discriminator Learning

Inspired by the discriminator formulation for semi-supervised learning [Salimans *et al.*, 2016], we use a multi-class classification objective that requires the discriminator to distinguish between the real texts with each sentiment type and the generated texts. In more detail, given the set of k generators, the discriminator produces a softmax probability distribution over $k+1$ classes. The score at i-th ($i \in \{1, .., k\}$) index (D_i) represents the probability that it belongs to the real i-th type sentimental texts, and the score at $(k+1)$-th index represents the probability that the sample is generated by generators. The objective function of the discriminator is to minimize:

$$\begin{aligned} J_D(\theta_d) = &- \mathbb{E}_{X \sim P_g} log D_{k+1}(X; \theta_d) \\ &- \sum_{i=1}^{k} \mathbb{E}_{X \sim P_{r_i}} log D_i(X; \theta_d) \end{aligned} \tag{5}$$

where P_g is texts generated by all generators, P_{r_i} is real i-th type sentimental texts, and $D_i(X; \theta_d)$ represents the i-th index of $D(X; \theta_d)$. Since CNN has recently been shown of great effectiveness in text classification [Zhang and Lecun, 2015], our discriminator here is a layer of CNN which has multiple filters.

We perform the adversarial training of generators and discriminator, and train them alternately, as shown in Algorithm 1.

Algorithm 1 The adversarial training process in SentiGAN

Input: Input noise, z; Generators, $\{G_i(X|S; \theta_g^i)\}_{i=1}^{i=k}$; Discriminator, $D(X; \theta_d)$; Real text dataset with k types of sentiment, $T = \{T_1, ..., T_k\}$;
Output: Well trained generators, $\{G_i(X|S; \theta_g^i)\}_{i=1}^{i=k}$;

```
Initialize {G_i}_{i=1}^{i=k}, D with random weights;
Pre-train {G_i}_{i=1}^{i=k} using MLE on T;
Generate fake texts F = {F_i}_{i=1}^{i=k} using {G_i}_{i=1}^{i=k};
Pre-train D(X; θ_d) using {T_1, ..., T_k, F};
repeat
    for g-steps do
        for i in 1 ~ k do
            Generate fake texts using G_i(z; θ_g^i);
            Calculate penalty V_{D_i}^{G_i} by Eq (3) ;
            Update G_i(z; θ_g^i) by minimizing Eq (2);
        end for
    end for
    for d-steps do
        Generate fake texts F = {F_i}_{i=1}^{i=k} using {G_i(X|S; θ_g^i)}_{i=1}^{i=k};
        Update D(X; θ_d) using {T_1, ..., T_k, F} by minimizing Eq (5);
    end for
until SentiGAN converges
return ;
```

3.2 The Multi-Class Classification Objective

Here we introduce how our multi-class classification objective can force each generator to focus more on generating its own sentimental texts that are far from sentimental texts generated by other generators, thus it helps improve the sentiment accuracy of the generated texts.

Firstly, the optimal i-th generator can learn the distribution of the real texts with the i-th sentiment. The objective func-

tion of the discriminator is show in Eq (5). Using $\sum_{i=1}^{k+1} D_i = 1$, $D_i \in [0,1], \forall i$, we obtain the optimal distribution learned by the discriminator D: $D_i(X;\theta_d) = \frac{P_{r_i}(X)}{\sum_{i=1}^{k} P_{r_i}(X)+P_g(X)}$, $i = \{1,...k\}$, $D_{k+1}(X;\theta_d) = \frac{P_g(X)}{\sum_{i=1}^{k} P_{r_i}(X)+P_g(X)}$. Then by using Eq (5), generators' goal is to minimize the following:

$$\begin{aligned}
&\mathbb{E}_{X\sim P_g} log[\frac{P_g(X)}{P_{avg}(X)}] + \sum_{i=1}^{k} \mathbb{E}_{X\sim P_{r_i}} log[\frac{P_{r_i}(X)}{P_{avg}(X)}] \\
&- (k+1)log(k+1) \\
=&KL(\sum_{i=1}^{k} P_{g_i}(X)||P_{avg}(X)) + \sum_{i=1}^{k} KL(P_{r_i}(X)||P_{avg}(X)) \\
&- (k+1)log(k+1),
\end{aligned} \tag{6}$$

where $P_{avg}(x) = \frac{\sum_{i=1}^{k} P_{r_i}(X)+P_g(X)}{k+1}$, KL means Kullback–Leibler divergence. The above objective obtains its global minimum if $P_{g_i} = P_{r_i}, (i = 1, ..., k)$ with the objective value of $-(k+1)log(k+1)$. In the case of one generator $(k = 1)$, Eq (6) obtains the Jensen-Shannon divergence (JS) with the minimum objective value of $-log4$.

Secondly, while keeping θ_d constant, the i-th generator aims to minimize the penalty $(V_{D_i}^{G_i})$ given by the discriminator. Under the setting of $\sum_{i=1}^{k+1} D_i = 1$, it is equivalent to minimize $\sum_{j=1,j\neq i}^{k+1} D_j(X;\theta_d)$. Intuitively, in order to get lower penalties from the discriminator, the texts generated by the i-th generator must be more consistent with the i-th sentiment type and be far away from other sentiment types.

3.3 The Penalty-Based Objective

Here we introduce how the penalty based objective forces generators to generate diversified examples rather than generate repetitive and "safe" samples, thus it helps improve the diversity and quality of generated texts. We compare the generator's objective function of GAN, SeqGAN and our SentiGAN as follows:

$$J_G(X) = \begin{cases} \mathbb{E}_{X\sim P_g}[-log(D(X;\theta_d))] & \text{GAN} \\ \mathbb{E}_{X\sim P_g}[-log(G(X|S;\theta g)D(X;\theta_d))] & \text{SeqGAN} \\ \mathbb{E}_{X\sim P_g}[G(X|S;\theta_g)V(X)] & \textbf{SentiGAN} \end{cases} \tag{7}$$

As can be seen, there are two main improvements in our objective function.

Firstly, our penalty based objective can be considered as a measure of wasserstein distance [Arjovsky *et al.*, 2017] which always provides meaningful gradients, even when the distributions of P_r and P_g do not overlap. But KL and JS can not do it. The wasserstein distance of the two distributions is:

$$W(P_r, P_g) = \frac{1}{K} sup_{||L||_L \leq K} \mathbb{E}_{X\sim P_r}[L(X)] - \mathbb{E}_{X\sim P_g}[L(X)]. \tag{8}$$

where function $L(X)$ is needed to satisfy Lipschitz continuity and its Lipschitz constant is K. Since the derivative of $log(x)$ has no upper bound, it does not satisfy Lipschitz continuity, that is, it can not be used as $L(X)$ here. In this paper, we define $L(X)$ as Eq(1).

Secondly, we use penalty $V(X)$ instead of reward $D(X)$ like SeqGAN. Our penalty-based loss function $G(X|S;\theta_g)V(X)$ can be thought of as adding $G(X|S;\theta_g)$ to the reward-based loss function $(-G(X|S;\theta_g)D(X;\theta_d))$.

$$\begin{aligned}
G(X|S;\theta_g)V(X) =& G(X|S;\theta_g)(1 - D(X;\theta_d) \\
=& G(X|S;\theta_g) - G(X|S;\theta_g)D(X;\theta_d)
\end{aligned} \tag{9}$$

Therefore, our penalty-based loss function forces the generator to prefer a smaller $G(X|S;\theta_g)$. Thus it results in the generation of diversified samples, rather than repetitive but "good" samples.

4 Experiments

4.1 Experiment Setup

The problem of generating very long texts is very challenging in the text generation area and we will study this problem in our future work. Therefore, we simply refer to the work of [Hu *et al.*, 2017] and focus on generating short sentences (length ≤ 15 words) of two sentiment types (positive and negative) on three real datasets.

Movie Reviews (MR). We use Stanford Sentiment Treebank [Socher *et al.*, 2013] which has two sentiment classes (negative and positive). The original dataset has a total of 9613 sentences. We select sentences containing at most 15 words, and the resulting dataset contains 2133 positive sentences and 2370 negative sentences.

Beer Reviews (BR). We use the data scraped from BeerAdvocate [Mcauley and Leskovec, 2013]. BeerAdvocate is a large online review community boasting 1,586,614 reviews of 66,051 distinct items composed by 33,387 users. Each review is accompanied by a number of numerical ratings, corresponding to "appearance", "aroma", "palate", "taste", and also the user's "overall" impression. We select sentences containing at most 15 words, and the resulting dataset contains 1437767 positive sentences and 11202 negative sentences.

Customer Reviews (CR). We use customer reviews of various products [Hu and Liu, 2004]. We select sentences containing at most 15 words, and the resulting dataset contains 1024 positive sentences and 501 negative sentences.

We train our model on each dataset respectively, and randomly initialize word embeddings with the dimension size of 300. The generators are set as single-layer LSTM-RNNs with input/hidden dimension size of 300 and max sample length of 15 words. The CNN in our discriminator is the same as [Zhang and Lecun, 2015]. The N in Monte Carlo search is set as 15. In the per-training step, we pre-train generators for 120 steps, pre-train the discriminator for 50 steps. In adversarial training, the g-steps is 5 and d-steps is 1. The optimization algorithm is RMSProp. We implement our model based on Tensorflow and use a TITAN X graphic card for learning.

4.2 Sentiment Accuracy of Generated Texts

We use a state-of-the-art sentiment classifier [Hu *et al.*, 2016] which achieves an accuracy of 90% on the SST test set, to automatically evaluate the sentiment accuracy of the generated

Accuracy	MR	BR	CR
Real Data	0.892	0.874	0.846
RNNLM	0.622	0.595	0.552
SeqGAN	0.717	0.684	0.632
VAE	0.751	0.721	0.643
SentiGAN(k=1)	0.803	0.750	0.731
C-GAN	0.822	0.773	0.762
S-VAE	0.831	0.793	0.727
SentiGAN(k=2)	**0.885**	**0.841**	**0.803**

Table 1: Comparison of sentiment accuracy of generated sentences. The real data is the training corpus.

texts. We compare with several state-of-art generic text generation methods, including **RNNLM** [Mikolov *et al.*, 2011], **SeqGAN** [Yu *et al.*, 2017], **Variational Autoencoders**(VAE) [Kingma and Welling, 2014], **Conditional GAN**(C-GAN) [Mirza and Osindero, 2014] and **Semi-supervised VAE**(S-VAE) [Kingma *et al.*, 2014]. It is worth noting that pre-training was used for all selected baselines.

We use each model to generate 5K positive sentences and 5K negative sentences, which is trained on each of the above three datasets, respectively. The results are shown in Table 1. In order to investigate whether it is better to train with multiple generators than a single generator, we made a comparison with SentiGAN(k=1). Note that RNNLM, SeqGAN, SentiGAN(k=1) can not generate texts with two sentiment labels simultaneously, so we train each of these models on positive reviews and negative reviews, respectively.

From the comparison results in Table 1 we can see that our proposed model (SentiGAN(k=2)) outperforms all other methods, including C-GAN and S-VAE. The accuracy achieved by our model is promisingly high, indicating that the framework with mixture of generators and one multi-class discriminator can make each generator to generate their own sentimental texts better. What's more, comparing SentiGAN(k=2) with SentiGAN(k=1) shows that multiple generators can help each other and thus greatly improve the sentiment accuracy of texts generated by each single generator. In addition, our model remains the leading result, even on the small CR dataset.

4.3 Quality of Generated Sentences

Further, we use four other evaluation metrics to measure the quality of generated sentences from various aspects.

Fluency: We use a language modeling toolkit - SRILM [Stolcke, 2002] to test the fluency of generated sentences. SRILM calculates the perplexity of generated sentences using the language model trained on respective corpus. The results are shown in Figure 2. We can see that C-GAN and S-VAE are not good at keeping the fluency of sentences. However, our model maintains good fluency while generating texts of different sentiment labels, and it even significantly outperforms the existing models on the small CR dataset.

Novelty: We want to investigate how different the generated sentences and the training corpus are. In other words, we want to see if the generator simply copies the sentence in the corpus instead of generating new ones. We calculate the

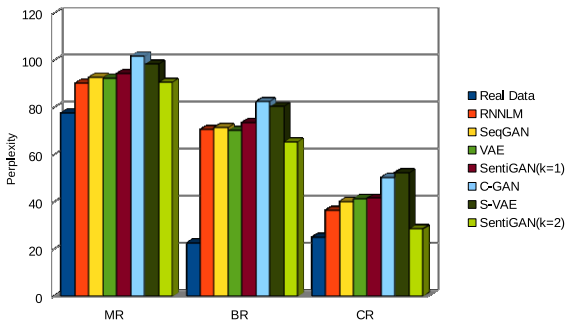

Figure 2: Comparison of fluency (Perplexity) of generated sentences (Lower perplexity means better fluency).

Methods	MR	BR	CR
RNNLM	0.267	0.283	0.399
SeqGAN	0.298	0.328	0.437
VAE	0.287	0.347	0.417
SentiGAN(k=1)	0.344	0.409	0.479
C-GAN	0.368	0.398	0.482
S-VAE	0.328	0.369	0.437
SentiGAN(k=2)	**0.395**	**0.427**	**0.549**

Table 2: Comparison of the novelty of generated sentences.

novelty of each generated sentence S_i as follows:

$$Novelty(S_i) = 1 - \max\{\varphi(S_i, C_j)\}_{j=1}^{j=|C|} \quad (10)$$

where C is the sentence set of the training corpus, φ is Jaccard similarity function. The average values over generated sentences are shown in Table 2, we can see that RNNLM, SeqGAN and VAE are not good at generating new sentences. On the contrary, our model performs exceptionally well, with the ability to generate sentences different from that in the training corpus.

Diversity: We want to see if the generator can produce a variety of sentences. Given a collection of generated sentences S, we define the diversity of sentences S_i as follows:

$$Diversity(S_i) = 1 - \max\{\varphi(S_i, S_j)\}_{j=1}^{j=|S|, j\neq i} \quad (11)$$

where φ is the Jaccard similarity function. We calculate the maximum Jaccard similarity between each sentence S_i and other sentences in the collection. The average values are shown in Table 3, and we can see that our model can generate a variety of sentences, while other models can not ensure the diversity of generated sentences.

Methods	MR	BR	CR
Real Data	0.753	0.705	0.741
RNNLM	0.691	0.677	0.663
SeqGAN	0.641	0.636	0.619
VAE	0.661	0.658	0.620
SentiGAN(k=1)	0.711	0.687	0.668
C-GAN	0.726	0.688	0.680
S-VAE	0.692	0.687	0.649
SentiGAN(k=2)	**0.741**	**0.713**	**0.708**

Table 3: Comparison of the diversity of generated sentences.

	SentiGAN(k=2)	C-GAN
Positive	a fantastic finally , simply perfect masterpiece.	give it credit , this is our 's brilliant . ***(Unreadable)***
	one of the greatest movies i have ever seen.	good , bloody fun movie
	funny and entertaining , just an emotionally idea but it was pretty good.	makes me smile every time to get on alien . ***(Unreadable)***
	the best comedy is a science fiction , captain is like a comic legend.	powerfully moving ! ***(Very short)***
Negative	one of the most disturbing and sickening movies i have ever seen.	very bad comedy. ***(Very short)***
	a story which fails to rise above its disgusting source material .	a mere shadow of its predecessors
	the comedy is nonexistent .	a timeless classic western dog ... ***(Wrong sentiment)***
	this is a truly bad movie .	one of those history movie traps

Table 4: Examples sentences generated by SentiGAN and Conditional GAN trained on MR.

Intelligibility: We use human evaluation for evaluating the intelligibility of generated sentences. We randomly extract 100 sentences from the generated sentences and then ask three graduate students to rate each of them according to its intelligibility. The rating score ranges from 1 to 5, and 5 is the best. We finally take the average score across the sentences and the three students, as shown in Figure 3. We can see that our model performs better than all other methods and our model can generate sentimental sentences with best intelligibility. Moreover, comparing the results on different datasets, we can see that more data can train better models with respect to intelligibility ($CR < MR < BR$).

4.4 Validation of Penalty-Based Objective

Here we use a synthetic data set to test our proposed model in the mere use of the penalty based objective (i.e., SentiGAN(k=1)). The synthetic data[1] consists of a set of sequential tokens which can be seen as the simulated data comparing to the real-word language data. We use the oracle model[1] to generate 10, 000 sequences as the training set. We compare our model with various published methods (SeqGAN [Yu *et al.*, 2017], RankGAN [Lin *et al.*, 2017]) on this dataset, as shown in Table 5. And the learning curves are shown in Figure 4. The results show the effectiveness of using penalty-based objective, and our model is better than the other models in capturing the dependency of the sequential tokens.

[1]The synthetic data and the oracle model (LSTM model) are publicly available at https://github.com/LantaoYu/SeqGAN

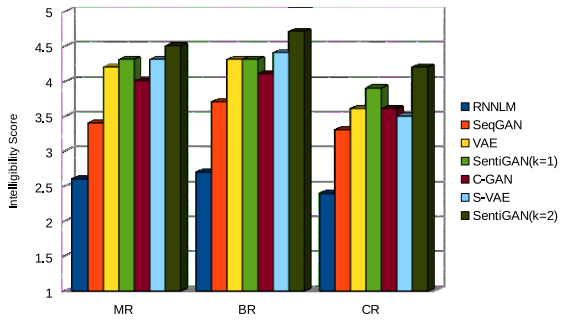

Figure 3: Comparison of intelligibility of generated sentences by human evaluation.

Method	MLE	SeqGAN	RankGAN	SentiGAN(k=1)
NLL	9.038	8.736	8.247	**6.924**

Table 5: The performance comparison of different methods on the synthetic data in terms of the negative log-likelihood (NLL) scores.

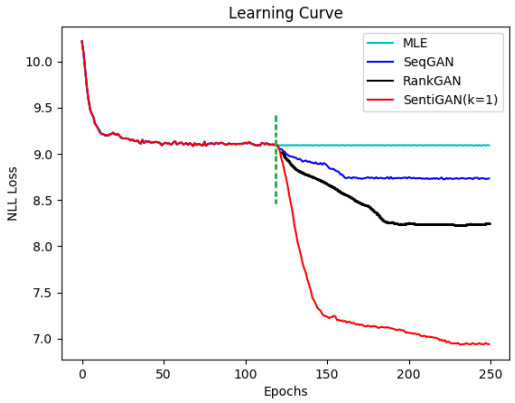

Figure 4: The illustration of learning curves. Dotted line is the end of pre-training.

4.5 Case Study

In Table 4, we show example sentences generated by SentiGAN(k=2) and C-GAN trained on the MR dataset. From the examples, we can see some problems(e.g., unreadable, very short, wrong sentiment) with the sentences generated by C-GAN. Whereas, our proposed model produces sentences that are more readable, sentimentally accurate, with better quality, and longer than that of C-GAN.

5 Conclusion and Future Work

In this paper, we propose SentiGAN, which can generate a variety of high-quality texts of different sentiment labels. Extensive experiments demonstrate the efficacy of SentiGAN. In future work, we will make use of more complex and sophisticated generators to enhance the quality of generated texts, especially for longer text generation. We will also apply our model to generate texts with other kinds of labels (e.g., different writing styles).

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772036, 61331011) and Key Laboratory

of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We thank the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

[Arjovsky *et al.*, 2017] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *CoRR*, abs/1701.07875, 2017.

[Bachman and Precup, 2015] Philip Bachman and Doina Precup. Data generation as sequential decision making. *NIPS*, 2015.

[Bengio *et al.*, 2015] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*, pages 1171–1179, 2015.

[Denton *et al.*, 2015] Emily L Denton, Soumith Chintala, Arthur Szlam, and Robert D Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *NIPS*, pages 1486–1494, 2015.

[Dong *et al.*, 2017] Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. Learning to generate product reviews from attributes. In *EACL*, volume 1, pages 623–632, 2017.

[Goodfellow *et al.*, 2014] Ian J Goodfellow, Jean Pougetabadie, Mehdi Mirza, Bing Xu, David Wardefarley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv: Machine Learning*, 2014.

[Guo *et al.*, 2017] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. Long text generation via adversarial training with leaked information. *CoRR*, 2017.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[Hu and Liu, 2004] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. pages 168–177, 2004.

[Hu *et al.*, 2016] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard H Hovy, and Eric P Xing. Harnessing deep neural networks with logic rules. *ACL*, pages 2410–2420, 2016.

[Hu *et al.*, 2017] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Toward controlled generation of text. In *ICML*, pages 1587–1596, 2017.

[Isola *et al.*, 2016] Phillip Isola, Junyan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2016.

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

[Kingma *et al.*, 2014] Diederik P Kingma, Danilo J Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. *NIPS*, 4:3581–3589, 2014.

[Kusner and Hernández-Lobato, 2016] Matt J. Kusner and José Miguel Hernández-Lobato. GANS for sequences of discrete elements with the gumbel-softmax distribution. *CoRR*, abs/1611.04051, 2016.

[Lamb *et al.*, 2016] Alex Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. *NIPS*, pages 4601–4609, 2016.

[Lin *et al.*, 2017] Kevin Lin, Dianqi Li, Xiaodong He, Ming-Ting Sun, and Zhengyou Zhang. Adversarial ranking for language generation. In *NIPS*, pages 3158–3168, 2017.

[Lipton *et al.*, 2015] Zachary C Lipton, Sharad Vikram, and Julian McAuley. Generative concatenative nets jointly learn to write and classify reviews. *arXiv preprint arXiv:1511.03683*, 2015.

[Mcauley and Leskovec, 2013] Julian Mcauley and Jure Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. *WWW*, pages 897–908, 2013.

[Mikolov *et al.*, 2011] Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. pages 5528–5531, 2011.

[Mirza and Osindero, 2014] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv: Learning*, 2014.

[Salimans *et al.*, 2016] Tim Salimans, Ian J Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NIPS*, pages 2234–2242, 2016.

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. 2013.

[Stolcke, 2002] Andreas Stolcke. Srilm – an extensible language modeling toolkit. 2002.

[Theis *et al.*, 2016] Lucas Theis, Aaron Van Den Oord, and Matthias Bethge. A note on the evaluation of generative models. *ICLR*, 2016.

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

[Zang and Wan, 2017] Hongyu Zang and Xiaojun Wan. Towards automatic generation of product reviews from aspect-sentiment scores. In *INLG*, pages 168–177, 2017.

[Zhang and Lecun, 2015] Xiang Zhang and Yann Lecun. Text understanding from scratch. *arXiv: Learning*, 2015.

[Zhou and Wang, 2017] Xianda Zhou and William Yang Wang. Mojitalk: Generating emotional responses at scale. *CoRR*, abs/1711.04090, 2017.

[Zhou *et al.*, 2017] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. Emotional chatting machine: Emotional conversation generation with internal and external memory. *CoRR*, abs/1704.01074, 2017.