Text Generation with Diffusion Language Models: A Pre-training Approach with Continuous Paragraph Denoise

Zhenghao Lin [1 2 3] **Yeyun Gong** [4] **Yelong Shen** [5] **Tong Wu** [6 2] **Zhihao Fan** [7 2]
Chen Lin [1 3] **Nan Duan** [4] **Weizhu Chen** [5]

Abstract

In this paper, we introduce a novel d**I**ffusion language mod**E**l pre-training framework for text generation, which we call **GENIE**. GENIE is a large-scale pre-trained diffusion language model that consists of an encoder and a diffusion-based decoder, which can generate text by gradually transforming a random noise sequence into a coherent text sequence. To pre-train GENIE on a large-scale language corpus, we design a new *continuous paragraph denoise* objective, which encourages the diffusion-decoder to reconstruct a clean text paragraph from a corrupted version while preserving the semantic and syntactic coherence. We evaluate GENIE on four downstream text generation benchmarks, namely XSUM, CNN/DAILYMAIL, GIGAWORD, and COMMONGEN. Our experimental results show that GENIE achieves comparable performance with the state-of-the-art autoregressive models on these benchmarks, and generates more diverse text samples. The code and models of GENIE are available at https://github.com/microsoft/ProphetNet/tree/master/GENIE.

1. Introduction

Text generation is a crucial task in natural language processing, which aims to produce fluent and coherent texts for various applications. Previous text generation methods mainly relied on recurrent neural networks (RNNs) (Pawade et al., 2018; Song et al., 2018; Gu et al., 2016a; Qi et al., 2021), which generate texts sequentially from left to right. However, RNNs suffer from issues such as long-term dependency and exposure bias. Recently, Transformer (Vaswani et al., 2017b), a self-attention-based neural network, has emerged as the dominant paradigm for text generation, thanks to its ability to capture global dependencies and leverage large-scale pre-trained language models (Qi et al., 2020; Lewis et al., 2019; Raffel et al., 2020a). Transformer-based methods typically adopt an encoder-decoder architecture, where the encoder maps the input text to a sequence of hidden vectors, and the decoder generates the output text either autoregressively (AR) or non-autoregressively (NAR). Generally, AR decoding is more accurate but slower, as it predicts each word conditioned on the previous ones. NAR decoding is faster but less precise, as it predicts all words simultaneously without modeling their dependencies.

This paper presents a new text generation approach, called GENIE, that integrates the diffusion model and Transformer-based method. The diffusion model is a generative model that reverses a stochastic process of adding noise to the data and has shown promising results in image (Ho et al., 2020; Song et al., 2020), molecule (Hoogeboom et al., 2022), video (Ho et al., 2022), and text (Li et al., 2022b; Gong et al., 2022; Strudel et al., 2022; Reid et al., 2022) generation. GENIE follows the encoder-decoder architecture, where the encoder transforms the input text to hidden vectors, and the diffusion model restores the output text from a random Gaussian noise, guided by the encoder's hidden vectors. The diffusion model iterates over multiple time steps and gradually denoises the output text at each step.

To leverage the large-scale unlabeled text data, we also propose an end-to-end pre-training method for GENIE. Unlike the existing pre-training tasks that involve masking or splitting tokens or texts (Qi et al., 2020; Lewis et al., 2019; Raffel et al., 2020a), we design a novel pre-training task, called *continuous paragraph denoise* (CPD). CPD requires the model to predict the noise added to continuous paragraphs in the current time step, given the paragraph context and the noisy paragraph information.

We evaluate GENIE on four popular text generation benchmarks: XSum (Narayan et al., 2018), CNN/DailyMail (Her-

[1]School of Informatics, Xiamen University [2]This work was done during an internship in MSRA [3]Shanghai Artificial Intelligence Laboratory [4]Microsoft Research Asia [5]Microsoft [6]Tsinghua University [7]Fudan University. Correspondence to: Chen Lin <chenlin@xmu.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

mann et al., 2015), Gigaword (Rush et al., 2015), and CommonGen (Lin et al., 2019). The experimental results demonstrate that GENIE achieves competitive performance with Transformer-based AR methods, and that the proposed pre-training method can effectively improve the performance. We notice that GENIE has significantly increased the diversity of the generated texts. To evaluate the multiple outputs of the generation model, we design an automatic annotation method based on the large language model. We also conduct ablation studies to analyze the impact of the diffusion steps and pre-training steps.

The main contributions of this work are summarized as follows:

- We propose GENIE, the first large-scale language pre-trained model based on the diffusion framework, which can generate high-quality texts for sequence-to-sequence tasks.
- We introduce a novel CPD loss as the pre-training objective, which can enhance the model's ability to denoise noisy texts and capture paragraph-level coherence.
- We validate the effectiveness of the pre-trained diffusion model on downstream tasks, and design a new automatic annotation method for the evaluation based on a large language model. We also provide extensive analyses of the model's behavior and properties.

2. Preliminary

2.1. Task Definition

In the classical sequence-to-sequence task, given a source text $\boldsymbol{s} = \{w_1^s, w_2^s, \ldots, w_n^s\}$ with n tokens, it generates target text sequence $\boldsymbol{y} = \{w_1^y, w_2^y, \ldots, w_n^y\}$. A sequence generation model can achieve this by modeling the conditional probability: $p(\boldsymbol{y} \mid \boldsymbol{s})$.

2.2. Diffusion model

In the diffusion model, the diffusion process can be regarded as a discrete-time Markov process. The diffusion process starts with the initial state $\boldsymbol{x}_0$ at time step $t = 0$, where $\boldsymbol{x}_0$ is the Gaussian distribution of the original data. It gradually adds Gaussian noises to $\boldsymbol{x}_0$ in the forward diffusion process according to a variance schedule $\beta_1, ..., \beta_T$. At the time step $t+1$, the latent variable $\boldsymbol{x}_{t+1}$ is only determined by the $\boldsymbol{x}_t$ at time t, expressed as:

$$q(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{x}_{t+1}; \sqrt{1-\beta_{t+1}}\boldsymbol{x}_t, \beta_{t+1}\mathbf{I}\right). \quad (1)$$

As t increases, $\boldsymbol{x}_t$ becomes closer to standard Gaussian noise $\mathcal{N}(\boldsymbol{x}_T; 0, \mathbf{I})$.

The diffusion model learns to perform the inverse diffusion process during generation, which predicts the noise given the current state $\boldsymbol{x}_t$ at time step t. The previous state $\boldsymbol{x}_{t-1}$ can be reconstructed by subtracting the noise and re-scaling the mean. Thus, the distribution of $\boldsymbol{x}_{t-1}$ given $\boldsymbol{x}_t$ is a Gaussian with mean μ_θ^{t-1} and variance $\sigma_\theta^{t-1\,2}$:

$$p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{x}_{t-1}; \mu_\theta^{t-1}, \sigma_\theta^{t-1}\right), \quad (2)$$

$$\mu_\theta^{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_\theta(\boldsymbol{x}_t, t)\right), \quad (3)$$

$$\sigma_\theta^{t-1\,2} = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t, \quad (4)$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ and z_θ is predicted by a neural network parameterized by θ. The diffusion model is trained by minimizing the mean squared error between μ_θ^{t-1} and the true mean $\hat{\mu}_{t-1}$, which is computed from the reverse conditional distribution $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$:

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_{t-1}; \hat{\mu}_{t-1}, \hat{\beta}_{t-1}\mathbf{I}\right), \quad (5)$$

$$\hat{\mu}_\theta^{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t. \quad (6)$$

Following the variational lower bound (VLB) approach (Ho et al., 2020), the diffusion model can be trained by minimizing the loss function:

$$\mathcal{L}_{diff} = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left\|\mu_\theta^{t-1} - \hat{\mu}_{t-1}\right\|^2. \quad (7)$$

3. Model

GENIE is the proposed diffusion language model for pre-training, it adopts the sequence-to-sequence framework as illustrated in Figure 1. GENIE could generate a high-quality text sequence $\boldsymbol{y}$ given a source text $\boldsymbol{s}$, such as producing $\boldsymbol{y}$: *Messi's performance* from $\boldsymbol{s}$: *In the World Cup 2022, [MASK] won people's praise.*. To achieve this, GENIE leverages two components: a bidirectional encoder model and a cross-attention diffusion model. The encoder model encodes the source text $\boldsymbol{s}$ into a set of hidden vectors $\boldsymbol{H_s}$ = Encoder($\boldsymbol{s}$), which indicates the distributed representation of $\boldsymbol{s}$. The diffusion model takes $\boldsymbol{H_s}$ and a Gaussian noise as inputs, and iteratively refines the data by applying a sequence of denoising operations. In contrast to the traditional autoregressive text generation paradigm, which generates one token at a time, the diffusion model in GENIE outputs the sequence of embeddings in parallel at each denoising step, making GENIE a non-autoregressive generation (NAR) model.

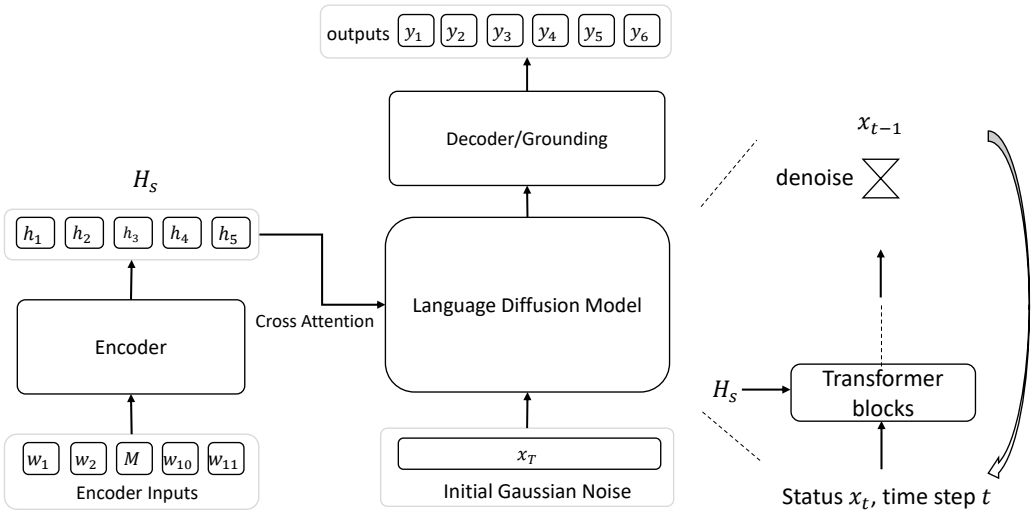

Figure 1. The framework of GENIE. We take the masked source sequence $\boldsymbol{s}$ as the input of the Encoder to obtain the hidden information $\boldsymbol{H_s}$, and interact with Language Diffusion Model through cross attention. The language Diffusion Model restores the randomly initial Gaussian noise to the output text $\boldsymbol{y}$ through the iterative denoising and grounding process.

Encoder The encoder in GENIE is a 6-layer transformer model which takes the source text $\boldsymbol{s}$ as input with bidirectional self-attention. Specifically, given a source text sequence $\boldsymbol{s} = \{w_1^s, w_2^s, \ldots, w_n^s\}$ with n tokens, the encoder model computes the vector $\boldsymbol{h}_i$ for each token w_i. Thus, the source text $\boldsymbol{s}$ can be represented as $\boldsymbol{H_s}$ by the encoder model:

$$\boldsymbol{H_s} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_n\} = \text{Encoder}(\boldsymbol{s}). \tag{8}$$

Language Diffusion Model The diffusion model in GENIE is a 6-layer transformer with cross-attention on the source text representation $\boldsymbol{H_s}$. It learns to predict Gaussian noise $z_\theta\left(\boldsymbol{x}_t, t, \boldsymbol{H_s}\right)$ conditioned on the current diffusion step t and the state $\boldsymbol{x}_t$, where $\boldsymbol{x}_t$ is the continuous latent representation of the target text. We use an embedding function and a clamping trick to ground the continuous state $\boldsymbol{x}_t$ with discrete target tokens, which will be elaborated in the following section.

Inference Phase To generate text from the diffusion model, we start from the final step $t = T$ and sample a state $\boldsymbol{x}_T$ from a standard Gaussian distribution. Then we iteratively generate the noise for the previous step using equations 3 and 4, and subtract it from the current state to obtain $\boldsymbol{x}_{t-1}$. After arriving at $t = 0$, we apply the clamping trick (Li et al., 2022b) to replace the values of $\boldsymbol{x}_0$ with its closest word embeddings, and then decode the discrete tokens from $\boldsymbol{x}_0$.

Training Phase To train the diffusion model for sequence-to-sequence tasks, we first convert the target sequence $\boldsymbol{y} = \{w_1^y, w_2^y, \ldots, w_n^y\}$ into a continuous state $\boldsymbol{x}_0$ using the embedding function with an additional Gaussian noise permutation, which can be expressed as:

$$q(\boldsymbol{x}_0|\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{x}_0; \text{Emb}(\boldsymbol{y}), \beta_0 \mathbf{I}\right), \tag{9}$$

where $\text{Emb}(\cdot)$ is embedding function, β_0 represents the scaling of variance at time step $t = 0$. Then we apply the forward diffusion process (equation 1) to obtain the state $\boldsymbol{x}_t$ at any step t as a function of $\boldsymbol{x}_0$, as shown in equation:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, \sqrt{1-\bar{\alpha}_t}\mathbf{I}\right), \tag{10}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. In the training phase, we sample a random step t to calculate $\boldsymbol{x}_t$, and then use the denoising architecture to predict the noise for that step, based on the cross-attention with the source representation $\boldsymbol{H}_s$. The mean and variance of the predicted noise are given by equations 11:

$$\mu_\theta^{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_\theta\left(\boldsymbol{x}_t, t, \boldsymbol{H}_s\right)\right), \tag{11}$$

where z_θ is the output of the denoising architecture and θ are its parameters. The training objective is to minimize the squared error between the predicted and true noise, as well as the reconstruction error between $\boldsymbol{x}_0$ and the target embeddings, as expressed in equation 12:

$$\begin{aligned}\mathcal{L}_{s2s} = \mathop{\mathbb{E}}_{q(\boldsymbol{x}_{0:T}|\boldsymbol{y})}[\sum_{t=1}^{T} \left\|\mu_\theta^{t-1} - \hat{\mu}_{t-1}\right\|^2 \\ + \left\|\text{Emb}(\boldsymbol{y}) - \mu_\theta^0\right\|^2 - \log p_\theta(\boldsymbol{y}|\boldsymbol{x}_0)],\end{aligned} \tag{12}$$

where $p_\theta(\boldsymbol{y}|\boldsymbol{x}_0) = \prod_{i=1}^{n} p_\theta(w_i^y|\boldsymbol{x}_0)$, represents mapping the continuous latent variable $\boldsymbol{x}_0$ into the discrete space token w_i^y.

3.1. Pre-training GENIE

Diffusion models have great potential for natural language generation (NLG) due to their ability to produce diverse outputs. However, they have been largely overlooked in NLG because of their slow convergence and low quality compared to autoregressive models. In this section, we address these challenges by pre-training a diffusion language model and introducing a novel, tailored pre-training task. The novel pre-training task we propose is called *continuous paragraph denoise* (CPD). CPD aims to train the model to predict the noise added to a continuous paragraph in the current diffusion step, given the paragraph and its surrounding context.

Specifically, given a document $\boldsymbol{d} = \{w_1^d, w_2^d, \ldots, w_l^d\}$ with l words, we randomly select a paragraph $\boldsymbol{p} = \{w_1^p, w_2^p, \ldots, w_m^p\}$ from $\boldsymbol{d}$, where $m = \lfloor \gamma * l \rfloor$ is the paragraph length and γ is a predefined ratio. We mask the paragraph in the document with a special token ([MASK]), and feed the masked document $\boldsymbol{d'} = \{w_1^{d'}, w_2^{d'}, \ldots, [\text{MASK}], \ldots, w_{l-m}^{d'}\}$ to the GENIE encoder. We also apply the forward diffusion process to the paragraph $\boldsymbol{p}$ and obtain a noisy version $\boldsymbol{x}_t$ at a random step t, and feed it to the GENIE denoising architecture. The denoising architecture then uses the cross-attention with the source representation $\boldsymbol{H}_s$ to predict the noise for the current step, using equations 11. In summary, the pre-training objective of CPD is to minimize the same loss as in equation 12, except that $\boldsymbol{y}$ is replaced by $\boldsymbol{p}$ and $\boldsymbol{x}_0$ is the embedded paragraph with noise.

Through this pre-training task, the diffusion model can enhance its semantic understanding of the continuous text and its denoising ability at each diffusion step. Moreover, the CPD task is self-supervised and does not rely on external labeled data sources, so it can fully exploit the information in the original pre-trained corpus.

4. Experiments and Results

In this section, we will introduce the details of GENIE pre-training, the data setting, and show extensive experimental results on various NLG downstream tasks.

4.1. GENIE Pre-training

Model Framework Our model uses a 6-layer transformer as the the encoder, and a 6-layer cross-attention transformer as the denoising architecture. In particular, in denoising architecture, we use the random embedding function to map discrete tokens into continuous variables. We set the latent variable dim to 768 and embedding dim to 128.

Pre-training Data Recent works have shown that pre-training on large-scale corpus can improve the performance of the model on downstream tasks (Lewis et al., 2019; Qi et al., 2020), which is also applicable to GENIE based on the diffusion model. Following BART (Lewis et al., 2019), we use pre-training data consisting of 160Gb of news, books, stories, and web text. We segment sentences belonging to different chapters and ensure that the input text length does not exceed 512.

Pre-training Setting We use the CPD task mentioned in §3.1 to pre-train GENIE on a large-scale corpus. The proportion of continuous paragraph γ sets to 30%. Hence, for the 512-length input, the target length is 153. We randomly extract 153 length targets from the input text and leave the [MASK] token at the extracted position. In the training process, we use Adam optimizer (Kingma & Ba, 2015) with a learning rate 1e-4, and we set the batch size to 512. We pre-trained our model on 8 × 40GB NVIDIA A100 GPUs with 5 million steps, lasting for 50 days. In the fine-tuning phase, we use the final pre-training model checkpoint to conduct fine-tuning on various downstream tasks.

4.2. Fine-tune on Downstream Tasks

To verify the effectiveness of pre-training on GENIE based on the diffusion model, we fine-tune and verify the effect of GENIE on various downstream tasks. Through the above task, we can prove that the pre-trained GENIE can quickly adapt to different types of NLG tasks without long-time training like other diffusion models.

Text Summarization As an important task in the NLG field, text summarization aims to summarize long documents into fluent short texts. In the experiment, we selected three widely used datasets: (a) GIGAWORD corpus (Rush et al., 2015), (b) CNN/DAILYMAIL (Hermann et al., 2015), and (c) XSUM (Narayan et al., 2018). In the process of fine-tuning, we set the learning rate to 5e-5 and the 120K training steps for all three datasets. In the inference process, we randomly sample 10 Gaussian noises for iteration denoising, and use the highest score as the final generated result. For different sample numbers, please refer to Appendix C. During evaluation, we following the existing work (Lewis et al., 2019; Qi et al., 2020), reporting F1 scores of **ROUGE-1**, **ROUGE-2**, and **ROUGE-L** on test set.

Common Sense Generation Common sense generation tasks require the model to have the ability of generative commonsense reasoning. Specifically, given a series of common sense concepts, the model needs to generate coherent statements based on these concepts that adhere to real-world

Table 1. Results of Semi-NAR, NAR and AR on XSUM. Index **OVERALL** represents the average value of **ROUGE-1**, **ROUGE-2** and **ROUGE-L**. It should be noted that GENIE belongs to Semi-NAR.

Methods	Pattern	XSUM			
		ROUGE-1	ROUGE-2	ROUGE-L	OVERALL
NAT (Gu et al., 2017)	NAR	24.0	3.9	20.3	16.1
iNAT (Lee et al., 2018)		24.0	4.0	20.4	16.1
CMLM (Ghazvininejad et al., 2019)		23.8	3.6	20.2	15.9
LevT (Gu et al., 2019b)		24.8	4.2	20.9	16.6
BANG (Qi et al., 2021)		32.6	9.0	27.4	23.0
ELMER (Li et al., 2022a)		**38.3**	**14.2**	**29.9**	**27.5**
LSTM (Greff et al., 2017)	AR	25.1	6.9	19.9	17.3
Transformer (Vaswani et al., 2017b)		30.7	10.8	24.5	22.0
MASS (Song et al., 2019)		39.7	17.2	31.9	29.6
BART (Lewis et al., 2019)		39.8	17.2	32.2	29.7
ProphetNet (Qi et al., 2020)		39.9	17.1	32.1	29.7
BANG (Qi et al., 2021)		**41.1**	**18.4**	**33.2**	**30.9**
InsT (Stern et al., 2019)	Semi-NAR	17.7	5.2	16.1	13.0
iNAT (Lee et al., 2018)		27.0	6.9	22.4	18.8
CMLM (Ghazvininejad et al., 2019)		29.1	7.7	23.0	20.0
LevT (Gu et al., 2019b)		25.3	7.4	21.5	18.1
BANG (Qi et al., 2021)		34.7	11.7	29.2	25.2
GENIE (w/o pre-train)		38.9	17.5	31.0	29.1
GENIE		**42.9**	**21.4**	**35.1**	**33.2**

Table 2. The main results on CNN/DAILYMAIL and GIGAWORD.

Method	CNN/DAILYMAIL			GIGAWORD		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
NAG-BERT (Su et al., 2021)	-	-	-	35.1	16.5	33.3
LSTM (Greff et al., 2017)	37.3	15.7	34.4	34.2	16.0	31.8
Transformer (Vaswani et al., 2017a)	39.5	16.7	36.7	37.1	18.4	34.5
BART (Lewis et al., 2019)	41.3	19.4	38.1	38.6	19.5	35.7
MASS (Song et al., 2019)	42.1	19.5	39.0	38.7	19.7	35.9
ProphetNet (Qi et al., 2020)	42.5	19.7	39.5	38.9	19.9	36.0
GENIE (w/o pre-train)	43.8	20.6	41.2	43.7	23.3	40.8
GENIE	**45.6**	**23.2**	**43.1**	**45.7**	**25.8**	**42.9**

scenarios. We select the widely used dataset COMMONGEN (Lin et al., 2019) to evaluate whether GENIE has good creativity and reasoning ability in natural language generating. In the fine-tuning phase, we set the learning rate to 1e-4 and train the model for 10k steps in total. Finally, we randomly sampled 10 Gaussian noises and selected the best sample as the final result. Referring to the previous work (Lin et al., 2019), we reported the indicators including F1 scores of **ROUGE-2/L**, **BLEU-3/4**, **CIDEr**, and **SPICE**.

4.3. Baselines

We compare GENIE with the baselines of several mainstream methods. Specifically, these methods can be divided into two groups. The first group is the NAR model, including NAT (Gu et al., 2017), iNAT (Lee et al., 2018), NAG-BERT (Su et al., 2021), CMLM (Ghazvininejad et al., 2019), LevT (Gu et al., 2019b), ConstLeven (Susanto et al., 2020), BANG (Qi et al., 2021), ELMER (Li et al., 2022a) and InsT (Stern et al., 2019). Among them, InsT, iNAT, CMLM, LevT, ConstLeven, and BANG can also be used in Semi-NAR, which can optimize the generation quality through multiple NAR iterations. It is worth noting that GENIE also belongs to the Semi-NAR model.

The second group is AR model, the model of encoder-decoder structure including LSTM (Greff et al., 2017), Transformer (Vaswani et al., 2017a), bRNN-CopyNet (Gu et al., 2016a), Trans-CopyNet (Lin et al., 2019), MeanPooling-CopyNet (Lin et al., 2019) without pre-training, and strong baselines MASS (Song et al., 2019), BART (Lewis et al., 2019), T5 (Raffel et al., 2020b), BANG (Qi et al., 2021), and ProphetNet (Qi et al., 2020) with large scale pre-training. For large-scale pre-training models mentioned above, we select the base version of the model, which is equivalent to the total number of GENIE parameters.

4.4. Main Results

We present the results of GENIE and the baselines on XSUM, CNN/DAILYMAIL, GIGAWORD, COMMONGEN in

Table 3. The main results on COMMONGEN.

Method	COMMONGEN					
	ROUGE-2/L		BLEU-3/4		CIDEr	SPICE
bRNN-CopyNet (Gu et al., 2016b)	9.2	30.6	13.6	7.8	6.0	16.9
Trans-CopyNet (Lin et al., 2019)	11.1	32.6	17.2	10.6	7.0	18.0
MeanPooling-CopyNet (Lin et al., 2019)	11.4	34.6	14.8	8.9	7.2	20.2
LevT (Gu et al., 2019a)	12.2	35.4	23.1	15.0	8.9	21.4
ConstLeven (Susanto et al., 2020)	13.5	35.2	21.3	12.3	**11.1**	23.2
T5-Base (Raffel et al., 2020b)	15.3	36.2	28.1	18.0	9.7	**23.4**
GENIE (w/o pre-train)	14.6	36.0	21.0	12.5	8.1	20.6
GENIE	**26.2**	**43.9**	**29.5**	**19.6**	10.3	**23.4**

Table 1, Table 2, and Table 3. Our results demonstrate that the pre-trained GENIE is a powerful NAR model for text generation. Especially on the XSUM dataset, GENIE outperforms other NAR and Semi-NAR methods by a large margin, and on all three text summarization datasets, GENIE achieves comparable quality to the pre-trained AR model. In addition, GENIE shows creativity and logic in common sense generation tasks. On COMMONGEN, GENIE surpasses other baseline models, including T5 which has been pre-trained on a large-scale corpus.

We also compare the pre-trained GENIE and GENIE trained from scratch (w/o pre-train). As shown in Table 1 and Table 2, pre-training significantly improves the **ROUGE-1**, **ROUGE-2**, **ROUGE-L** scores of GENIE on the three text summarization datasets. Similarly, the results on COMMONGEN in Table 3 indicate that pre-training enhances the performance of GENIE on this task. These results confirm the effectiveness of our pre-training method.

4.5. Generate Diversity Comparison

With the emergence of the diffusion-based model such as GENIE, the advantages of text generation in diversity will be gradually valued. In this experiment, we will use both quantitative metrics and qualitative examples to show the richness of GENIE in text generation.

To measure the diversity of GENIE generation, we use **SELF-BLEU** as the metric. The lower the **SELF-BLEU** score, the more diverse the generated texts are. For comparison, we use BART, a state-of-the-art autoregressive model, which is pre-trained on large-scale corpora. For BART, we apply different decoding methods of autoregressive models, such as greedy search, beam search (Xiao et al., 2022), diverse beam search (diversity strength = 0.8) (Vijayakumar et al., 2016), typical sampling ($\tau = 1.2$) (Meister et al., 2022), top-k sampling ($k = 50$) (Fan et al., 2018), and nucleus sampling ($p = 0.92$) (Holtzman et al., 2020). These decoding methods can generate multiple texts from the same source sequence. In this experiment, we generate 10 different target sequences for each source sequence using GENIE and BART. Then we use the 10 summaries generated from XSUM, CNN/DAILYMAIL, and GIGAWORD to calculate the **SELF-BLEU** scores.

As shown in Table 4, although the diversity of autoregressive generation can be slightly improved by using diverse beam search or some sampling methods with BART, the improvement is not significant. On the other hand, the diversity of generation is greatly enhanced by using the GENIE. The large gaps in **SELF-BLEU** indicate that GENIE can generate more diverse texts, not just varying a few words.

To complement the quantitative metrics, we also provide a case study in Appendix A to analyze the quality of the texts generated by BART and GENIE. We find that the autoregressive generation method can produce high-quality texts when there is only one output, but when generating multiple outputs, even with different decoding methods, it is hard to increase its diversity, and there may be many repeated prefixes. In contrast, the diffusion generation method can maintain the quality of generation while offering rich diversity.

However, it may not be fair to compare GENIE directly with the single reference to prove that GENIE can achieve diversity without compromising quality. Therefore, we design a new evaluation method. We use `text-davinci-003` version of InstructGPT (Ouyang et al., 2022), which is based on the large language model (LLM) GPT-3.5, to score our generated texts, that is, to evaluate the quality of the generated summaries. Specifically, we first obtain the sample set (10 summaries generated by BART using diverse beam search and 10 summaries generated by GENIE), and design a prompt to input into `text-davinci-003` to score the generated summaries, while counting the number of high-quality summaries within the 10 summaries generated by BART and GENIE respectively. We conduct the experiment on the three different text summarization datasets and use two evaluation methods, *Average Summary Score* represents the average score given by `text-davinci-003`, ranging from 1 to 3, and *Average High-quality Summary* represents the average number of high-quality summaries in 10 samples, ranging from 0 to 10. For more detailed experimental settings, please refer to Appendix B.

As shown in Table 5, although GENIE's scores are slightly

Table 4. **SELF-BLEU** score of BART and GENIE generated results. For each data sample, we use BART and GENIE to generate 10 summaries to evaluate diversity.

Model	Generate Method	XSUM	CNN/DAILYMAIL	GIGAWORD
BART	Greedy Search	100.0	100.0	100.0
	Beam Search	93.4	96.2	90.2
	Diverse Beam Search	75.6	84.1	71.8
	Typical Sample	76.9	84.6	80.1
	Top-k Sample	80.2	85.2	82.6
	Nucleus Sample	79.1	83.5	79.4
GENIE	Diffusion	**29.3**	**37.6**	**39.9**

Table 5. Large language model evaluation on three summarization benchmarks.

Method	XSUM		CNN/DAILYMAIL		GIGAWORD	
	BART	GENIE	BART	GENIE	BART	GENIE
Average Summary Score	2.69	2.58	2.96	2.90	2.58	2.46
Average High-quality Summary	6.91	5.95	9.66	9.04	5.99	4.99

Table 6. Effect of pre-training step, on XSUM. The result is the optimal value of 5 Gaussian samples.

Model	ROUGE-1	ROUGE-2	ROUGE-L
w/o pre-train	37.3	15.3	29.4
GENIE(100w)	39.4	17.1	31.5
GENIE(200w)	40.4	18.2	32.5
GENIE(300w)	40.6	18.5	32.8
GENIE(400w)	40.9	18.7	33.0
GENIE(500w)	41.2	19.1	33.4

Table 7. Effect of the proportion of continuous paragraphs, on XSUM. The result is the optimal value of 5 Gaussian samples.

CPD Proportion	ROUGE-1	ROUGE-2	ROUGE-L
$\gamma = 15\%$	$40.24_{\pm 0.03}$	$18.03_{\pm 0.01}$	$32.30_{\pm 0.02}$
$\gamma = 20\%$	$40.14_{\pm 0.10}$	$17.90_{\pm 0.04}$	$32.21_{\pm 0.08}$
$\gamma = 25\%$	$40.24_{\pm 0.10}$	$18.12_{\pm 0.04}$	$32.37_{\pm 0.04}$
$\gamma = 30\%$	$\mathbf{40.37_{\pm 0.04}}$	$\mathbf{18.20_{\pm 0.02}}$	$\mathbf{32.58_{\pm 0.05}}$
$\gamma = 35\%$	$40.29_{\pm 0.06}$	$18.18_{\pm 0.02}$	$32.45_{\pm 0.06}$
$\gamma = 40\%$	$40.15_{\pm 0.08}$	$17.87_{\pm 0.07}$	$32.12_{\pm 0.09}$

Table 8. Difference between uniform time schedule sample(UI) and loss aware sample(LA), on XSUM. The result is the optimal value of 5 Gaussian samples.

Method	ROUGE-1	ROUGE-2	ROUGE-L
$\gamma = 15\%$, UI	$40.24_{\pm 0.03}$	$18.03_{\pm 0.01}$	$32.30_{\pm 0.02}$
$\gamma = 15\%$, LA	$40.06_{\pm 0.04}$	$17.90_{\pm 0.02}$	$32.17_{\pm 0.05}$
$\gamma = 30\%$, UI	$40.37_{\pm 0.04}$	$18.20_{\pm 0.02}$	$32.58_{\pm 0.05}$
$\gamma = 30\%$, LA	$40.18_{\pm 0.03}$	$17.94_{\pm 0.02}$	$32.24_{\pm 0.02}$

lower than BART's, according to the results in Table 4, the diversity of samples generated by BART is much lower than GENIE. Given the trade-off between diversity and quality, the score difference is within the acceptable range. Moreover, the result of *Average High-quality Summary* shows that there are still enough high-quality summaries in the case of high diversity. Such advantages of GENIE deserve our attention and further exploration in our future work.

4.6. Impact of Pre-training Steps

Our pre-training method and the diffusion model are designed to achieve long-term convergence and unlimited potential, but they also require a large amount of pre-training time. Here we investigate how the pre-training steps affect the performance of our model compared with a non-pre-trained GENIE on the XSUM dataset. We fine-tune the checkpoints obtained at 1 million step intervals from pre-training and evaluate them using 5 random Gaussian noises, selecting the highest score as the final result. As shown in Table 6, pre-training for only 1 million steps can significantly improve the quality of generation over the non-pre-trained GENIE. Moreover, we can see from the results that pre-training continues to steadily boost the performance of the GENIE on the downstream task as the pre-training steps increase.

4.7. Impact of Pre-training Parameters

In this subsection, we examine the effect of important pre-training parameters on the pre-training performance. First, in the unsupervised pre-training method CPD, we need to explore how the proportion of continuous paragraphs γ influences the pre-training performance. We vary the value of γ from 15% to 40% (with 5% intervals) and conduct 2 million pre-training steps for each value. After the pre-training, we evaluate the pre-training effect by fine-tuning on XSUM. For a rigorous evaluation, we sample 5 Gaussian noises, repeat the experiment 5 times with different random seeds, and report the mean and standard deviation of the results, each time choosing the highest score as the final result. As shown in Table 7, too large or too small values

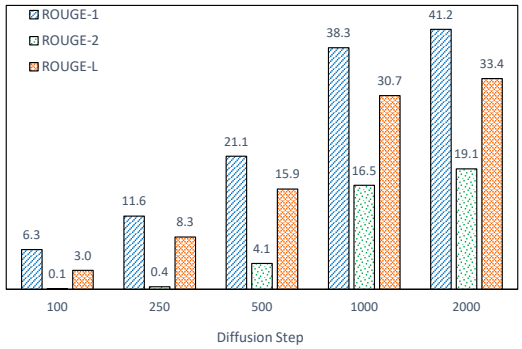

Figure 2. Effect of different diffusion steps on text generation quality, on XSUM. The result is the optimal value of 5 Gaussian samples.

Table 9. Using the same time step T during the training and inference phases. The result is the optimal value of 5 Gaussian samples, on XSUM.

Total Time Step	ROUGE-1	ROUGE-2	ROUGE-L
T=25	40.3	18.0	32.4
T=50	40.7	18.5	32.9
T=100	40.9	18.7	33.1
T=200	41.1	18.7	33.2
T=500	41.1	18.8	33.2
T=1000	41.1	18.9	33.2
T=2000	41.2	19.1	33.4

of γ lead to instability and poor performance of the pre-trained model. Pre-training is more stable and effective when $\gamma = 30\%$.

Second, we investigate the time step sampling method used in the pre-training. Before each training step, we need to sample a time step as part of the model input. The existing two common time step sampling methods are uniform sample and loss-aware sample. The former assigns equal probabilities to each time step, while the latter updates the sampling weights according to the training loss, so that more important time steps have higher chances of being sampled. In the experiment, we use these two sampling methods, test them on two different values of γ (15% and 30%), and perform a rigorous evaluation similar to the previous experiment. As shown in Table 8, we observe that under 2 million steps of pre-training, the uniform sample outperforms the loss-aware sample for different values of γ. Intuitively, although the loss-aware sample can speed up the convergence of the diffusion model, we hope that the model can learn sufficient knowledge at each time step during the pre-training, so that it can converge faster and perform better on downstream tasks.

Table 10. Generate sequences with length of 100 using batch size=1 and batch size=100, and compare the generation speed.

Model	batch = 1(s)	batch = 100(s, s/Sample)
BART	1.81	19.33, 0.19
GENIE(T=25)	0.61	4.62, 0.046
GENIE(T=50)	1.21	9.25, 0.093
GENIE(T=100)	2.43	19.18, 0.19
GENIE(T=200)	4.79	38.32, 0.38
GENIE(T=500)	11.71	96.12, 0.96
GENIE(T=1000)	23.72	193.71, 1.94
GENIE(T=2000)	47.63	387.79, 3.88

4.8. Impact of Diffusion Time Step

The number of diffusion time steps has a great impact on the quality of generation. We explore how the GENIE performs under two different settings for diffusion steps on the XSUM dataset.

Assuming that the total number of diffusion steps $T = 2000$, we set the interval step of inverse diffusion to 1, 2, 4, 8, 20, and the corresponding numbers of inverse diffusion steps are 2000, 1000, 500, 250, 100. In this experiment, we sample 5 Gaussian noises and choose the best denoising result. As shown in Figure 2, we can clearly see that when the number of inverse diffusion steps is small, the quality of generation with GENIE deteriorates significantly. As the number of inverse diffusion steps increases to 1000, the generation quality of GENIE becomes stable.

In the second experimental setting, we make the total number of diffusion time steps T in the fine-tuning and inference phase consistent. Specifically, we set different total time steps $T = 25, 50, 100, 200, 500, 1000, 2000$, and discuss the performance of the fine-tuned model in the corresponding time steps on the downstream tasks. The experimental results (XSum) under the second experimental setup are shown in Table 9. We found that under this setting, GENIE has less loss in generation quality and can generate text more quickly at the expense of a bit of quality.

Moreover, we hope to present the generation speed of the GENIE at different diffusion time steps through specific metrics. Specifically, we set batch size=1 and batch size=100 to generate sequences with a length of 100, and record the generation time. For the setting of batch size=100, we simultaneously calculate the time required to generate a batch of text and the average time required to generate each sample. As shown in Table 10, we calculated the time required to generate text under different diffusion time steps, and compared it with the autoregression model BART under the same settings. It can be found that the generation speed of the BART and GENIE is similar when the time step T=100.

5. Related Work

5.1. Large Scale Pre-training Language Models

Recently, a major breakthrough has been made in the model of pre-training on large scale corpus. As unidirectional language models, GPT (Radford et al., 2018), GPT2 (Radford et al., 2019) modeling the text based on left-to-right, and predict the next token according to the token appearing on the left. At the same time, bidirectional language models, which uses bidirectional encoder to model text, can obtain better context sensitive representation, such as BERT (Devlin et al., 2019) and RoBERT (Liu et al., 2019). RoBERT optimizes pre-training tasks compared to BERT, both of which significantly improve the ability of natural language understanding. In order to improve the performance of the large scale pre-training model in natural language generation, some works has designed pre-training tasks based on the standard framework of sequence-to-sequence. MASS (Song et al., 2019) lets the model predict the short masked token span step by step, while ProphetNet (Qi et al., 2020) predict more words in each step to ease local over fitting.

5.2. Diffusion Models for Text

In recent years, diffusion model has achieved great success in the domains of image generation (Ramesh et al., 2022; Saharia et al., 2022; Rombach et al., 2022). Because of its amazing generation quality, some works apply diffusion model in text generation domains. Diffusion-LM (Li et al., 2022b) maps discrete tokens into continuous latent variable, achieving more complex controllable text generation through continuous diffusion. In the field of text revision where non-autoregressive method is widely used, DiffusER (Reid et al., 2022) also uses the diffusion model to implement the edit based generative processes. DiffuSeq (Gong et al., 2022) achieves conditional text generation with a new method which controlled information is also involved in the diffusion process. Different from the above work, we build a novel language model based on the diffusion model for the first time, using the standard enocoder-decoder framework. For our best knowledge, we are the first to adopt large scale pre-training on the language model based on the diffusion model.

6. Conclusion

In this paper, we have presented a novel diffusion language model GENIE, which leverages a large-scale corpus for pre-training. Our model adopts a sequence-to-sequence framework, where a bidirectional encoder encodes the source sequence and a denoising decoder predicts and removes noise from the target sequence in a non-auto-regressive fashion. This design allows us to generate diverse text by gradually refining the output from a noisy initial state. Moreover, we have introduced a new pre-training method called *continuous paragraph denoise*, which aims to denoise whole paragraphs as the target sequence. Our experiments on various NLG tasks demonstrate that GENIE can produce high-quality and diverse text, and validate the benefits of pre-training our diffusion model on a large-scale corpus. However, the inference speed and training speed of text diffusion models still need to be improved. In the future, how to achieve a fast and better text generation diffusion model is a direction worthy of in-depth research.

Acknowledgments

Chen Lin is the corresponding author. Chen Lin is supported by National Key R&D Program of China (No. 2022ZD0160501) and the Natural Science Foundation of China (No. 61972328).

References

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Fan, A., Lewis, M., and Dauphin, Y. N. Hierarchical neural story generation. In *ACL (1)*, pp. 889–898. Association for Computational Linguistics, 2018.

Ghazvininejad, M., Levy, O., Liu, Y., and Zettlemoyer, L. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*, 2019.

Gong, S., Li, M., Feng, J., Wu, Z., and Kong, L. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Networks Learn. Syst.*, 28(10):2222–2232, 2017.

Gu, J., Lu, Z., Li, H., and Li, V. O. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*, 2016a.

Gu, J., Lu, Z., Li, H., and Li, V. O. K. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL (1)*. The Association for Computer Linguistics, 2016b.

Gu, J., Bradbury, J., Xiong, C., Li, V. O., and Socher, R. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

Gu, J., Wang, C., and Zhao, J. Levenshtein transformer. In *NeurIPS*, pp. 11179–11189, 2019a.

Gu, J., Wang, C., and Zhao, J. Levenshtein transformer. In *Advances in Neural Information Processing Systems*, pp. 11181–11191, 2019b.

Hermann, K. M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. Teaching machines to read and comprehend. In *NIPS*, pp. 1693–1701, 2015.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. In *ICLR*. OpenReview.net, 2020.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

Lee, J., Mansimov, E., and Cho, K. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv preprint arXiv:1802.06901*, 2018.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Li, J., Tang, T., Zhao, W. X., Nie, J., and Wen, J. ELMER: A non-autoregressive pre-trained language model for efficient and effective text generation. *CoRR*, abs/2210.13304, 2022a.

Li, X. L., Thickstun, J., Gulrajani, I., Liang, P., and Hashimoto, T. B. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022b.

Lin, B. Y., Zhou, W., Shen, M., Zhou, P., Bhagavatula, C., Choi, Y., and Ren, X. Commongen: A constrained text generation challenge for generative commonsense reasoning. *arXiv preprint arXiv:1911.03705*, 2019.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

Meister, C., Pimentel, T., Wiher, G., and Cotterell, R. Typical decoding for natural language generation. *CoRR*, abs/2202.00666, 2022.

Narayan, S., Cohen, S. B., and Lapata, M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *EMNLP*, pp. 1797–1807. Association for Computational Linguistics, 2018.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Pawade, D., Sakhapara, A., Jain, M., Jain, N., and Gada, K. Story scrambler-automatic text generation using word level rnn-lstm. *International Journal of Information Technology and Computer Science (IJITCS)*, 10(6):44–53, 2018.

Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., and Zhou, M. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*, 2020.

Qi, W., Gong, Y., Jiao, J., Yan, Y., Chen, W., Liu, D., Tang, K., Li, H., Chen, J., Zhang, R., et al. Bang: Bridging autoregressive and non-autoregressive generation with large scale pretraining. In *International Conference on Machine Learning*, pp. 8630–8639. PMLR, 2021.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140): 1–67, 2020a. URL http://jmlr.org/papers/v21/20-074.html.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020b.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022.

Reid, M., Hellendoorn, V. J., and Neubig, G. Diffuser: Discrete diffusion via edit-based reconstruction. *arXiv preprint arXiv:2210.16886*, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10674–10685. IEEE, 2022.

Rush, A. M., Chopra, S., and Weston, J. A neural attention model for abstractive sentence summarization. In *EMNLP*, pp. 379–389. The Association for Computational Linguistics, 2015.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding. *CoRR*, abs/2205.11487, 2022.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T. MASS: masked sequence to sequence pre-training for language generation. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5926–5936. PMLR, 2019.

Song, L., Zhang, Y., Wang, Z., and Gildea, D. A graph-to-sequence model for amr-to-text generation. *arXiv preprint arXiv:1805.02473*, 2018.

Stern, M., Chan, W., Kiros, J., and Uszkoreit, J. Insertion transformer: Flexible sequence generation via insertion operations. *arXiv preprint arXiv:1902.03249*, 2019.

Strudel, R., Tallec, C., Altché, F., Du, Y., Ganin, Y., Mensch, A., Grathwohl, W., Savinov, N., Dieleman, S., Sifre, L., et al. Self-conditioned embedding diffusion for text generation. *arXiv preprint arXiv:2211.04236*, 2022.

Su, Y., Cai, D., Wang, Y., Vandyke, D., Baker, S., Li, P., and Collier, N. Non-autoregressive text generation with pre-trained language models. In *EACL*, pp. 234–243. Association for Computational Linguistics, 2021.

Susanto, R. H., Chollampatt, S., and Tan, L. Lexically constrained neural machine translation with levenshtein transformer. In *ACL*, pp. 3536–3543. Association for Computational Linguistics, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NIPS*, pp. 5998–6008, 2017a.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017b.

Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D. J., and Batra, D. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424, 2016.

Xiao, Y., Wu, L., Guo, J., Li, J., Zhang, M., Qin, T., and Liu, T. A survey on non-autoregressive generation for neural machine translation and beyond. *CoRR*, abs/2204.09269, 2022.

A. Case Study

In the section§4, we have made a rigorous analysis of the quality and diversity of GENIE. We hope that by comparing diffusion model with traditional autoregressive generation models, we can find the potential of diffusion model in natural language generation tasks. Nowadays, most of the excellent language generation models belong to autoregressive generation models, but at the same time, we also need

Table 11. Summary examples of BART and GENIE generated results.

source sequence I (abbreviated)	Those who participated in the Aberdeen Children of the 1950s project, which saw all primary pupils aged seven to 12 surveyed by the Medical Research Council in 1962, have been contacted. They have been asked to take part in the Scottish Family Health Study. It aims to investigate why diseases such as cancer can run in families.Those recruited will have their health tracked, with the intention of creating a Scottish "bio-bank" containing genetic, medical and family history and lifestyle information. The data gathered would help future research into the prevention, treatment and diagnosis of illnesses.
GENIE summaries I (diffusion)	1. health information have been recruited by university school primary pupils to help improve their lives. 2. a health project is to be recruited by learning for university researchers in scotland. 3. scientists in aberdeen are to meet experts in scotland to get more health data for their children.
BART summaries I (diversity beam search)	1. thousands of children from aberdeen are being recruited to help scientists investigate why diseases run in families. 2. thousands of children in scotland have been asked to take part in a new project to study their health. 3. thousands of children in scotland have been asked to take part in a new project to study their health..
source sequence II (abbreviated)	Elin Jones is expected to lay out plans where some areas of Welsh forest could be transferred to the private sector or to not for profit organisations.But she has already ruled out the widespread sale of Welsh woodlands.Forestry Commission Wales said it would explore the feasibility of transfer to the private sector case by case.The minister told BBC Radio Wales she plans to "compensate" the public by buying new land for new planting or management if any forest was sold off on a case-by-case basis."I don't want any stagnancy in the forest estate. I want it to work for public benefit whether that's economic or environmental or access benefit," she said."It's my view there should be no reduction in the publicly owned estate and I have asked the Forestry Commission to look at how it can make that estate work harder, provide a better return for the public."Whether that's in terms of public access, in terms of environmental benefit in the production of renewable energy or biomass potential or also in terms of the economic return from that forestry estate."
GENIE summaries II (diffusion)	1. the forestry minister is preparing to fill out its plan for some members of the public on wales' forests to be reduced. 2. the forestry minister is picking forward plans to tackle some of the companies in wales to develop a boost in the management of forests. 3. the forestry minister hopes to face plans continue on the future of wales' forest estate to be held to a growing and better access to the private sector.
BART summaries II (diversity beam search)	1. the environment minister has said there should be no "stagnancy" in the size of the welsh forest estate. 2. the forestry minister has said there should be no "stagnancy" in the size of the welsh forest estate. 3. the future of wales' forests could be decided by the environment minister.

Table 12. Example of prompt used by text-davinci-003.

Prompt Input > Deborah Steel, who was 37 and ran the Royal Standard in Ely, was last seen in the early hours of 28 December. Her body has not been found. No further action will be taken against a 50-year-old and a 70-year-old, both from Ely, Cambridgeshire Police said .A 72-year-old man from Ely has been re-bailed until 17 February. Ms Steel's disappearance was recently reclassified from a long-term missing person inquiry to a murder inquiry by officers.

If the following sentences as summary of the above article, please assign an overall score. Scores range from 1 to 3, 1 represents bad, 2 represents neutral, 3 represents good. The output format is 'Score: 1'.

Two of the three men arrested by detectives investigating the disappearance of a Cambridgeshire pub landlady in 1997 have been released.

Output > Score: 3

some new ideas and generation paradigm to make natural language generation not limited to autoregressive. The ways of natural language generation need to be diversified to broaden researchers' thinking, just as the application of diffusion model in natural language generation can bring rich diversity to the generated text. We are excited that the diversity of the content generated by the diffusion model does not come from a large number of wrong words or unrelated texts, but from different sentence patterns and different information obtained from the original text. This shows us the future prospects of the diffusion model in the natural language generation task.

In order to more intuitively show the quality of the text generated by the diffusion model and the autoregressive generation model, we selected two samples from the text summarization dataset XSUM in Table 11. For each sample, we used GENIE and BART to generate three summaries respectively, of which the BART generation method is diversity beam search. For display purposes, the source sequence has been intercepted and abbreviated to reduce the length. It can be seen from the generated summaries that if we only observe one sentence of the generated summary, the summary generated by the autoregressive model BART is of good quality and is related to the content of the source sequence, the generated text is relatively fluent due to the autoregressive generation mode. But if we look at the three generated summaries, the text generated by the autoregressive model BART obviously has a lot of duplication. In the second example, "there should be no 'stagnancy' in the size of the Welsh forest estate" has repeated descriptions in two different summaries. In the first example, there are even two summaries that are almost identical. The difference is only one meaningless full stop at the end. Although we use the diversity beam search generation method when generating, it is difficult to let the autoregressive model jump out of the essence of iterative generation to generate creative text.

In contrast, we can see that the summary generated by GENIE may not be as fluent as BART due to the non-autoregressive generation mode if only from the quality of single sentence generation. Once multiple summaries are generated, we can observe GENIE's creative generating ability. In the first example, GENIE describes the medical project in the source sequence from three related direction. Health information collection, project information and project objectives are mentioned in the generated summary. In the second example, "the forestry minister" mentioned a variety of measures on the forest industry, and the three summaries generated by GENIE described different parts respectively, including the formulation of public plans, cooperation with Welsh companies to promote forest management, and the involvement of private enterprises in the forest industry. Compared with the single information "there should be no 'stagnancy' in the size of the Welsh forest estate" provided by BART, although BART also provides a concise summary, it is difficult to conclude that BART's summary is better, because people are accustomed to understanding some problems from multiple perspectives in practice, rather than a conclusive conclusion.

After analyzing the above examples, we can observe the great potential of new language model GENIE. In the practical application of text generation, diversified generation results can be used in many scenarios. The unique generation method of diffusion model brings new ideas to text generation, and also lets us consider whether the single-label text generation really meets our needs. We do not need the diffusion model to be superior to the autoregressive generation model in all aspects. What we need is a new idea to bring more possibilities to text generation. We believe that the diffusion model can be widely used in text generation in the future.

B. Large Language Model Evaluation

Recently, the large language model has been widely used in various tasks with its amazing performance. In this paper, we use the large language model to evaluate the quality of the generated summary. We select `text-davinci-003` as the evaluation model in our experiment, the most important thing is the construction of the prompt which will input into the model.

As prompt example shown in Table 12, we divide the prompt into three parts. The first part is the text of the original article, the middle part is the evaluation requirements, and the end part is the summary that needs to be evaluated. Finally, we can get output score through the large language model. For each summary, we need to organize the above prompt for the model input, but the evaluation requirements for each summary are the same. During the test, we asked the model to give the following score to the summary: 1 represents bad, 2 represents neutral, 3 represents goods. After getting the score of each summary, we will further count the number of high-quality samples in the 10 samples generated. Here, we define high-quality summary as summary with a score equal to 3. Finally, we can summarize and integrate all the scores obtained, and count the average summary score and the average number of high-quality summaries.

C. Impact of Sample Number

Compared with the autoregressive model, GENIE based on the diffusion model can generate more diverse texts in the inference phase, even under the guidance of the source sequence. Different Gaussian noises sampled during denoising can often lead to completely different generation results. This method is more flexible, but it is not conducive to the evaluation against a single reference answer. However,

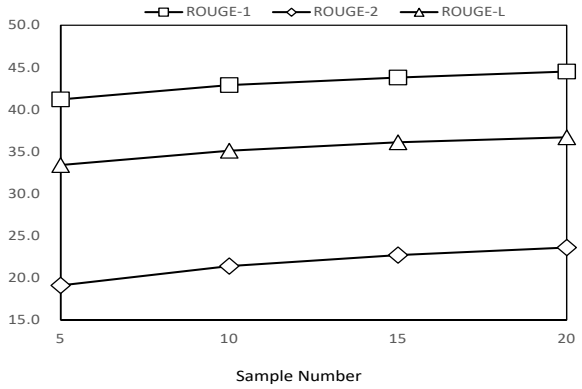

Figure 3. Effect of sample number, on XSUM.

as the number of Gaussian noises sampled increases, the generated text has a higher probability of approaching the single reference answer, and the corresponding evaluation score is higher. To this end, we test the performance of the model on the test set under different numbers of samples on the XSUM dataset. As shown in Figure 3, we evaluate the results of 5, 10, 15 and 20 samples. We can observe that as the number of samples increases, the more likely the generated sample is to be similar to the original label. The improvement of the similarity is more noticeable in the early stage of increasing the number of samples.