Differentiated Distribution Recovery for Neural Text Generation

Jianing Li,† Yanyan Lan,†,* Jiafeng Guo,† Jun Xu,† Xueqi Cheng†
†CAS Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
†University of Chinese Academy of Sciences, Beijing, China
∗Department of Statistics, University of California, Berkeley
{lijianing,lanyanyan,guojiafeng,cxq}@ict.ac.cn, junxu@ruc.edu.cn

Abstract

Neural language models based on recurrent neural networks (RNNLM) have significantly improved the performance for text generation, yet the quality of generated text represented by Turing Test pass rate is still far from satisfying. Some researchers propose to use adversarial training or reinforcement learning to promote the quality, however, such methods usually introduce great challenges in the training and parameter tuning processes. Through our analysis, we find the problem of RNNLM comes from the usage of maximum likelihood estimation (MLE) as the objective function, which requires the generated distribution to precisely recover the true distribution. Such requirement favors high generation diversity which restricted the generation quality. This is not suitable when the overall quality is low, since high generation diversity usually indicates lot of errors rather than diverse good samples. In this paper, we propose to achieve differentiated distribution recovery, DDR for short. The key idea is to make the optimal generation probability proportional to the β-th power of the true probability, where $\beta > 1$. In this way, the generation quality can be greatly improved by sacrificing diversity from noises and rare patterns. Experiments on synthetic data and two public text datasets show that our DDR method achieves more flexible quality-diversity trade-off and higher Turing Test pass rate, as compared with baseline methods including RNNLM, SeqGAN and LeakGAN.

1 Introduction

The ability of generating fluent, grammatical, and logical text which can pass the Turing Test is crucial for many natural language processing (NLP) tasks, such as machine writing (Zhang et al. 2017a), machine translation (Bahdanau, Cho, and Bengio 2014), dialogue generation (Li et al. 2017), and image captioning (Rennie et al. 2017). This paper focuses mainly on unconditional text generation, which is principle for the text generation task (Yu et al. 2017; Guo et al. 2017; Fedus, Goodfellow, and Dai 2018).

Recently, neural generation models have gained great attention in this area. The typical method is to apply recurrent neural networks based language models (RNNLM) (Mikolov et al. 2010), in which the model is trained using maximum likelihood estimation (MLE). However, this objective function is insufficient for generating high-quality text, as shown in (Huszár 2015; Arjovsky and Bottou 2017). The reason is that MLE is equivalent to minimizing KL divergence, which will force the generation probability to be high for all training instances. As a result, RNNLM trained with MLE favors high generation diversity and is easy to be influenced by noises or rare patterns in the training data, which harms its generation quality. What's worse, such high diversity usually comes from lots of generation errors, rather than diverse good samples as expected. Beam Search can be applied to guarantee high generation quality by only picking the sample with highest generation probability, but significantly reduces the generation diversity and is not applicable where far more than one samples need to be generated. How to achieve high generation quality while making acceptable trade-off with diversity is still an open problem.

In real applications, generation quality usually holds much higher priority than diversity, since users are quite sensitive to logical or grammatical errors but less sensitive to relatively low diversity. According to our observation, the Turing Test pass rates for generation models trained with MLE are usually far from satisfying. There are many works focusing on achieving higher generation quality, among which adversarial training based methods are most successful(Yu et al. 2017; Lin et al. 2017; Guo et al. 2017; Zhang et al. 2017b; Fedus, Goodfellow, and Dai 2018). While making a big progress in the generation quality, the training with adversarial mechanism (Goodfellow et al. 2014) is notoriously hard to tune, which easily leads to unstable training(Arjovsky and Bottou 2017). Some of the methods also introduce reinforcement learning as a solution for gradient passing difficulties(Yu et al. 2017; Lin et al. 2017; Guo et al. 2017), which greatly slows down the training process as compared with MLE-based methods and makes the parameter tuning much harder. Further more, such methods are also observed to cause decrease in generation diversity while promoting the quality(Lu et al. 2018), but the trade-off can hardly be controlled.

In this paper, we propose a new approach to achieve flexible quality-diversity trade-off and higher Turing Test pass rate, without the costly adversarial training reinforcement learning process. Firstly, we find that the problem of MLE comes from the theoretical fact that the optimal generated distribution Q in MLE equals to the true probability $P(x)$, namely precise distribution recovery. We can see that

this rigorous requirement favors high generation probability over all training instances, thus is easily to be influenced by the noises or rare patterns in the training data. This paper aims at achieving a differentiate distribution recovery instead, namely DDR. The key idea is to make the optimal generation probability proportional to the β-th power of the true probability with normalization, where $\beta > 1$. In this way, significant patterns and rare patterns will be differentiated more clearly, and the influence of noises or rare patterns can be alleviated. Under a generalized form of objective functions $\mathbb{E}_{x\sim P}\, f[Q(x)]$, we find a family of functions $f(z) = \alpha z^{1/\alpha} - \alpha, \alpha > 1$, to satisfy our requirements with $\alpha = \frac{\beta}{\beta-1}$. In this case, the optimal generated probability can be represented as $Q^*(x) = P(x)^\beta / \sum_x P(x)^\beta$. More interestingly, if α goes to infinity, the function f becomes the log function used in MLE, i.e. $f(z) = \ln z$. Therefore, this function acts as a good substitution for the MLE as the objective function.

Compared with SeqGAN (Yu et al. 2017) and LeakGAN (Guo et al. 2017), models in our DDR method can be trained end to end similarly as RNNLM, without costly adversarial training or reinforcement learning process. We conduct experiments on both a synthetic dataset and two real-world public datasets. We observe that the generation quality gradually increases as we lowering α on all datasets, indicating quality improvement. While achieving competitive generation quality with SeqGAN and LeakGAN at the same diversity level, DDR can achieve higher quality by further lowering α, showing flexible trade-off control. Human evaluation results gives accordant results, where DDR achieves higher Turing Test pass rate. In addition, we also show the robustness of DDR by adding noise in training data. While RNNLM starts to generate bad samples, our method keeps its ability of generating high-quality samples.

2 Related Work

Generating high-quality text has been a critical challenge for the NLP community. Since the success application of deep learning, neural text generation models have been developed rapidly, and gained some improvements. Among these neural text generation models, the most typical one is RNNLM proposed by (Mikolov et al. 2010). RNNLM models the distribution of text data with recurrent neural networks by decomposing the joint probability into a product of a series of conditional probabilities. The model is usually trained with maximum likelihood estimation (MLE) for an precise distribution recovery, which tries to maximize the probability of each training instance.

RNNLM trained with MLE are claimed to be suffering from the exposure bias problem, as shown by (Bengio et al. 2015), which leads to globally-incoherent generation. This problem occurs when an unseen prefix were generated during decoding, causing confusion for the generation of next word. (Bengio et al. 2015) propose Scheduled Sampling to alleviate this problem, which forces the model to take into account of decoding behaviour in the training period. Later (Lamb et al. 2016) provide a stronger solution called Professor Forcing which utilizes adversarial training to ensure

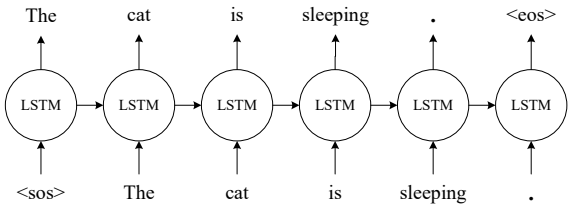


Figure 1: Illustration of the RNNLM architecture.

similar behaviours between training and decoding period. Despite successful targeting on the exposure bias problem, these works achieve relatively small improvement on generation quality. Our method does not target this problem directly, but in fact alleviate it by discouraging the generation of rare patterns, thus lowering the probability of generating an unseen prefix.

Recently, some works propose to use reinforcement learning techniques to guide the RNNLM to further improve the generation quality. For example, (Norouzi et al. 2016) and (Jaques et al. 2016) propose to combine MLE objective together with task-specific expected reward for better generation of sequential data. However there's currently no satisfying reward for high-quality text generation. PG-BLEU introduced in (Yu et al. 2017) use the target BLEU scores (Papineni et al. 2002) as the reward function, but later shows unacceptable low generation diversity(Lu et al. 2018).

Significant progress in this area was made after wide applications of adversarial training. (Yu et al. 2017) first introduce GANs to text generation tasks, called SeqGAN. This work uses policy gradient method together with Monte-Carlo sampling, making gradient-passing possible for discrete data. With the help of adversarial training, a higher generation quality is achieved. Later (Lin et al. 2017) introduce RankGAN, which uses a ranking mechanism for better reward signals and further improves the generation quality. Afterwards, (Guo et al. 2017) propose LeakGAN, which aims at long text generation using a hierarchical architecture. By leaking information from Discriminator to the Generator and incorporating dilated connections in the Generator, LeakGAN gets strong generation ability and achieves current state-of-the-art on text generation quality (Lu et al. 2018). Other works include that, (Fedus, Goodfellow, and Dai 2018) propose another adversarial framework to predict a word using surrounding words with actor-critic method. (Zhang et al. 2017b) apply feature matching for adversarial training instead of policy gradient method and also achieve higher generation quality.

Although recent methods have gained improvements on generation quality, they rely on adversarial training or reinforcement learning framework for modeling and optimization, which usually introduces difficulties in training and parameter tuning process. In this paper, we propose DDR instead to improve the generation quality of RNNLM, without the costly adversarial training and reinforcement learning process.

3 RNN Language Models

First, we introduce the typical RNNLM. Recurrent neural networks (RNNs) are well suited for processing sequential data, due to its recurrent nature. (Mikolov et al. 2010) first introduce RNNs for language modeling (RNNLM), that is, modeling the probability distribution of the language given the dataset. Here we denote sequence with length T as $Y_{1:T}$. RNNLM models the probability of $Y_{1:T}$ by decomposing $P(Y_{1:T})$ into a product of a series of conditional probabilities:

$$P(Y_{1:T}) = \prod_{t=1}^{T} P(y_t|Y_{1:t-1}),$$

where y_t is the t-th token in $Y_{1:T}$, and $Y_{1:0}$ represents a special start token. To encourage capturing long-term dependencies, the RNN cells are usually implemented by LSTM (Hochreiter and Schmidhuber 1997). The model architecture is shown in Fig. 1.

Maximum likelihood estimation (MLE, a.k.a. Teacher Forcing) (Williams and Zipser 1989) is usually used for training RNNLM. Given a dataset $\{Y^i_{1:T}\}_{i=1}^N$ whose elements are i.i.d. sampled from the real distribution P, MLE tries to find a model with distribution Q by maximizing the log-likelihood:

$$\begin{aligned}\max_Q \mathbb{E}_{Y_{1:T}\sim P}\, log\, Q(Y_{1:T}) =& \frac{1}{N}\sum_{i=1}^{N} log\, Q(Y^i_{1:T}) \\ =& \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} log\, Q(y_t|Y^i_{1:t-1}).\end{aligned}$$

This objective is equivalent to minimizing the Kullback-Leibler (KL) divergence, as shown in (Arjovsky and Bottou 2017).

$$KL(P||Q) = \sum_{Y_{1:T}} P(Y_{1:T}) log \frac{P(Y_{1:T})}{Q(Y_{1:T})}.$$

The value of KL divergence reaches its minimum at $Q = P$, which means MLE pushes model Q towards the real distribution P and achieves its optimal solution with a precise recovery of the real distribution.

Considering the expression of KL divergence, we can see that KL divergence is very sensitive on the value of $Q(Y_{1:T})$. If the true probability of $Y_{1:T}$ is positive, i.e. $P(Y_{1:T}) > 0$, the value inside KL goes to infinity as $Q(Y_{1:T}) \to 0$. This means KL divergence punishes much if real samples are not fitted well, showing a favor of high generation diversity. However, this is problematic when the training data contains some noises or rare patterns, which is often the case for real-world text data. The model wastes lot of its attention trying to fit such data, thus hurting its generation quality. In fact, such diversity from noises and rare pattern is not necessary, since it usually leads to generation errors in practice.

Notice that in theory, a model achieves highest diversity with uniform distribution, and achieves highest quality by always generating one best sample. However, neither of these two trivial cases are acceptable in practice. Our goal is thus to achieve higher Turing Test pass rate with flexible control of quality-diversity trade-off, while keeping competitive generation quality at certain diversity level.

4 Method

In this section, we propose a new differentiated distribution recovery approach, to replace the precise distribution recovery approach of MLE.

4.1 Differentiated Distribution Recovery

To make the model less influenced by samples with noises and rare patterns, we require the model to focus on fitting more significant patterns or instances with higher true probability. Instead of recovering the true probability distribution, the generated distribution need to differentiate between instances with higher probability and lower ones. To achieve this target, we require the generated probability be proportional to the true probability, shown as

$$Q(x) \propto P(x)^\beta.$$

With $\beta > 1$, we could expect that samples with higher real probability are augmented, while samples with lower real probability are discouraged. We call this approach *Differentiated Distribution Recovery* (DDR), since it gives differentiated treatment for different samples when trying to recover the real distribution. An illustration of DDR applied on a toy distribution is shown in Fig. 2.

We show that DDR achieves flexible quality-diversity trade-off control on this toy distribution. In Fig. 2, we see the quality increases while diversity decreases as β grows, such trade-off can be flexibly controlled by the parameter β. Also, DDR becomes more robust against noises with larger βs on quality evaluation, especially for smaller noises.

The next question is, how to make this approach possible in practice. In the following section, we propose a new objective function, with which the optimal generated probability can be written as the above form.

4.2 Objective Function

Consider a generalized form of the objective functions

$$\max_Q \mathbb{E}_{x\sim P}\, f[Q(x)],$$

where f is a real-valued monotonously increasing function.

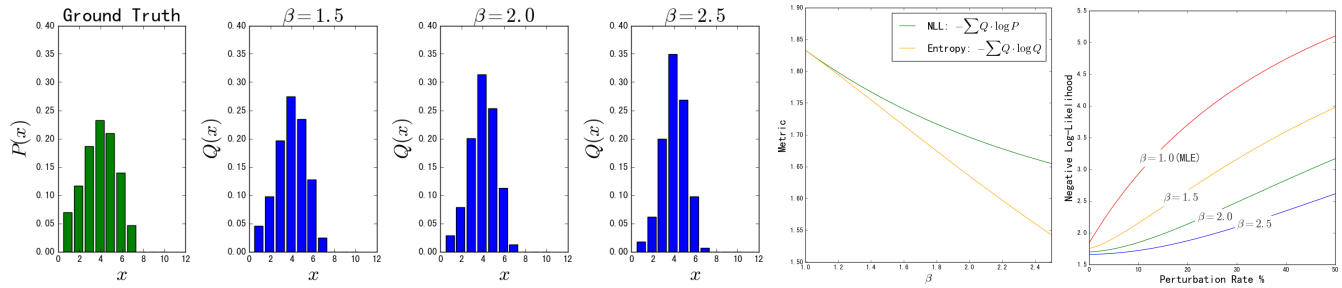
We prove that DDR is achieved by using a family of f functions, as shown in the following theorem.

Theorem 1. *Let P and Q be two discrete distributions. With an objective defined as*

$$\max_Q \mathbb{E}_{x\sim P}\, f[Q(x)],$$

where $f(Q(x);\alpha) = \alpha\cdot Q(x)^{\frac{1}{\alpha}} - \alpha$ and $\alpha > 1$. The optimal Q with respect to the objective can be written as:

$$Q^*(x) = \frac{P(x)^\beta}{\sum_x P(x)^\beta}, \quad \beta = \frac{\alpha}{\alpha-1}.$$



(a) (b) (c)

Figure 2: Illustration of differentiated distribution recovery behavior of DDR on a toy categorical distribution. **(a)**:The ground truth distribution $P(x)$ as well as optimal model distribution $Q(x)$ under different βs. **(b)**: Changes of generation quality and diversity as β grows in DDR. Lower Negative Log-Likelihood(NLL) indicates higher quality, and higher Entropy indicates higher diversity. **(c)**: Generation quality changes as we blend the toy distribution with a uniform distribution as noise. Perturbation Rate shows the proportion of added noise.

Proof. The objective is a constrained optimization problem with following constraints

$$\sum_x Q(x) = 1, \quad 0 \le Q(x) \le 1,$$

which can be solved by applying method of Lagrange multipliers. The Lagrange function is

$$L(Q(x), \lambda, \gamma, \eta) = \sum_x P(x) \cdot f[Q(x)] + \lambda[1 - \sum_x Q(x)] - \gamma Q(x) + \eta[Q(x) - 1].$$

Optimal solution is reached where the first derivatives equal 0:

$$\frac{\partial L}{\partial Q(x)} = 0, \frac{\partial L}{\partial \lambda} = 0, \frac{\partial L}{\partial \gamma} = 0, \frac{\partial L}{\partial \eta} = 0.$$

From the first equation, we have

$$P(x) \cdot f'[Q^*(x)] = c,$$

where $c = \lambda + \gamma - \eta$. Note that f' takes derivative on $Q(x)$ rather than x, so that

$$f'[Q(x)] = Q(x)^{\frac{1-\alpha}{\alpha}}.$$

Put this equation back into the first restriction above, we get

$$P(x) \cdot Q^*(x)^{\frac{1-\alpha}{\alpha}} = c,$$

which can be rewritten as $Q^*(x) = [\frac{P(x)}{c}]^{\frac{\alpha}{\alpha-1}}$. Then we denote $\frac{\alpha}{\alpha-1}$ as β and do summation over x on both sides , we get

$$\sum_x [\frac{P(x)}{c}]^{\beta} = 1,$$

by solving which we can get the value of c:

$$c = (\sum_x P(x)^{\beta})^{\frac{1}{\beta}}.$$

Put c back into the equation above, we have

$$Q^*(x) = \frac{P(x)^{\beta}}{\sum_x P(x)^{\beta}},$$

thus finish the proof. □

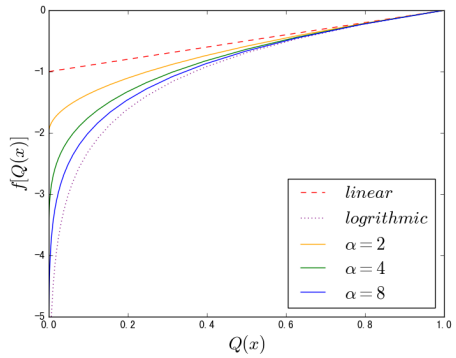


Figure 3: Illustration of our proposed function f. We can see f gradually changes from linear to logarithmic as α grows.

According to Theorem 1, DDR can be realised by setting the function f as

$$f(Q(x); \alpha) = \alpha \cdot Q(x)^{\frac{1}{\alpha}} - \alpha, \quad \alpha = \frac{\beta}{\beta - 1} > 1.$$

As long as $\alpha > 1$, we have $\beta > 1$, and β grows monotonously as α decreases. This is an asymptotic function family from linear to logarithmic as α grows (See Fig. 3 for illustration). We have

$$f(Q(x); 1) = Q(x) - 1,$$
$$\lim_{\alpha \to \infty} f(Q(x); \alpha) = ln\ Q(x).$$

These cases correspond to $\beta \to \infty$ and $\beta = 1$ respectively, where $\beta = 1$ leads to MLE exactly.

We apply the above objective function for training RNNLM. Given a text dataset $\mathcal{D} = \{Y^i_{1:T}\}_{i=1}^N$ and a RNN

language model Q, the loss function is defined as follows:

$$\begin{aligned}\mathcal{L}(\mathcal{D};\alpha) &= -\frac{1}{N}\sum_{i=1}^{N}\alpha\cdot Q(Y^i_{1:T})^{\frac{1}{\alpha}} \\ &= -\frac{\alpha}{N}\sum_{i=1}^{N}\prod_{t=1}^{T}Q(y^i_t|Y^i_{1:t-1})^{\frac{1}{\alpha}} \\ &= -\frac{\alpha}{N}\sum_{i=1}^{N}exp\{\frac{1}{\alpha}\sum_{t=1}^{T}log\,Q(y^i_t|Y^i_{1:t-1})\}.\end{aligned}$$

Constant terms are omitted for simplicity. We can see that the optimization can be conducted similarly to MLE, without costly adversarial training or reinforcement learning process, as compared to SeqGAN and LeakGAN.

5 Experiments

In this section, we conduct experiments on both synthetic and real datasets to compare our proposed DDR with other baselines, including RNNLM, SeqGAN and LeakGAN.

5.1 Empirical Settings

Following the experiment settings in (Guo et al. 2017), we conduct experiments on synthetic data, MSCOCO Image Caption dataset (Chen et al. 2015) and EMNLP2017 WMT News dataset[1]. We compare our results with three baseline methods, including RNNLM trained with MLE, SeqGAN (Yu et al. 2017), and LeakGAN (Guo et al. 2017).

For each dataset, architecture of the RNN part used for generation (also called the generator) in each models are kept the same. The generators all contain an embedding layer, an LSTM layer, and a softmax output layer. The architecture of discriminators and managers in SeqGAN and LeakGAN remain unchanged. Embedding dimensions and number of LSTM hidden nodes are all set to 32 on synthetic data and 128 on other two datasets. All models are trained using the Adam optimizer (Kingma and Ba 2014). Similar to SeqGAN and LeakGAN, we also pre-train our model using MLE before applying DDR.

For our DDR method in each experiment, we only report the results on some picked αs according to different datasets and neglect those causing extremely low diversity.

5.2 Experiments on Synthetic Data

We first test generation quality of DDR on synthetic data, under which setting we know the real data distribution in advance. We follow exactly the same setting as (Yu et al. 2017), and use the same oracle model G_{oracle}. An RNN model with LSTM is adopted, with all parameters initialized following the normal distribution $\mathcal{N}(0,1)$. 10,000 sequences of length 20 are sampled using the oracle model, which constitute the training set. We use *Negative Log-Likelihood(NLL)* as quality evaluation metric, which is defined as

$$NLL = -\mathbb{E}_{Y_{1:T}\sim Q}[\sum_{t=1}^{T} log\, G_{oracle}(y_t|Y_{1:t-1})],$$

[1]http://statmt.org/wmt17/translation-task.html

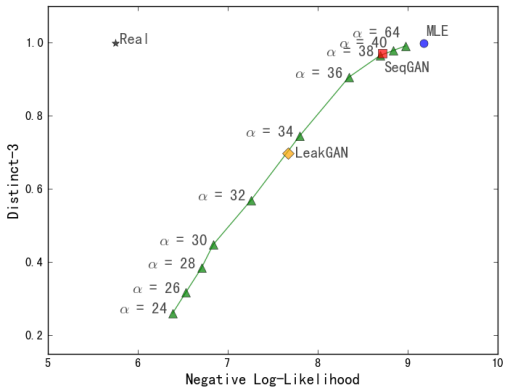


Figure 4: Evaluation of quality and diversity on synthetic dataset. Lower NLL indicates higher quality, and higher Distinct-3 indicates higher diversity.

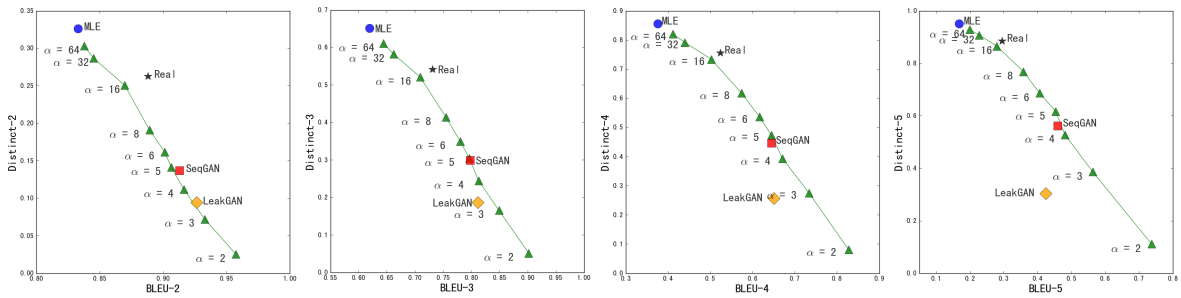
where Q still denotes the distribution learned by generation models. This is a metric used in many works (Yu et al. 2017; Lin et al. 2017; Guo et al. 2017), which is representative of generation quality since a higher probability under the real distribution indicates higher chance to pass the Turing Test as explained in (Yu et al. 2017). For the diversity evaluation metric, we use *Distinct-n*(Li et al. 2015), which estimates the richness of distinct n-grams in generated samples. The value is the number of distinct n-grams divided by total number of n-grams. Here we set $n = 3$ according to the maximal sequence length 20, and evaluate on 5,000 random generated samples.

The results are shown in Fig. 4. We see that DDR consistently outperforms MLE on quality evaluation for all αs, showing a good substitution of MLE for higher generation quality. As we lowering the value of α, the generation quality gradually increases. Although the diversity is also decreasing, DDR achieves competitive quality compared with SeqGAN and LeakGAN at their corresponding diversity level. Moreover, α can be further lowered for even higher generation quality than all baseline methods. The quality-diversity curve of DDR is quite smooth, showing flexible trade-off control with the parameter α. Notice that DDR has approximately the same training overhead with MLE, which is much more time-efficient than SeqGAN and LeakGAN. We also observe that the evaluation of real data is far above all models on this synthetic dataset, this can be explained by the random initialization of parameters, causing the real distribution near uniform, where almost no training data can be viewed as noise under NLL evaluation.

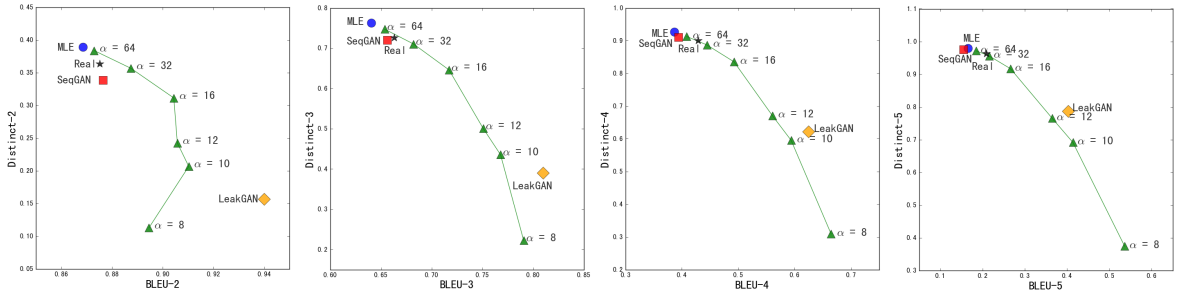
5.3 Experiments on Real Data

We also conduct experiments on real text data. We use two public datasets, MSCOCO Image Caption dataset and EMNLP2017 WMT News dataset.

MSCOCO dataset contains natural images and their corresponding human-labeled descriptions. We only use the captions in our experiments. The whole dataset contains 417,126 sentences. We follow (Guo et al. 2017) to do the preprocessing and use exactly the same final training and



(a) MSCOCO dataset



(b) WMT dataset

Figure 5: Evaluation of quality and diversity on MSCOCO and WMT dataset. Higher BLEU-n and Distinct-n indicates higher quality and diversity respectively.

test set. Words with frequency lower than 10 as well as the sentences containing them are removed. Then 80,000 unique sentences are sampled to constitute the training set, and another 5,000 unique sentences for the test set. The maximum sentence length is 32 and the vocabulary size is 4,840.

WMT News dataset contains long sentences of News. We use the Europarl part of the dataset in our experiment. The whole dataset contains 2,218,201 sentences. Words with frequency lower than 400 as well as the sentences containing them are removed. Sentences with length shorter than 15 or longer than 50 are also removed. Then we sample 200,000 sentences for the training set and another 10,000 sentences for test set. The final vocabulary size is 6,655.

To evaluate the quality of generated samples, we calculate BLEU scores (Papineni et al. 2002) following (Yu et al. 2017; Lin et al. 2017; Guo et al. 2017). The BLEU score measures n-gram overlap on generated samples with the whole test set as reference, so it reflects the similarity between generated samples and real data. We randomly generate 5,000 samples for each method and report the BLEU-(2 to 5) scores as well as Distinct-(2 to 5) in Fig. 5. We also consider the case where a model achieves high BLEU scores by memorizing and reproducing training data. We remove generated samples which appeared in the training data before quality evaluation, but the BLEU scores make little difference. Therefore, we did not report these results.

According to the results on real datasets, DDR achieves higher generation quality than MLE even with the largest α. As α decreases, generation quality increases as diversity drops as trade-off. The parameter α still provides flexible trade-off control, shown by smooth quality-diversity curves. The result of 2-grams on WMT dataset seems to be an exception, the reason may be that short n-grams are not representative enough for long text, since results of longer n-grams all perform as expected. We see that DDR achieves competitive quality with SeqGAN at the same diversity level, however, LeakGAN performs better on WMT dataset but worse on MSCOCO dataset. This can be explained by the specially designed model structure for long text generation in LeakGAN. Even though, DDR shows the ability to outperform LeakGAN on 4 and 5-gram quality evaluation with small αs, revealing superiority of DDR on quality-diversity trade-off control. Here the evaluation of real data is not far from other models compare with which on synthetic data, since we use a test set for evaluation, thus lots of rare patterns in training data would be viewed as noise.

5.4 Human Evaluation

Since BLEU scores only focus on local structures of text, it may not be sufficient to use them alone for evaluating the generation quality. As a result, we also conduct human evaluation by running a Turing Test on MSCOCO dataset. We invite 10 CS majored Ph.D students, to give scores on generated text samples. For each method, we sample 50 sentences randomly from each trained model. We show some of the generated samples here in Table 1. We mix all samples from different models, and ask the human annotators to make judgements on whether a sample is a real one or not. If the sample is regarded as a real one, it gets +1 score and 0 otherwise. A sample should be regarded as real, as long as one think it is possibly written by a human. Finally, the average score acquired by a model can be viewed as its Turing Test pass rate.

Table 1: Generated samples on MSCOCO dataset.

Method	Generated samples
Real data	A cat stuck in a car with a slightly opened window.
	Bicycles , cars and a trash can in a garage.
	A lady talking a self portrait in a fancy bathroom.
	A man standing in a white kitchen with his arms folded.
MLE	Two young children playing a video game on the Nintendo Wii.
	Two pancakes on a white paper plate with sauce on the plate.
	A suitcase with vanilla and yellow markings on top of it.
	Birds flying on a stone bench next to the tree.
SeqGAN	A group of people standing on top of a snow covered mountain.
	Two people standing next to each other in the dirt.
	A brown horse standing next to a white fence on the beach.
	A cow standing in a grassy area near a body of water.
LeakGAN	A bicycle is locked to a fence by a truck.
	The interior of a bathroom with a long mirror and partially tiled walls.
	A small bathroom has toilet , medicine cabinet , and small sink.
	A woman riding a bicycle down a street in front of shops.
DDR ($\alpha = 64$)	A woman wearing tennis gear holding a racket and her racquet.
	A dog sitting on a chair in front of a birthday cake.
	A bald man lays on a bed in the yellow floral pot.
	A girl is flying a kite in the sky into the airport.
DDR ($\alpha = 8$)	A man is sitting in a chair with a white cat.
	A guy is jumping in the air with a skateboard.
	A tall giraffe standing on top of a lush green field.
	Two women pose in front of a very tall building.
DDR ($\alpha = 2$)	A couple of young men playing a game of baseball.
	A couple of zebra standing on top of a lush green field.
	A red stop sign sitting on the side of a road.
	A man hitting a tennis ball with a tennis racquet.

The human evaluation results are shown in Table 2. We find the Turing Test scores are in accordance with BLEU scores, from which we also see that DDR clearly outperforms all other baseline methods. MLE method has a Turing Test pass rate lower than 50%, which means more than half of its samples contain logical or grammatical errors, showing its insufficiency for high-quality text generation. Our DDR method gets higher pass rate with lower α, and achieve a high pass rate of 0.932 with the lowest α, which is even much higher than the ground-truth. This can be explained by the behavior of DDR, that is, DDR promotes generation quality by neglecting bad training instances with low quality.

5.5 Robustness Test

To show proof for the advantage of DDR in fighting with noises, we further run a robustness test on MSCOCO dataset. We add 10% noise samples into the training set and keep the test set unchanged. A noise sample is composed of random words picked from the vocabulary. Due to the randomness, we believe these samples are of low quality and has low probability $P(x)$. We expect our method to neglect these samples, that is, to generate similar samples with very low probability, especially with small αs. We generate 5,000 samples for each model trained on this corrupted dataset, and calculate the percentage of generated bad samples. A generated sample is regarded as bad if its BLUE-2 score on the test set is lower than 0.001.

Table 2: The average Turing Test scores on MSCOCO dataset and percentage of generated bad samples on corrupted MSCOCO dataset.

Method	Turing Test Score	Bad Samples %
Ground Truth	0.772	-
MLE	0.490	8.0
SeqGAN	0.706	2.2
LeakGAN	0.758	**0.0**
DDR($\alpha=64$)	0.586	0.4
DDR($\alpha=32$)	-	0.1
DDR($\alpha=16$)	-	**0.0**
DDR($\alpha=8$)	0.692	**0.0**
DDR($\alpha=2$)	**0.932**	**0.0**

We report the percentage of generated bad samples for each method in Table 2. From the result, we can see that MLE is strongly influenced by the noises, and starts to generate bad samples with a high probability. However, DDR methods are much less influenced, and even totally neglect the impact of bad training instances with a α smaller than 16. LeakGAN also shows such robustness, but SeqGAN does not. These results show that DDR has the ability to alleviate the influence of noisy data on generation quality, showing robustness against noises as compared with MLE.

6 Conclusion and Future Work

In this paper, we proposed a new differentiated distribution recovery approach to train RNN-based text generation models for higher Turing Test pass rate, instead of the traditional MLE approach. Specifically, the objective function $\mathbb{E}_{x\sim P}\, f[Q(x)]$ is used for optimization, with $f(Q(x);\alpha) = \alpha \cdot Q(x)^{\frac{1}{\alpha}} - \alpha \quad (\alpha > 1)$. With this objective function, the closed form of the optimal generated probability can be written as $Q^*(x) = P(x)^\beta / \sum_x P(x)^\beta$, with $\beta = \frac{\alpha}{\alpha-1} > 1$. We can see that this kind of optimal generation probability has the ability to eliminate the influences of the noises or rare pattern, thus enhance the quality of generated texts by trading-off some diversity. With the parameter α, such trade-off can be flexibly controlled. Furthermore, the objective function can be directly optimized with a similar end-to-end approach as RNNLM, without expensive adversarial training or reinforcement learning process as in SeqGAN and LeakGAN. Finally, our experiments on both synthetic data and real-world data show clearly the ability of our DDR method to generate competitively high quality texts, as compared with several strong baselines, including SeqGAN and LeakGAN, in terms of both automatic evaluation metrics and human evaluations.

For future work, we plan to apply DDR for conditional text generation tasks, such as image captioning or dialogue generation. We are also interested in investigating deeper on the differences between accurate distribution recovery and differentiated distribution recovery approaches.

Acknowledgement

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 61773362, 61425016, 61472401, 61722211, and 61872338, the Youth Innovation Promotion Association CAS under Grants No. 20144310, and 2016102, and the National Key R&D Program of China under Grants No. 2016QY02D0405.

References

Arjovsky, M., and Bottou, L. 2017. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*.

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

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, 1171–1179.

Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Fedus, W.; Goodfellow, I.; and Dai, A. M. 2018. Maskgan: Better text generation via filling in the _. *arXiv preprint arXiv:1801.07736*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Guo, J.; Lu, S.; Cai, H.; Zhang, W.; Yu, Y.; and Wang, J. 2017. Long text generation via adversarial training with leaked information. *arXiv preprint arXiv:1709.08624*.

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

Huszár, F. 2015. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*.

Jaques, N.; Gu, S.; Bahdanau, D.; Hernández-Lobato, J. M.; Turner, R. E.; and Eck, D. 2016. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. *arXiv preprint arXiv:1611.02796*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

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

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

Lu, S.; Zhu, Y.; Zhang, W.; Wang, J.; and Yu, Y. 2018. Neural text generation: Past, present and beyond. *arXiv preprint arXiv:1803.07133*.

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

Norouzi, M.; Bengio, S.; Jaitly, N.; Schuster, M.; Wu, Y.; Schuurmans, D.; et al. 2016. Reward augmented maximum likelihood for neural structured prediction. In *Advances In Neural Information Processing Systems*, 1723–1731.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Rennie, S. J.; Marcheret, E.; Mroueh, Y.; Ross, J.; and Goel, V. 2017. Self-critical sequence training for image captioning. In *CVPR*, volume 1, 3.

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

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

Zhang, J.; Feng, Y.; Wang, D.; Wang, Y.; Abel, A.; Zhang, S.; and Zhang, A. 2017a. Flexible and creative chinese poetry generation using neural memory. *arXiv preprint arXiv:1705.03773*.

Zhang, Y.; Gan, Z.; Fan, K.; Chen, Z.; Henao, R.; Shen, D.; and Carin, L. 2017b. Adversarial feature matching for text generation. *arXiv preprint arXiv:1706.03850*.